Variance Reduction on Adaptive Stochastic Mirror Descent

Wenjie Li, Zhanyu Wang, Yichen Zhang, Guang Cheng
{li3549,wang4094,zhang,chengg}@purdue.edu

Department of Statistics
Purdue University

December 3, 2020

Introduction

- Gradient Descent (GD) is stable but slow in terms of computing since it requires full-batch gradient.
- Stochastic Gradient Descent (SGD) is fast in terms of computing but slow in terms of convergence since it explores more region than needed due to the incessant noise.
- Johnson and Zhang used the [1] Variance Reduction (VR) method to speed up SGD by controlling the noise by mini-batch gradient.
- Can the idea of variance reduction be applied to more general optimization algorithms?

SVRG Algorithm

Key Idea: Use the large(full)-batch gradient g_t and the snapshot x_t to reduce the variance of mini-batch gradients.

Algorithm 1 Variance Reduction Algorithm

Input: Number of stages T, initial x_1, step sizes $\{\alpha_t\}_{t=1}^T$, batch sizes $\{B_t\}_{t=1}^T$, mini-batch sizes $\{b_t\}_{t=1}^T$
for $t = 1$ **to** T **do**
  Randomly sample a batch $\mathcal{I}_t$ with size B_t
  $g_t = \nabla f_{\mathcal{I}_t}(x_t),\ y_1^t = x_t$
  for $k = 1$ **to** K **do**
    Randomly sample a batch $\tilde{\mathcal{I}}_t$ of size b_t
    $v_k^t = \nabla f_{\tilde{\mathcal{I}}_t}(y_k^t) - \nabla f_{\tilde{\mathcal{I}}_t}(y_1^t) + g_t$
    $y_{k+1}^t = y_k^t - \alpha_t v_k^t$
  end for
  $x_{t+1} = y_{K+1}^t$
end for
Return $x_{t^*} = x_{T+1}$

SMD for Non-smooth Optimization

Algorithm 2 Adaptive Stochastic Mirror Descent (SMD) Algorithm

1: **Input:** Number of stages T, initial x_1, batch size b, step sizes $\{\alpha_t\}_{t=1}^T$
2: **for** $t = 1$ **to** T **do**
3: Randomly sample a batch $\mathcal{I}_t$ with size b
4: $g_t = \nabla f_{\mathcal{I}_t}(x_t)$
5: $x_{t+1} = \text{argmin}_x\{\alpha_t\langle g_t, x\rangle + \alpha_t h(x) + B_{\psi_t}(x, x_t)\}$
6: **end for**
7: **Return** Uniformly sample t^* from $\{t\}_{t=1}^T$ and ouput x_{t^*}

- Bregman divergence: $B_{\psi_t}(x, y) = \psi_t(x) - \psi_t(y) - \langle \nabla\psi_t(y), x - y\rangle$
- AdaGrad: let $S_t = \sum_{i=1}^t \text{diag}(g_i^2) + m$, $H_t = S_t^{1/2}$, $\psi_t(x) = \frac{1}{2}\langle x, H_t x\rangle$.

Theoretical Results

Let us consider a Non-convex Non-smooth Optimization Problem:

$$\min_{x\in\mathbb{R}^d} F(x) + h(x) \tag{1}$$

where $h(x)$ is a convex function that can be non-smooth, and $F(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$ with each $f_i(x)$ being a non-convex, L-smooth function.

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|, \ \forall i. \tag{2}$$

Their stochastic gradients are unbiased with bounded variance σ^2, i.e., sample $j \in \{1, 2, \cdots, n\}$ uniformly

$$\mathbb{E}[\nabla f_j(x)] = \nabla F(x), \ \mathbb{E}\|\nabla f_j(x) - \nabla F(x)\|_2^2 \leq \sigma^2. \tag{3}$$

Theoretical Results (cont.)

(Additional Assumption) The proximal functions $\psi_t(x)$ are all m-strongly convex with respect to $\|\cdot\|_2$, i.e.,

$$\psi_t(y) \geq \psi_t(x) + \langle \nabla\psi_t(x), y - x\rangle + \frac{m}{2}\|y - x\|_2^2, \forall t > 0$$

E.g.,

1. $\psi_t(x) = \phi_t(x) + \frac{c}{2}\|x\|_2^2, c > 0$, where each $\phi_t(x)$ is an arbitrary convex function
2. $\psi_t(x) = \frac{1}{2}\langle x, H_t x\rangle$ and $\exists c > 0$, s.t. $H_t \succeq cI, \forall t$

Theoretical Results (cont.)

Define the generalized stochastic gradient as $\tilde{g}_{X,t} = \frac{1}{\alpha_t}(x_t - x_{t+1})$.

The corresponding Convergence Criterion $\mathbb{E}[\|g_{X,t}\|_2^2] \leq \epsilon^2$ is defined with generalized gradient $g_{X,t}$, i.e., $\mathcal{I}_t$ being full dataset in $\tilde{g}_{X,t}$.

We measure the performance of algorithms using the number of calls to a Stochastic First-order Oracle (SFO), e.g., $\nabla f_i(x)$ for some i and x, needed to obtain a result satisfying our Convergence Criterion.

Adaptive SMD + VR

Algorithm 3 General Adaptive SMD with Variance Reduction Algorithm

Input: Number of stages T, initial x_1, step sizes $\{\alpha_t\}_{t=1}^T$, batch sizes $\{B_t\}_{t=1}^T$, mini-batch sizes $\{b_t\}_{t=1}^T$
for $t = 1$ **to** T **do**
  Randomly sample a batch $\mathcal{I}_t$ with size B_t
  $g_t = \nabla f_{\mathcal{I}_t}(x_t)$; $y_1^t = x_t$
  for $k = 1$ **to** K **do**
    Randomly pick sample $\tilde{\mathcal{I}}_t$ of size b_t
    $v_k^t = \nabla f_{\tilde{\mathcal{I}}_t}(y_k^t) - \nabla f_{\tilde{\mathcal{I}}_t}(y_1^t) + g_t$
    $y_{k+1}^t = \mathrm{argmin}_y\{\alpha_t\langle v_k^t, y\rangle + \alpha_t h(y) + B_{\psi_{tk}}(y, y_k^t)\}$
  end for
  $x_{t+1} = y_{K+1}^t$
end for
Return (Smooth case) Uniformly sample t^* from $\{t\}_{t=1}^T$ and output x_{t^*}; (P-L case) $x_{t^*} = x_{T+1}$

Theoretical Performance Comparison

Table 1: Performance Comparison Between Different Algorithms

Algorithms	SFO computations
GD	$O(n/\epsilon^2)$
SGD	$O(1/\epsilon^4)$
SVRG [2]	$O(n^{2/3}/\epsilon^2)$
SCSG [3]	$O(n/\epsilon^2 \wedge 1/\epsilon^{10/3})$
SNVRG [4]	$\tilde{O}(n^{1/2}/\epsilon^2 \wedge 1/\epsilon^3)$
ProxGD [5]	$O(n/\epsilon^2)$
ProxSVRG/SAGA [6]	$O(n/(\epsilon^2\sqrt{b}) + n)$
ProxSVRG+ [7]	$O(n/(\epsilon^2\sqrt{b}) \wedge (1/(\epsilon^4\sqrt{b})) + b/\epsilon^2)$
Adaptive SMD	$O(n/\epsilon^2 \wedge 1/\epsilon^4)$
Adaptive SMD + VR	$O(n/(\epsilon^2\sqrt{b}) \wedge 1/(\epsilon^4\sqrt{b}) + b/\epsilon^2)$

n is the total number of samples. b is the mini-batch size. ϵ is defined in our convergence criterion.

Theoretical Performance Comparison (cont.)

Corollary

With all the assumptions and parameter settings in the main theorem, further assume that $b = \epsilon^{-4/3}$, *where* $\epsilon^{-4/3} \leq n$. *Then the output of algorithm 3 converges with gradient computations*

$$O(\frac{n}{\epsilon^{4/3}} \wedge \frac{1}{\epsilon^{10/3}} + \frac{1}{\epsilon^{10/3}}) \tag{4}$$

Results Inferred

When we take the proximal function to be $\psi_{tk}(x) = \frac{c_{tk}}{2}\|x\|_2^2, c_{tk} \geq m$, Algorithm 3 reduces to ProxSVRG+ with time-varying effective step size α_t/c_{tk}. As long as the effective step sizes α_t/c_{tk} are upper bounded by $(m/L)/m = 1/L$, Algorithm 3 still convergences with the same complexity.

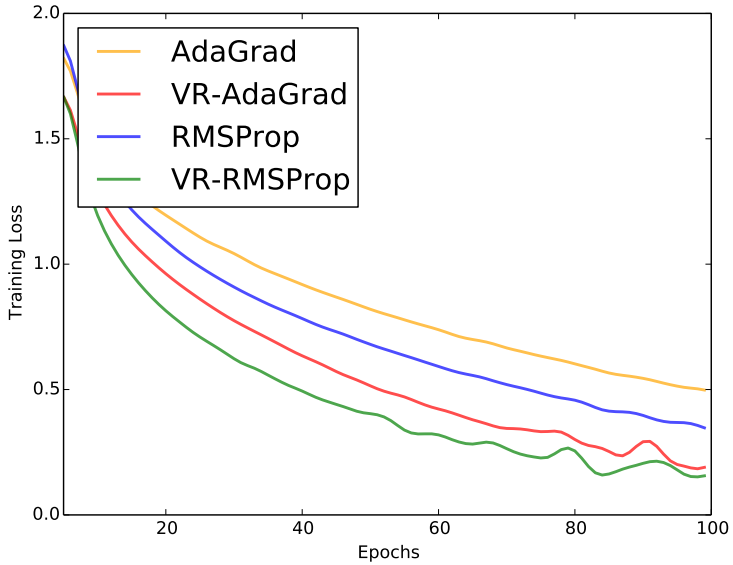
Besides, ψ_{tk} can be more complicated, such as $\psi_{tk}(x) = \phi_{tk}(x) + \frac{c}{2}\|x\|_2^2, c > 0$, where each $\phi_{tk}(x)$ is an arbitrary convex function, or $\psi_{tk}(x) = \frac{1}{2}\langle x, H_{tk}x\rangle$ as in adaptive algorithms.

Results Inferred (cont.)

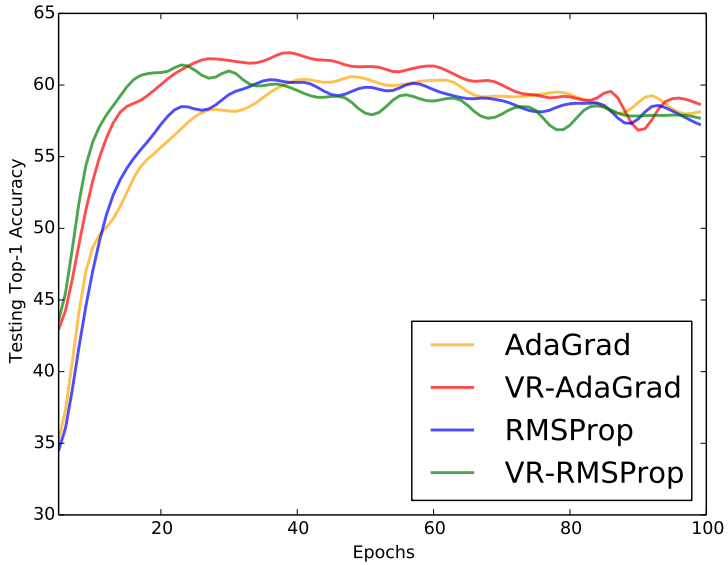
Another interesting result observed in our theorem is that when m is small, we require relatively larger batch sizes B_t to guarantee the fast convergence. This claim is supported by our experiments with adaptive algorithms.

Experimental Performance Comparison

AdaGrad and RMSProp



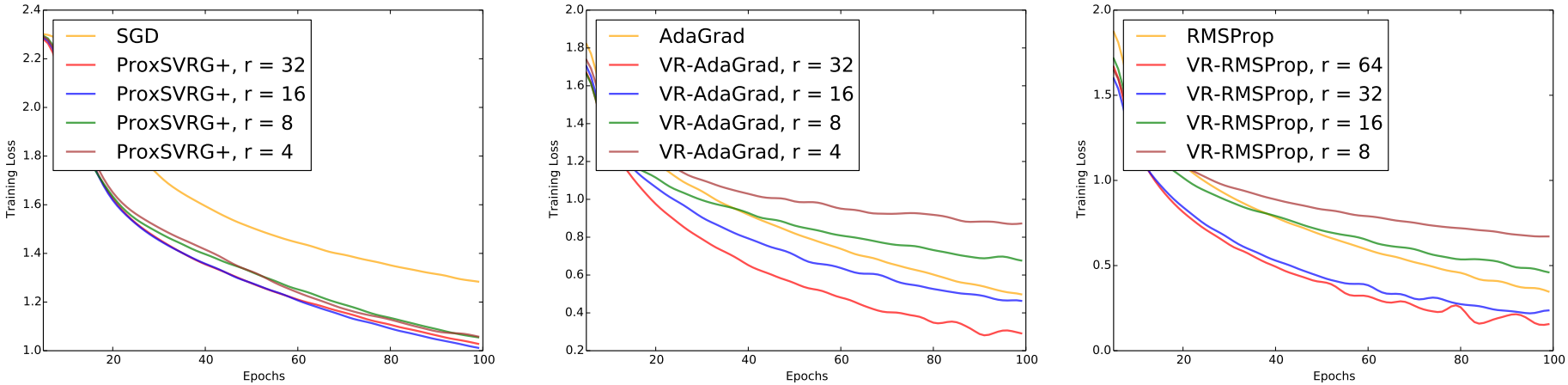
(a) CIFAR-10 Training Loss.



(b) CIFAR-10 Testing Acc.

Figure 1: Training Loss and Testing Acc on CIFAR-10. Variance reduction does make the convergence of AdaGrad and RMSProp faster.

Experimental Performance Comparison (cont.)



(a) ProxSVRG+ (b) VR-AdaGrad (c) VR-RMSProp

Figure 2: We used the notation of batch size ratio $r = B_t/b_t$. Note that AdaGrad and RMSProp need a much larger batch size ratio to become faster than the original algorithm. However, ProxSVRG+ works with $r = 4$

Future Directions

1. Can we combine the analysis with the tricks in SNVRG or SPIDER to further fasten the convergence?
2. Can the specific VR-RMSProp and VR-AdaGrad algorithms converge faster than ProxSVRG+?

Bibliography

[1] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," *Advances in Neural Information Processing Systems 27*, 2013.

[2] S. J. Reddi, A. Hefny, S. Sra, B. Poczos, and A. Smola, *Stochastic variance reduction for nonconvex optimization*, 2016a. arXiv: 1603.06160 [math.OC].

[3] L. Lei, C. Ju, J. Chen, and M. I. Jordan, "Non-convex finite-sum optimization via scsg methods," *Advances in Neural Information Processing Systems 30*, pp. 2348–2358, 2017.

[4] D. Zhou, P. Xu, and Q. Gu, "Stochastic nested variance reduction for nonconvex optimization," *Advances in Neural Information Processing Systems 32*, 2018.

[5] S. Ghadimi, G. Lan, and H. Zhang, "Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization," *arXiv preprint arXiv:1308.6594*, 2016.

[6] S. Reddi, S. Sra, B. Poczos, and A. J. Smola, "Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization," *Advances in Neural Information Processing Systems 29*, 2016b.

Bibliography (cont.)

[7] Z. Li and J. Li, "A simple proximal stochastic gradient method for nonsmooth nonconvex optimization," *Advances in Neural Information Processing Systems 31*, pp. 5564–5574, 2018.