Variance Reduction for Faster Non-Convex Optimization

Zeyuan Allen-Zhu   Elad Hazan

presented by Mike & Vamsi

October 24, 2016

Problem

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

Problem

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

Functions $f(\cdot)$ are L-smooth.

Problem

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

Functions $f(\cdot)$ are L-smooth.

Typical ERM setting in

- single/multi index models
- neural newtorks etc.

Context

Neural networks are thriving

- From object/scene recognition to web-based search engines;
- From self-driving cars to gene decoding/synthesis

Context

Neural networks are thriving

- From object/scene recognition to web-based search engines;
- From self-driving cars to gene decoding/synthesis

Very large datasets

Extremely large computational load

Context

Neural networks are thriving

- From object/scene recognition to web-based search engines;
- From self-driving cars to gene decoding/synthesis

Very large datasets
Extremely large computational load

i.e., Need faster optimization schemes

Context – What's the bottleneck

No convexity

Context – What's the bottleneck

No convexity

$\implies$ One can only talk about getting to an ϵ-approximate stationary point

i.e., we need x s.t. $\|\nabla f(x)\|^2 < \epsilon$

Context – What's the bottleneck

No convexity

$\implies$ One can only talk about getting to an ϵ-approximate stationary point

i.e., we need x s.t. $\|\nabla f(x)\|^2 < \epsilon$

Stochastic Gradients *with random stopping*

Ghadimi and Lan 2013

$$\mathcal{O}\left(\left(\frac{L}{\epsilon} + \frac{L\sigma^2}{\epsilon^2}\right)(f(x_0) - f(x^*))\right)$$

x_0 : starting point, x^* : global minimizer

Context – What's the bottleneck

Stochastic Gradients *with random stopping*

$$O\left(\left(\frac{L}{\epsilon}+\frac{L\sigma^2}{\epsilon^2}\right)(f(x_0)-f(x^*))\right)$$

σ^2 : Variance of the stochastic gradient

Context – What's the bottleneck

Stochastic Gradients *with random stopping*

$$O\left(\left(\frac{L}{\epsilon}+\frac{L\sigma^2}{\epsilon^2}\right)(f(x_0)-f(x^*))\right)$$

σ^2 : Variance of the stochastic gradient

Costly for non-trivial losses (e.g., ReLU)!

Unless you can afford cluster-level computational power

Context – The fix

Stochastic Variance Reduced Gradients (SVRG)

Context – The fix

Stochastic Variance Reduced Gradients (SVRG)

For sum-of-smooth function objectives,

- #iterations independent of σ^2
- Overall cost (#iterations * gradient computation per iteration) cheaper than gradient descent!

Context – The fix

Stochastic Variance Reduced Gradients (SVRG)

For sum-of-smooth function objectives,

- #iterations independent of σ^2
- Overall cost (#iterations * gradient computation per iteration) cheaper than gradient descent!

$$O\left(\frac{n^{2/3}L}{\epsilon}(f(x_0) - f(x^*))\right)$$

Outline

- Why Variance reduction?
- The algorithm
- Convergence Proof
- Some experiments

Why Variance Reduction?

$$G := \frac{1}{n}\sum_{i=1}^{n} g(x_i) \qquad x_i \sim p(x)$$

G is an unbiased estimate of some unknown function
but may have *large* variance

Why Variance Reduction?

$$G := \frac{1}{n}\sum_{i=1}^{n} g(x_i) \qquad x_i \sim p(x)$$

G is an unbiased estimate of some unknown function
but may have *large* variance

replace G with some $\hat{G}$ s.t.

$$\mathbb{E}G = \mathbb{E}\hat{G} \qquad Var(G) \geq Var(\hat{G})$$

Why Variance Reduction? – basic probability

Monte-Carlo Variance reduction technique with a control variate ϕ

$$\hat{g} = g - a(\phi - \mu_\phi)$$

where $\mu_\phi = \mathbb{E}\phi$ is known, and a is some scalar.

Why Variance Reduction? – basic probability

Monte-Carlo Variance reduction technique with a control variate ϕ

$$\hat{g} = g - a(\phi - \mu_\phi)$$

where $\mu_\phi = \mathbb{E}\phi$ is known, and a is some scalar.

Hence $\mathbb{E}\hat{g} = \mathbb{E}g$ but

$$\frac{Var(\hat{g})}{Var(g)} = 1 - corr^2(g, \phi)$$

Why Variance Reduction? – basic probability

Monte-Carlo Variance reduction technique with a control variate ϕ

$$\hat{g} = g - a(\phi - \mu_\phi)$$

where $\mu_\phi = \mathbb{E}\phi$ is known, and a is some scalar.

Hence $\mathbb{E}\hat{g} = \mathbb{E}g$ but

$$\frac{Var(\hat{g})}{Var(g)} = 1 - corr^2(g, \phi)$$

g is the noisy-gradient for us

Variance Reduction – Stochastic gradients

Classical SVRG:

Variance Reduction – Stochastic gradients

Classical SVRG:

Initial estimate x_0^1

Outer epochs $(s = 1, \ldots, S)$

Inner iterations $(k = 0, \ldots, m-1)$

Final estimate x_m^S

Variance Reduction – Stochastic gradients

Classical SVRG:

Initial estimate x_0^1

Outer epochs $(s = 1, \ldots, S)$

- A snapshot vector x_0^s – Average estimate from pervious epoch
- $\tilde{\mu} = \nabla f(x_0^s)$

Variance Reduction – Stochastic gradients

Classical SVRG:

Initial estimate x_0^1

Outer epochs $(s = 1, \ldots, S)$

- A snapshot vector x_0^s – Average estimate from pervious epoch
- $\tilde{\mu} = \nabla f(x_0^s)$

Inner iterations $(k = 0, \ldots, m-1)$

- Gradient updates
- $i \in [n]$

$$x_{k+1}^s \leftarrow x_k^s - \eta \tilde{\nabla}$$

Variance Reduction – Stochastic gradients

Classical SVRG:

Initial estimate x_0^1

Outer epochs $(s = 1, \dots, S)$

- A snapshot vector x_0^s – Average estimate from pervious epoch
- $\tilde{\mu} = \nabla f(x_0^s)$

Inner iterations $(k = 0, \dots, m-1)$

- Gradient updates
- $i \in [n]$

$$x_{k+1}^s \leftarrow x_k^s - \eta \tilde{\nabla}$$
$$\tilde{\nabla}_k^s := \nabla f_i(x_k^s) - \nabla f_i(x_0^s) + \tilde{\mu}$$

Variance Reduction – Stochastic gradients

Classical SVRG:

Initial estimate x_0^1

Outer epochs $(s = 1, \ldots, S)$

- A snapshot vector x_0^s – Average estimate from pervious epoch
- $\tilde{\mu} = \nabla f(x_0^s)$

Inner iterations $(k = 0, \ldots, m-1)$

- Gradient updates
- $i \in [n]$

$$x_{k+1}^s \leftarrow x_k^s - \eta \tilde{\nabla}$$
$$\tilde{\nabla}_k^s := \nabla f_i(x_k^s) - \nabla f_i(x_0^s) + \tilde{\mu}$$

- $x_0^{s+1} \leftarrow x_m^s$

Variance Reduction – Stochastic gradients

Classical SVRG:

Initial estimate x_0^1

Outer epochs $(s = 1, \ldots, S)$

- A snapshot vector x_0^s – Average estimate from pervious epoch
- $\tilde{\mu} = \nabla f(x_0^s)$

Inner iterations $(k = 0, \ldots, m-1)$

- Gradient updates
- $i \in [n]$

$$x_{k+1}^s \leftarrow x_k^s - \eta \tilde{\nabla}$$
$$\tilde{\nabla}_k^s := \nabla f_i(x_k^s) - \nabla f_i(x_0^s) + \tilde{\mu}$$

- $x_0^{s+1} \leftarrow x_m^s$

Final estimate x_m^S

Non-convex SVRG:

Initial estimate x_0^1

Outer epochs ($s = 1, \dots, S$)

- A snapshot vector x_0^s – Average estimate from pervious epoch
- $\tilde{\mu} = \nabla f(x_0^s)$

Inner iterations ($k = 0, \dots, m-1$)

- Gradient updates
- $i \in [n]$

$$x_{k+1}^s \leftarrow x_k^s - \eta \tilde{\nabla}$$
$$\tilde{\nabla}_k^s := \nabla f_i(x_k^s) - \nabla f_i(x_0^s) + \tilde{\mu}$$

- $x_0^{s+1} \leftarrow x_m^s$

Return x_k^s for some $s \in [S]$ and $k \in [m]$

Proof Idea

Trick lies in handling $(\sigma_k^s)^2 := \|\nabla_k^s - \tilde{\nabla}_k^s\|^2$

Proof Idea

Trick lies in handling $(\sigma_k^s)^2 := \|\nabla_k^s - \tilde{\nabla}_k^s\|^2$

- Upper bound $\mathbb{E}(\sigma_k^s)^2$ by $\mathcal{O}(\|x_k^s - x_0^s\|^2)$
- Then argue that $\|x_k^s - x_0^s\|^2$ is at most a constant times $f(x_k^s) - f(x_0^s)$

Bounding the Variance

Upper bound $\mathbb{E}(\sigma_k^s)^2$ by $\mathcal{O}(\|x_k^s - x_0^s\|^2)$

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

Bounding the Variance

Upper bound $\mathbb{E}(\sigma_k^s)^2$ by $\mathcal{O}(\|x_k^s - x_0^s\|^2)$

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

- Each f_i is L-smooth

$$-\frac{L}{2}\|x-y\|^2 \leq f_i(x) - f_i(y) - \langle \nabla f_i(y), x-y\rangle \leq \frac{L}{2}\|x-y\|^2$$

Bounding the Variance

Using basic properties of the expectation with L-smoothness

$$\mathbb{E}(\sigma_m^s)^2 \leq L^2 \|x_m^s - x_0^s\|^2$$

Bounding the Variance

Using basic properties of the expectation with L-smoothness

$$\mathbb{E}(\sigma_m^s)^2 \leq L^2 \|x_m^s - x_0^s\|^2$$

Assume the previous bound holds for all $k = 0, \ldots, m-1$ and $s = 1, \ldots, S$ (Simplification 3 in the paper)

Mirror Descent Lemma

Want a bound on $\|x_k^s - x_0^s\|^2$ in terms of $f(x_k^s) - f(x_0^s)$

Mirror Descent Lemma

Want a bound on $\|x_k^s - x_0^s\|^2$ in terms of $f(x_k^s) - f(x_0^s)$

Use basic lemma from mirror descent:

Lemma 3.2: *If* $x_{k+1} = x_k - \eta\tilde{\nabla}_k$, *then for all* $u \in \mathbb{R}^d$ *it satisfies*

$$f(x_k) - f(u) \leq \frac{\eta}{2}(\|\nabla_k\|^2 + \mathbb{E}[\sigma_k^2]) + (\frac{1}{2\eta} + \frac{L}{2})\|x_k - u\|^2 - \frac{1}{2\eta}\mathbb{E}[\|x_{k+1} - u\|^2]$$

Using Lemma 3.2

Want to telescope this sum for the whole epoch to bound variance

Using smoothness instead of convexity cost us a factor of $\frac{L}{2}\|x_k - u\|^2$

Using Lemma 3.2

Want to telescope this sum for the whole epoch to bound variance

Using smoothness instead of convexity cost us a factor of $\frac{L}{2}\|x_k - u\|^2$

Can no longer telescope the whole epoch

Subepochs

Idea: Split each epoch into subepochs

Telescope terms in each subepoch

Bound full epoch in terms of sum of these subepochs

Telescope it!

Let $\eta = \frac{1}{m_0 L}$ where m_0 is the subepoch length

Apply Lemma 3.2 with $u = x_k$:

$$\sum_t \beta_t (f(x_{k+t}) - f(x_k)) \leq \frac{\eta}{2} \sum_t \beta_t (\|\nabla_{k+t}\|^2 + \sigma_{k+t}^2) - \frac{\beta_{m_0-1}}{6\eta} \|x_{k+m_0} - x_k\|^2$$

$\beta_0 = 1$ and $\beta_t = (1 + 1/m_0)^{-t}$ for $t = 1, \ldots, m_0 - 1$

Telescope it!

Let $\eta = \frac{1}{m_0 L}$ where m_0 is the subepoch length

Apply Lemma 3.2 with $u = x_k$:

$$\sum_t \beta_t (f(x_{k+t}) - f(x_k)) \le \frac{\eta}{2} \sum_t \beta_t (\|\nabla_{k+t}\|^2 + \sigma_{k+t}^2) - \frac{\beta_{m_0-1}}{6\eta} \|x_{k+m_0} - x_k\|^2$$

$\beta_0 = 1$ and $\beta_t = (1 + 1/m_0)^{-t}$ for $t = 1, \ldots, m_0 - 1$

Pretty ugly, let's add some simplifying assumptions

Simplifications

By definition, all β's are within a constant factor of each other

Simplifications

By definition, all β's are within a constant factor of each other

Simplification 1:

$$\text{Assume that } \beta_t = 1 \text{ for all } t = 0, \ldots, m_0 - 1$$

Simplifications

The average objective value is difficult to analyze

Simplifications

The average objective value is difficult to analyze
Simplification 2:

Assume that we can replace the average value with the last iterate

Telescoping sum (again)

Using these simplifications, we get:

$$f(x_{k+m_0}) - f(x_k) \leq \frac{\eta}{2m_0} \sum_t (\|\nabla_{k+t}\|^2 + \sigma_{k+t}^2) - \frac{1}{6\eta m_0} \|x_{k+m_0} - x_k\|^2$$

Bounding the Variance

Have bound on $\|x_k^s - x_0^s\|^2$ for each sub epoch

Recall $\mathbb{E}_{i_m}[\sigma_m^2] \leq L^2\|x_m - x_0\|^2$

Split $\|x_m - x_0\|^2$ for each subepoch and apply telescoping sum

Bounding the Variance

Bounded variance:

$$\mathbb{E}[\sigma_m^2] \leq L^2 d \mathbb{E}[6\eta m_0 (f(x_0) - f(x_m)) + 3\eta^2 \sum_{t=0}^{m-1} (\|\nabla_{k+t}\|^2 + \sigma_{k+t}^2)]$$

Holds for all iterates by simplification 3

Using Variance Bound

Have the variance bound, need lemma from gradient descent

Lemma 3.1:

If $x_{k+1} = x_k - \eta \tilde{\nabla}_k$ *with* $\eta \leq \frac{1}{L}$, *then*

$$f(x_k) - \mathbb{E}[f(x_{k+1})] \geq \frac{\eta}{2} \|\nabla f(x_k)\|^2 - \frac{\eta^2 L}{2} \mathbb{E}[\sigma_k^2]$$

Using Lemma 3.1

Telescope this over the whole epoch and use the variance bound

Using Lemma 3.1

Telescope this over the whole epoch and use the variance bound

$$f(x_0) - \mathbb{E}[f(x_m)] \geq \frac{\eta}{6}\mathbb{E}[\sum_{t=0}^{m-1} \|\nabla f(x_t)\|^2]$$

Average Iterate Bound

Have bound for one epoch, telescope over all epochs

Average Iterate Bound

Have bound for one epoch, telescope over all epochs

Average iterate bound:

$$\frac{1}{Sm} \sum_{s=1}^{S} \sum_{t=1}^{m-1} \mathbb{E}[\|\nabla f(x_t^s)\|^2] \leq \frac{6(f(x_0^1) - \min_x f(x))}{\eta Sm}$$

Expected Iterate Bound

We return a random iterate x_k^s with $k \in 1, \ldots, m$ and $s \in 1, \ldots S$

Expected Iterate Bound

We return a random iterate x_k^s with $k \in 1, \dots, m$ and $s \in 1, \dots S$

In expectation we have:

$$\mathbb{E}[\|\nabla f(x_k^s)\|^2] \leq \frac{6(f(x_0^1) - \min_x f(x))}{\eta S m}$$

Complexity

If we want a point x with $\mathbb{E}[\|\nabla f(x)\|^2] \leq \epsilon$

Complexity

If we want a point x with $\mathbb{E}[\|\nabla f(x)\|^2] \leq \epsilon$

Theorem 5.1:

$$\# \text{ iterations} = O\left(\frac{n^{2/3}L}{\epsilon} f(x_0 - f(x^*))\right)$$

Complexity

If we want a point x with $\mathbb{E}[\|\nabla f(x)\|^2] \leq \epsilon$

Theorem 5.1:

$$\#\text{ iterations} = O\left(\frac{n^{2/3}L}{\epsilon} f(x_0 - f(x^*))\right)$$

Factor of $n^{1/3}$ faster than GD

$O(\frac{1}{\epsilon})$ instead of $O(\frac{1}{\epsilon^2})$ for SGD

Full Non-convex SVRG Algorithm

Non-convex SVRG:

Initial estimate x_0^1

Outer epochs ($s = 1, \dots, S$)

- A snapshot vector x_0^s – Average estimate from pervious epoch
- $\tilde{\mu} = \nabla f(x_0^s)$

Inner iterations ($k = 0, \dots, m-1$)

- $i \in [n]$

$$x_{k+1}^s \leftarrow x_k^s - \eta \tilde{\nabla}$$
$$\tilde{\nabla}_k^s := \nabla f_i(x_k^s) - \nabla f_i(x_0^s) + \tilde{\mu}$$

- Randomly choose $m^s \in \{m, \dots, m - m_0 + 1\}$ with probability $\{\beta_{m_0-1}, \frac{10}{9}\beta_{m_0-1}, \frac{10}{9}(\beta_{m_0-1}+\beta_{m_0-2}), \dots, \frac{10}{9}(\beta_{m_0-1}+\dots+\beta_1)\}$
- $x_0^{s+1} \leftarrow x_{m^s}^s$

Return x_k^s for some $s \in [S]$ and $k \in [m]$

Computational Experiments

ERM with non-convex loss functions

- Accuracy experiment
- Running-time experiment

Neural net training error

ERM Accuracy

Train with different ℓ_2-regularized loss functions

Logistic loss, squared loss, smoothed hinge loss, sigmoid loss

Add outliers by flipping labels randomly

ERM Accuracy

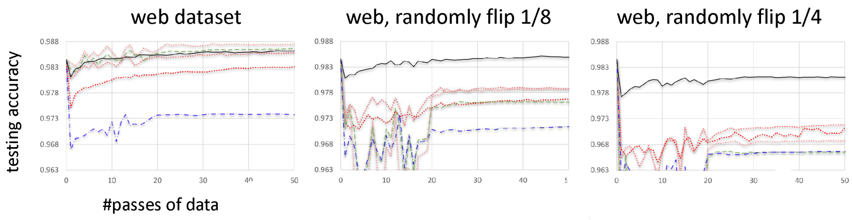


Figure: ERM Accuracy

Black lines: sigmoid loss, Blue lines: square loss,

Green lines: logistic loss, Red lines: hinge loss

ERM Accuracy – Results

As the number of outliers grows, sigmoid loss outperforms others

SVRG running time comparable, regardless of convexity of function

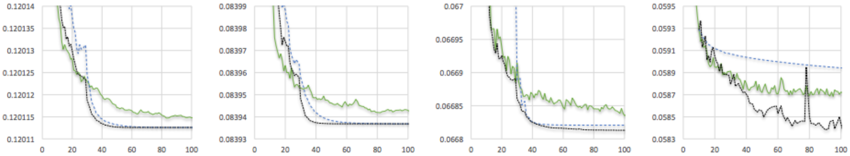
ERM Running-Time

Train with ℓ-2 regularized sigmoid loss

Compare running time of SGD and SVRG

Vary regularization parameter to "control" non-convexity

ERM Running-Time



(a) web, $\lambda = 10^{-3}$ (b) web, $\lambda = 10^{-4}$ (c) web, $\lambda = 10^{-5}$ (d) web, $\lambda = 10^{-6}$

Figure: ERM training error versus dataset passes

Green lines: Best tuned SGD

Blue lines: Constant step SVRG

Black lines: Best tuned SVRG

ERM Running-Time

SVRG better than SGD for small ϵ

SVRG performs better than SGD for small λ (more "nonconvex")

Neural Net Training Error

Compare SGD, SVRG and AdaGrad

Mini-batch size of 100 for all methods (except SVRG-4)

Compare different parameter settings for SVRG

Neural Net Training Error

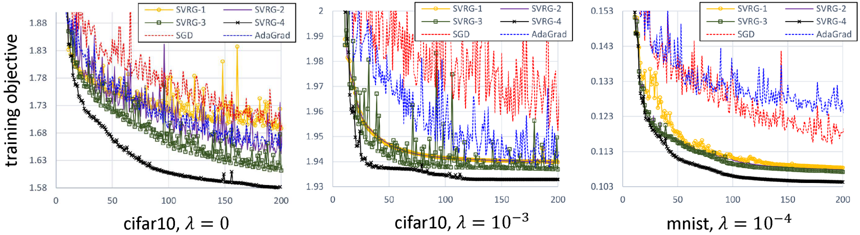


Figure: Neural Net Training Error

Blue lines: AdaGrad, Red lines: SGD, Yellow lines: SVRG-1

Purple lines: SVRG-2, Green lines: SVRG-3, **Black lines**: SVRG-4

Neural Net Training Error – Results

All forms of SVRG tend to outcompete SGD and AdaGrad on this dataset Each form of SVRG tends to outcompete one before it

- SVRG-1: simple Algorithm with tuned learning rate and batch size of 100
- SVRG-2: full Algorithm with tuned learning rate and batch size of 100
- SVRG-3: same as SVRG-2 with adaptive learning rate
- SVRG-4: same as SVRG-3 with batch size of 16

Questions

Any questions?