MOMENTUM-BASED VARIANCE REDUCTION IN NON-CONVEX SGD

VIDITHA WUDARU

50442670

CONTENTS

Abstract

- Variance reduction technique typically requires - carefully tuned learning rates and use of “mega-batches” in order to achieve improved results.
- STORM - Does not require any batches and uses adaptive learning rate.
- Enables simpler implementation.
- Less hyper parameter tuning.
- Our technique for removing the batches uses a variant of momentum to achieve variance reduction in non-convex optimization.

CONTENTS

Introduction

- Classic stochastic optimization problem, in which we are given a function $F : R^d \to R$, and wish to find $x \in R^d$ such that $F(x)$ is as small as possible.
- We can obtain sample functions $f(\cdot, \xi)$ where ξ represents some sample variable (e.g. a mini-batch index) such that $E[f(\cdot, \xi)] = F(\cdot)$.
- SGD produces a sequence of iterates $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ using the recursion

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \boldsymbol{g}_t,$$

- where $\boldsymbol{g}_t = \nabla f(\boldsymbol{x}_t, \xi_t)$, $f(\cdot, \xi_1), \ldots, f(\cdot, \xi_T)$ are i.i.d. samples from a distribution D, and $\eta_1, \ldots \eta_T \in \mathbb{R}$ are a sequence of learning rates that must be carefully tuned to ensure good performance.

Introduction

- SVRG
  - $\boldsymbol{g}_t$ is a *variance reduced* estimate of $\nabla F(\boldsymbol{x}_t)$
- SVRG algorithms have improved the convergence rate to critical points of non-convex SGD from $O(1/T^{1/4})$ to $O(1/T^{3/10})$ to $O(1/T^{1/3})$.
- Despite this improvement, SVRG has not seen as much success in practice in non-convex machine learning problems.
- Two potential issues - Use of non-adaptive learning rates and reliance on giant batch sizes.

Introduction

- STORM - STochastic Recursive Momentum.
- Achieves variance reduction through the use of a variant of the momentum term.
- Hence, our algorithm does not require a gigantic batch to compute.
- Storm achieves the optimal convergence rate of $O(1/T^{1/3})$, and it uses an adaptive learning rate schedule that will automatically adjust to the variance values of $\nabla f(x_t, \xi_t)$.

CONTENTS

Related Work

- Johnson and Zhang, Zhang et al., Mahdavi et al., and Wang et al. started working in 3 groups.
- Achieved much better convergence rates for critical points in non-convex SGD.
- These are different from previous because of calculation of gradient at checkpoints.

CONTENTS

Momentum and Variance Reduction

- The stochastic gradient descent with momentum algorithm is typically implemented as
  - $d_t = (1 - a)d_{t-1} + a\nabla f(x_t, \xi_t)$
  - $x_{t+1} = x_t - \eta d_t$
- A variant of momentum can provably reduce the variance of the gradients.
  - $d_t = (1 - a)d_{t-1} + a\nabla f(x_t, \xi_t) + (1 - a)(\nabla f(x_t, \xi_t) - \nabla f(x_{t-1}, \xi_t))$
  - $x_{t+1} = x_t - \eta d_t$

CONTENTS

STORM Algorithm

Algorithm 1 STORM: STOchastic Recursive Momentum

Input: Parameters k, w, c, initial point $\boldsymbol{x}_1$
Sample ξ_1
$G_1 \leftarrow \|\nabla f(\boldsymbol{x}_1, \xi_1)\|$
$\boldsymbol{d}_1 \leftarrow \nabla f(\boldsymbol{x}_1, \xi_1)$
$\eta_0 \leftarrow \frac{k}{w^{1/3}}$
for $t = 1$ **to** T **do**
  $\eta_t \leftarrow \frac{k}{(w+\sum_{i=1}^{t} G_t^2)^{1/3}}$
  $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t - \eta_t \boldsymbol{d}_t$
  $a_{t+1} \leftarrow c\eta_t^2$
  Sample ξ_{t+1}
  $G_{t+1} \leftarrow \|\nabla f(\boldsymbol{x}_{t+1}, \xi_{t+1})\|$
  $\boldsymbol{d}_{t+1} \leftarrow \nabla f(\boldsymbol{x}_{t+1}, \xi_{t+1}) + (1 - a_{t+1})(\boldsymbol{d}_t - \nabla f(\boldsymbol{x}_t, \xi_{t+1}))$
end for
Choose $\hat{\boldsymbol{x}}$ uniformly at random from $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$. (In practice, set $\hat{\boldsymbol{x}} = \boldsymbol{x}_T$).
return $\hat{\boldsymbol{x}}$

CONTENTS

Theorem

Theorem 1. *Under the assumptions in Section 3, for any* $b > 0$*, we write* $k = \frac{bG^{\frac{2}{3}}}{L}$*. Set* $c = 28L^2 + G^2/(7Lk^3) = L^2(28 + 1/(7b^3))$ *and* $w = \max\left((4Lk)^3, 2G^2, \left(\frac{ck}{4L}\right)^3\right) = G^2 \max\left((4b)^3, 2, (28b + \frac{1}{7b^2})^3/64\right)$*. Then,* STORM *satisfies*

$$\mathbb{E}\left[\|\nabla F(\hat{\boldsymbol{x}})\|\right] = \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla F(\boldsymbol{x}_t)\|\right] \leq \frac{w^{1/6}\sqrt{2M} + 2M^{3/4}}{\sqrt{T}} + \frac{2\sigma^{1/3}}{T^{1/3}},$$

where $M = \frac{8}{k}(F(\boldsymbol{x}_1) - F^\star) + \frac{w^{1/3}\sigma^2}{4L^2k^2} + \frac{k^2c^2}{2L^2}\ln(T+2)$.

Theorem

In words, Theorem 1 guarantees that STORM will make the norm of the gradients converge to 0 at a rate of $O(\frac{\ln T}{\sqrt{T}})$ if there is no noise, and in expectation at a rate of $\frac{2\sigma^{1/3}}{T^{1/3}}$ in the stochastic case. We remark that we achieve both rates automatically, without the need to know the noise level nor the need to tune stepsizes. Note that the rate when $\sigma \neq 0$ matches the optimal rate [3], which was previously only obtained by SVRG-based algorithms that require a "mega-batch" [8, 31].

Proof of Theorem

Lemma 1. *Suppose $\eta_t \leq \frac{1}{4L}$ for all t. Then*

$$\mathbb{E}[F(\boldsymbol{x}_{t+1}) - F(\boldsymbol{x}_t)] \leq \mathbb{E}\left[-\eta_t/4\|\nabla F(\boldsymbol{x}_t)\|^2 + 3\eta_t/4\|\boldsymbol{\epsilon}_t\|^2\right] .$$

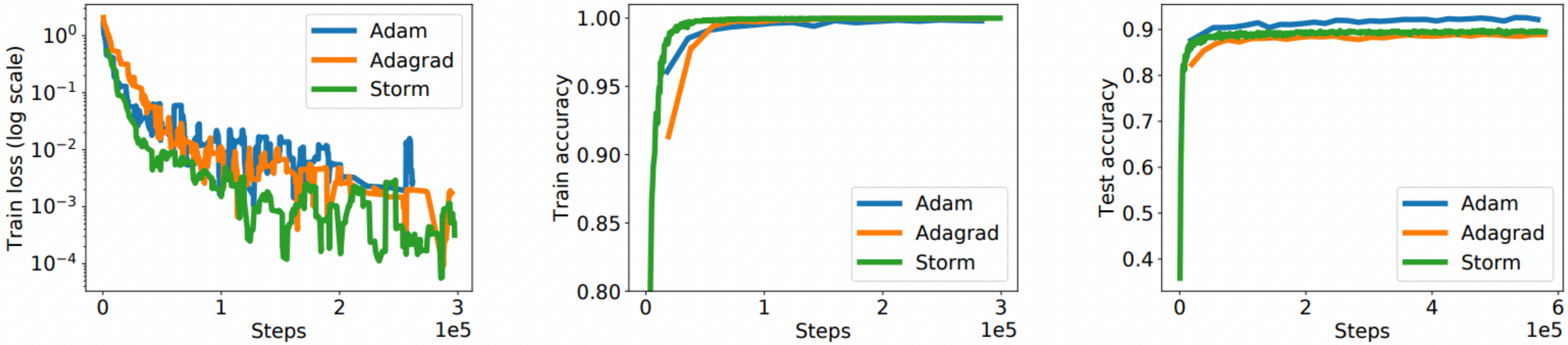
Proof. Using the smoothness of F and the definition of $\boldsymbol{x}_{t+1}$ from the algorithm, we have

$$\begin{aligned}
\mathbb{E}[F(\boldsymbol{x}_{t+1})] &\leq \mathbb{E}\left[F(\boldsymbol{x}_t) - \nabla F(\boldsymbol{x}_t)\cdot \eta_t \boldsymbol{d}_t + \frac{L\eta_t^2}{2}\|\boldsymbol{d}_t\|^2\right] \\
&= \mathbb{E}\left[F(\boldsymbol{x}_t) - \eta_t\|\nabla F(\boldsymbol{x}_t)\|^2 - \eta_t \nabla F(\boldsymbol{x}_t)\cdot \boldsymbol{\epsilon}_t + \frac{L\eta_t^2}{2}\|\boldsymbol{d}_t\|^2\right] \\
&\leq \mathbb{E}\left[F(\boldsymbol{x}_t) - \frac{\eta_t}{2}\|\nabla F(\boldsymbol{x}_t)\|^2 + \frac{\eta_t}{2}\|\boldsymbol{\epsilon}_t\|^2 + \frac{L\eta_t^2}{2}\|\boldsymbol{d}_t\|^2\right] \\
&\leq \mathbb{E}\left[F(\boldsymbol{x}_t) - \frac{\eta_t}{2}\|\nabla F(\boldsymbol{x}_t)\|^2 + \frac{\eta_t}{2}\|\boldsymbol{\epsilon}_t\|^2 + L\eta_t^2\|\boldsymbol{\epsilon}_t\|^2 + L\eta_t^2\|\nabla F(\boldsymbol{x}_t)\|^2\right] \\
&\leq \mathbb{E}\left[F(\boldsymbol{x}_t) - \frac{\eta_t}{2}\|\nabla F(\boldsymbol{x}_t)\|^2 + \frac{3\eta_t}{4}\|\boldsymbol{\epsilon}_t\|^2 + \frac{\eta_t}{4}\|\nabla F(\boldsymbol{x}_t)\|^2\right] ,
\end{aligned}$$

where in the second inequality we used Young's inequality, the third one uses $\|\boldsymbol{x}+\boldsymbol{y}\|^2 \leq 2\|\boldsymbol{x}\|^2 + 2\|\boldsymbol{y}\|^2$, and the last one uses $\eta_t \leq 1/4L$. $\square$

CONTENTS

Validation - Experiments on CIFAR-10 with ResNet-32 Network



(a) Train Loss vs Iterations

(b) Train Accuracy vs Iterations

(c) Test Accuracy vs Iterations

CONTENTS

Conclusion

- STORM finds critical points in stochastic, smooth, non-convex problems.
- Removes the need for batch-size, and incorporates adaptive learning rates.
- Storm is substantially easier to tune.
- CIFAR-10 with a ResNet architecture, Storm indeed seems to be optimizing the objective in fewer iterations than baseline algorithms

CONTENTS

References

- https://proceedings.neurips.cc/paper/2019/hash/b8002139cdde66b87638f7f91d169d96-Abstract.html
- Z. Allen-Zhu and E. Hazan. Variance reduction for faster non-convex optimization. In International conference on machine learning, pages 699–707, 2016.
- S. Ghadimi and G. Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. SIAM Journal on Optimization, 23(4):2341–2368, 2013.

THANK YOU