AdaGrad Stepsizes: Sharp Convergence Over Nonconvex Landscapes

Xiaoxia(Shirley) WU⋆

PhD Candidate, The University of Texas at Austin

June 11th, 2019

⋆ joint work with Rachel Ward and Léon Bottou, at Facebook AI Research.

Outline

Motivations

Theoretical Contributions

We provide a novel convergence result for AdaGrad-Norm to emphasize its robustness to the hyper-parameter tuning over nonconvex landscapes.

Practical Implications

Outline

Motivations

Theoretical Contributions

We provide a novel convergence result for AdaGrad-Norm to emphasize its robustness to the hyper-parameter tuning over nonconvex landscapes.

Practical Implications

Motivation

Problem Setup

Given a differentiable **non-convex** function, $F : \mathbb{R}^d \to \mathbb{R}$,

- $\|\nabla F(x) - \nabla F(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^d$

Motivation

Problem Setup

Given a differentiable **non-convex** function, $F : \mathbb{R}^d \to \mathbb{R}$,

- $\|\nabla F(x) - \nabla F(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^d$

$$\textbf{Our desired goal} \quad \Rightarrow \quad \min_{x \in \mathbb{R}^d} F(x)$$

$$\textbf{We can achieve} \quad \Rightarrow \quad \|\nabla F(x)\|^2 \leq \varepsilon$$

Motivation

Problem Setup

Given a differentiable **non-convex** function, $F : \mathbb{R}^d \to \mathbb{R}$,

- $\|\nabla F(x) - \nabla F(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^d$

$$\textbf{Our desired goal} \quad \Rightarrow \quad \min_{x \in \mathbb{R}^d} F(x)$$

$$\textbf{We can achieve} \quad \Rightarrow \quad \|\nabla F(x)\|^2 \leq \varepsilon$$

Algorithm

Stochastic Gradient Descent (SGD) at the jth iteration

$$x_{j+1} \leftarrow x_j - \eta_j G(x_j), \tag{1}$$

where $\mathbb{E}[G(x_j)] = \nabla F(x_j)$ and $\eta_j > 0$ is the **stepsize**.

Motivation

Algorithm: SGD

Set a sequence $\{\eta_j\}_{j\geq 0}$ for

$$x_{j+1} \leftarrow x_j - \eta_j G(x_j)$$

Q: How to set the sequence $\{\eta_j\}_{j\geq 0}$?

[1] $\mathbb{E}[\|G(x) - \nabla F(x)\|^2] \leq \sigma^2$

Motivation

Algorithm: SGD

Set a sequence $\{\eta_j\}_{j\geq 0}$ for

$$x_{j+1} \leftarrow x_j - \eta_j G(x_j)$$

Q: How to set the sequence $\{\eta_j\}_{j\geq 0}$?

Difficulty in Choosing Stepsizes

The classical Robbins/Monro theory (Robbins and Monro, 1951) if

$$\sum_{j=1}^{\infty} \eta_j = \infty \quad \text{and} \quad \sum_{j=1}^{\infty} \eta_j^2 < \infty; \tag{2}$$

and the variance of the gradient is bounded [1], then

$$\lim_{j\to\infty} \mathbb{E}[\|\nabla F(x_j)\|^2] = 0.$$

---

[1] $\mathbb{E}[\|G(x) - \nabla F(x)\|^2] \leq \sigma^2$

Motivation

Algorithm: SGD

Set a sequence $\{\eta_j\}_{j\geq 0}$ for

$$x_{j+1} \leftarrow x_j - \eta_j G(x_j)$$

Q: How to set the sequence $\{\eta_j\}_{j\geq 0}$?

Difficulty in Choosing Stepsizes

The classical Robbins/Monro theory (Robbins and Monro, 1951) if

$$\sum_{j=1}^{\infty} \eta_j = \infty \quad \text{and} \quad \sum_{j=1}^{\infty} \eta_j^2 < \infty; \tag{3}$$

and the variance of the gradient is bounded, then
$\lim_{j\to\infty} \mathbb{E}[\|\nabla F(x_j)\|^2] = 0.$

However, the rule is too general for practical applications.

Motivation

Algorithm: SGD

Set a sequence $\{\eta_j\}_{j\geq 0}$ for

$$x_{j+1} \leftarrow x_j - \eta_j G(x_j)$$

Possible Choice: Manual Tuning

$$\eta_j = \begin{cases} \eta & j \leq T_1 \\ \alpha_1 \eta & T_1 \leq j \leq T_2 \\ \alpha_2 \eta & T_2 \leq j \leq T_3 \\ \dots \end{cases}$$

[2] $\|\nabla F(x) - \nabla F(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^d$

Motivation

Algorithm: SGD

Set a sequence $\{\eta_j\}_{j \geq 0}$ for

$$x_{j+1} \leftarrow x_j - \eta_j G(x_j)$$

Possible Choice: Manual Tuning

$$\eta_j = \begin{cases} \eta & j \leq T_1 \\ \alpha_1 \eta & T_1 \leq j \leq T_2 \\ \alpha_2 \eta & T_2 \leq j \leq T_3 \\ \dots & \end{cases}$$

However, tuning η, α_1, α_2, T_1, T_2, $\dots$ are computationally costly.
In particular, it requires $\eta \leq 2/L$.[2]

[2] $\|\nabla F(x) - \nabla F(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^d$

Motivation

Algorithm: SGD with Adaptive Stepsize

Set a sequence $\{b_j\}_{j \geq 0}$ for $\ell = 1, 2, \cdots, d$

$$[x_{j+1}]_\ell \leftarrow [x_j]_\ell - \frac{\eta}{[b_{j+1}]_\ell}[G(x_j)]_\ell$$

Possible Choice: Adaptive Gradient Methods

Among many variants, one is *AdaGrad*

$$([b_{j+1}]_\ell)^2 = ([b_j]_\ell)^2 + ([G(x_j)]_\ell)^2$$

Motivation

Algorithm: SGD with Adaptive Stepsize

Set a sequence $\{b_j\}_{j\geq 0}$ for $\ell = 1, 2, \cdots, d$

$$[x_{j+1}]_\ell \leftarrow [x_j]_\ell - \frac{\eta}{[b_{j+1}]_\ell}[G(x_j)]_\ell$$

Possible Choice: Adaptive Gradient Methods

Among many variants, one is *AdaGrad*

$$([b_{j+1}]_\ell)^2 = ([b_j]_\ell)^2 + ([G(x_j)]_\ell)^2$$

- It helps with "increasing the stepsize for more sparse parameters and decreasing the stepsize for less sparse ones." (Duchi et al. 2011)

Motivation

Algorithm: SGD with Adaptive Stepsize

Set a sequence $\{b_j\}_{j\geq 0}$ for $\ell = 1, 2, \cdots, d$

$$[x_{j+1}]_\ell \leftarrow [x_j]_\ell - \frac{\eta}{[b_{j+1}]_\ell}[G(x_j)]_\ell$$

Possible Choice: Adaptive Gradient Methods

Among many variants, one is *AdaGrad*

$$([b_{j+1}]_\ell)^2 = ([b_j]_\ell)^2 + ([G(x_j)]_\ell)^2$$

- It helps with "increasing the stepsize for more sparse parameters and decreasing the stepsize for less sparse ones." (Duchi et al. 2011)
- However, "co-ordinate" AdaGrad changes the optimization problem by introducing the "bias" in the solutions, leading to worse generalization (Wilson et al. 2017)

Motivation

Algorithm: SGD with Adaptive Stepsize

Set a sequence $\{b_j\}_{j\geq 0}$ for $\ell = 1, 2, \cdots, d$

$$[x_{j+1}]_\ell \leftarrow [x_j]_\ell - \frac{\eta}{b_{j+1}}[G(x_j)]_\ell$$

Possible Variant: Norm Version of AdaGrad

$$\text{(AdaGrad-Norm)} \qquad b_{j+1}^2 = b_j^2 + \|G(x_j)\|^2$$

Motivation

Algorithm: SGD with Adaptive Stepsize

Set a sequence $\{b_j\}_{j\geq 0}$ for $\ell = 1, 2, \cdots, d$

$$[x_{j+1}]_\ell \leftarrow [x_j]_\ell - \frac{\eta}{b_{j+1}}[G(x_j)]_\ell$$

Possible Variant: Norm Version of AdaGrad

$$\text{(AdaGrad-Norm)} \qquad b_{j+1}^2 = b_j^2 + \|G(x_j)\|^2$$

- Auto-tuning property (Wu, Ward, and Bottou, 2018): robustness to the choices of hyper-parameters (b_0 and η); connection to Weight/Layer/Batch Normalization;
- Does not affect generalization.

Outline

Motivations

Theoretical Contributions

We provide a novel convergence result for AdaGrad-Norm to emphasize its robustness to the hyper-parameter tuning over nonconvex landscapes.

Practical Implications

Theory

Algorithm: SGD with Adaptive Stepsize

$$x_{j+1} \leftarrow x_j - \frac{\eta}{b_{j+1}} G(x_j) \quad \text{with} \quad b_{j+1}^2 = b_j^2 + \|G(x_j)\|^2$$

What is the convergence rate of AdaGrad-Norm?

- Intuition: if $\mathbb{E}[\|G(x_j)\|^2] \leq \gamma^2$, then the **effective stepsize** $\frac{\eta}{b_j}$

$$\mathbb{E}\left[\frac{\eta}{b_j}\right] \geq \frac{\eta}{\sqrt{j\gamma^2 + b_0^2}}$$

Theory

Algorithm: SGD with Adaptive Stepsize

$$x_{j+1} \leftarrow x_j - \frac{\eta}{b_{j+1}} G(x_j) \quad \text{with} \quad b_{j+1}^2 = b_j^2 + \|G(x_j)\|^2$$

What is the convergence rate of AdaGrad-Norm?

- Intuition: if $\mathbb{E}[\|G(x_j)\|^2] \leq \gamma^2$, then the **effective stepsize** $\frac{\eta}{b_j}$

$$\mathbb{E}\left[\frac{\eta}{b_j}\right] \geq \frac{\eta}{\sqrt{j\gamma^2 + b_0^2}}$$

- Convex Landscapes $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ (Levy, 2018)

Theory

Algorithm: SGD with Adaptive Stepsize

$$x_{j+1} \leftarrow x_j - \frac{\eta}{b_{j+1}} G(x_j) \quad \text{with} \quad b_{j+1}^2 = b_j^2 + \|G(x_j)\|^2$$

What is the convergence rate of AdaGrad-Norm?

- Intuition: if $\mathbb{E}[\|G(x_j)\|^2] \leq \gamma^2$, then the **effective stepsize** $\frac{\eta}{b_j}$

$$\mathbb{E}\left[\frac{\eta}{b_j}\right] \geq \frac{\eta}{\sqrt{j\gamma^2 + b_0^2}}$$

- Convex Landscapes $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ (Levy, 2018)
- **Nonconvex Landscapes $\mathcal{O}\left(\frac{\log(T)}{\sqrt{T}}\right)$ (Ours, Theorem 2.1)**

Theory

Algorithm: SGD with Adaptive Stepsize

(1) At jth iteration, generate ξ_j and $G(x_j) = G(x_j, \xi_j)$

(2) $x_{j+1} \leftarrow x_j - \frac{\eta}{b_{j+1}} G(x_j)$ with $b_{j+1}^2 = b_j^2 + \|G(x_j)\|^2$

Theorem

Under the assumption:

1. *The random vectors $\xi_j, j = 0, 1, 2, \ldots,$ are mutually independent and also independent of x_j;*
2. *Bounded variance[3]: $\mathbb{E}_{\xi_j}[\|G(x_j, \xi_j) - \nabla F(x_j)\|^2] \leq \sigma^2$;*
3. *Bounded gradient norm: $\|\nabla F(x_j)\| \leq \gamma$ uniformly;*

[3] It means the expectation with respect to ξ_j conditional on x_j.

Theory

Algorithm: SGD with Adaptive Stepsize

(1) At jth iteration, generate ξ_j and $G(x_j) = G(x_j, \xi_j)$

(2) $x_{j+1} \leftarrow x_j - \frac{\eta}{b_{j+1}} G(x_j)$ with $b_{j+1}^2 = b_j^2 + \|G(x_j)\|^2$

Theorem

Under the assumption:

1. *The random vectors* $\xi_j, j = 0, 1, 2, \ldots,$ *are mutually independent and also independent of* x_j*;*
2. *Bounded variance*[3]*:* $\mathbb{E}_{\xi_j}[\|G(x_j, \xi_j) - \nabla F(x_j)\|^2] \le \sigma^2$*;*
3. *Bounded gradient norm:* $\|\nabla F(x_j)\| \le \gamma$ *uniformly;*

AdaGrad-Norm converges to a stationary point w.h.p. at the rate

$$\min_{\ell=0,1,\ldots,T-1} \|\nabla F(x_\ell)\|^2 \le \frac{C^2}{T} + \frac{\sigma C}{\sqrt{T}}$$

where $C = \widetilde{\mathcal{O}}(\log(T/b_0 + 1))$ *and* $\widetilde{\mathcal{O}}$ *hides* η, L *and* $F(x_0) - F^*$*.*

[3] It means the expectation with respect to ξ_j conditional on x_j.

Theory

Algorithm: SGD with Adaptive Stepsize

(1) At jth iteration, generate ξ_j and $G(x_j) = G(x_j, \xi_j)$

(2) $x_{j+1} \leftarrow x_j - \frac{\eta}{b_{j+1}} G(x_j)$ with $b_{j+1}^2 = b_j^2 + \|G(x_j)\|^2$

Challenges in the proof:

b_{j+1} **is a random variable correlated with** $\nabla F(x_j)$ **and** $G(x_j)$

- L-Lipschitz continuous gradient [4]

$$\frac{F_{j+1} - F_j}{\eta} \leq -\frac{\|\nabla F_j\|^2}{b_{j+1}} + \underbrace{\frac{\langle \nabla F_j, \nabla F_j - G_j \rangle}{b_{j+1}}}_{KeyTerm} + \frac{\eta L \|G_j\|^2}{2 b_{j+1}^2}.$$

[4] We write $F(x_j) = F_j$, $\nabla F(x_j) = \nabla F_j$ and $G(x_j) = G_j$.

Theory

Algorithm: SGD with Adaptive Stepsize

(1) At jth iteration, generate ξ_j and $G(x_j) = G(x_j, \xi_j)$

(2) $x_{j+1} \leftarrow x_j - \frac{\eta}{b_{j+1}} G(x_j)$ with $b_{j+1}^2 = b_j^2 + \|G(x_j)\|^2$

Challenges in the proof:

b_{j+1} **is a random variable correlated with** $\nabla F(x_j)$ **and** $G(x_j)$

- L-Lipschitz continuous gradient [4]

$$\frac{F_{j+1} - F_j}{\eta} \leq -\frac{\|\nabla F_j\|^2}{b_{j+1}} + \underbrace{\frac{\langle \nabla F_j, \nabla F_j - G_j \rangle}{b_{j+1}}}_{KeyTerm} + \frac{\eta L \|G_j\|^2}{2 b_{j+1}^2}.$$

- Unlike the standard SGD with constant stepsize

$$\mathbb{E}_{\xi_j}\left[\frac{\langle \nabla F_j, \nabla F_j - G_j \rangle}{b_{j+1}}\right] \neq 0;$$

[4] We write $F(x_j) = F_j$, $\nabla F(x_j) = \nabla F_j$ and $G(x_j) = G_j$.

Theory

Algorithm: SGD with Adaptive Stepsize

(1) At jth iteration, generate ξ_j and $G(x_j) = G(x_j, \xi_j)$

(2) $x_{j+1} \leftarrow x_j - \frac{\eta}{b_{j+1}} G(x_j)$ with $b_{j+1}^2 = b_j^2 + \|G(x_j)\|^2$

Challenges in the proof:

b_{j+1} **is a random variable correlated with** $\nabla F(x_j)$ **and** $G(x_j)$

- L-Lipschitz continuous gradient [4]

$$\frac{F_{j+1} - F_j}{\eta} \leq -\frac{\|\nabla F_j\|^2}{b_{j+1}} + \underbrace{\frac{\langle \nabla F_j, \nabla F_j - G_j \rangle}{b_{j+1}}}_{KeyTerm} + \frac{\eta L \|G_j\|^2}{2 b_{j+1}^2}.$$

- Unlike the standard SGD with constant stepsize

$$\mathbb{E}_{\xi_j}\left[\frac{\langle \nabla F_j, \nabla F_j - G_j \rangle}{b_{j+1}}\right] \neq 0;$$

- New techniques needed to bound *KeyTerm*:
  careful Tower rule, Cauchy-Schwarz, Hölder's Inequality, etc.

---

[4] We write $F(x_j) = F_j$, $\nabla F(x_j) = \nabla F_j$ and $G(x_j) = G_j$.

Outline

Motivations

Theoretical Contributions

We provide a novel convergence result for AdaGrad-Norm to emphasize its robustness to the hyper-parameter tuning over nonconvex landscapes.

Practical Implications

Practice

AdaGrad-Norm

We show that AdaGrad-Norm converges [5]

$$\min_{\ell=0,1,\ldots,T-1} \|\nabla F(x_\ell)\|^2 \leq \mathcal{O}\left(\frac{C_1}{T} + \frac{\sigma C_2}{\sqrt{T}}\right)$$

where the constants C_1 and C_2 are explicit and **robust to hyper-parameters b_0 and η**.

Recall: $\mathbb{E}_{\xi_j}[\|G(x_j,\xi_j) - \nabla F(x_j)\|^2] \leq \sigma^2$

[5]Note we combine Theorem 2.1 and Theorem 2.2

[6]For the case $b_1 \geq \eta L \approx \Delta L$

Practice

AdaGrad-Norm

We show that AdaGrad-Norm converges [5]

$$\min_{\ell=0,1,\ldots,T-1} \|\nabla F(x_\ell)\|^2 \leq \mathcal{O}\left(\frac{C_1}{T} + \frac{\sigma C_2}{\sqrt{T}}\right)$$

where the constants C_1 and C_2 are explicit and **robust to hyper-parameters b_0 and η.**

Recall: $\mathbb{E}_{\xi_j}[\|G(x_j,\xi_j) - \nabla F(x_j)\|^2] \leq \sigma^2$

- For $\sigma \approx 0$
  Suppose we know F^* and set $\eta = F(x_0) - F^*$; the constant C_1 almost matches GD with best stepsize. [6]

[5] Note we combine Theorem 2.1 and Theorem 2.2

[6] For the case $b_1 \geq \eta L \approx \Delta L$

Practice

AdaGrad-Norm

We show that AdaGrad-Norm converges [5]

$$\min_{\ell=0,1,\ldots,T-1} \|\nabla F(x_\ell)\|^2 \leq \mathcal{O}\left(\frac{C_1}{T} + \frac{\sigma C_2}{\sqrt{T}}\right)$$

where the constants C_1 and C_2 are explicit and **robust to hyper-parameters** b_0 **and** η.

Recall: $\mathbb{E}_{\xi_j}[\|G(x_j,\xi_j) - \nabla F(x_j)\|^2] \leq \sigma^2$

- For $\sigma \approx 0$
  Suppose we know F^* and set $\eta = F(x_0) - F^*$; the constant C_1 almost matches GD with best stepsize. [6]
- For $\sigma > 0$
  Set $\eta = 1$, the constant C_2 almost matches SGD with well-tuned stepsize up to a factor of $L \log(T/b_0 + 1)$

[5] Note we combine Theorem 2.1 and Theorem 2.2

[6] For the case $b_1 \geq \eta L \approx \Delta L$

Practice: Synthetic Data with Linear Regression

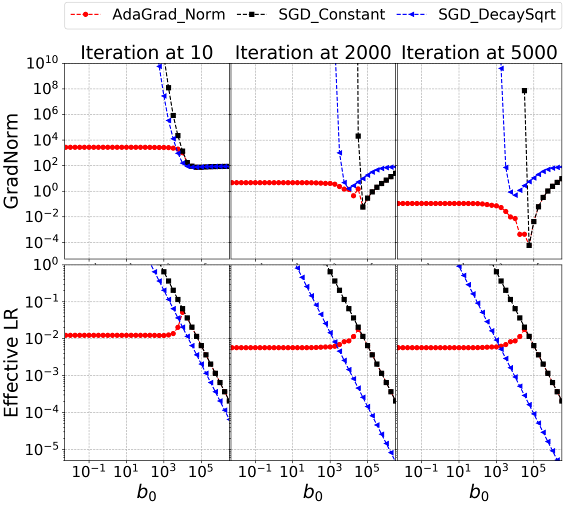

Figure 1: Random initialized x_0 with $\eta = F(x_0) - F^* = 650 - 0$.
(AdaGrad-Norm) $\frac{650}{b_j}$; (SGD-Constant) $\frac{650}{b_0}$; (SGD-DecaySqrt) $\frac{650}{b_0\sqrt{j}}$

Practice: ResNet-18 on CIFAR10

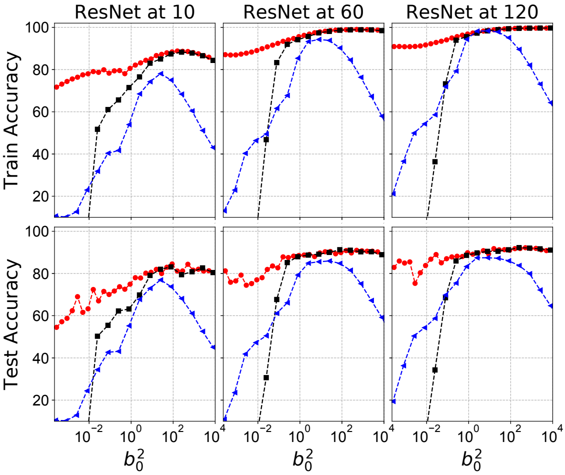

Figure 2: Random initialized x_0 with $\eta = 1$. (AdaGrad-Norm) $\frac{1}{b_j}$; (SGD-Constant) $\frac{1}{b_0}$; (SGD-DecaySqrt) $\frac{1}{b_0\sqrt{j}}$

AdaGrad-Norm code: https://github.com/xwuShirley/pytorch/blob/master/torch/optim/adagradnorm.py

Practice: ResNet-50 on ImageNet

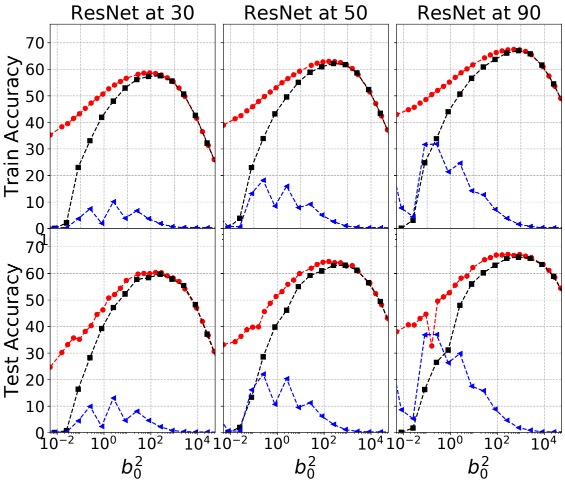

Figure 3: Random initialized x_0 with $\eta = 1$. (AdaGrad-Norm) $\frac{1}{b_j}$; (SGD-Constant) $\frac{1}{b_0}$; (SGD-DecaySqrt) $\frac{1}{b_0\sqrt{j}}$

Conclusion

- We provide a novel convergence result for AdaGrad-Norm in non-convex optimization.The analysis is useful to adaptive-type methods.

Conclusion

- We provide a novel convergence result for AdaGrad-Norm in non-convex optimization.The analysis is useful to adaptive-type methods.

- The convergence bound for AdaGrad-Norm is explicit and comparable with well-tuned stepsize choice in SGD, but without careful tuning of the AdaGrad-Norm's hyper-parameters

Conclusion

- We provide a novel convergence result for AdaGrad-Norm in non-convex optimization.The analysis is useful to adaptive-type methods.

- The convergence bound for AdaGrad-Norm is explicit and comparable with well-tuned stepsize choice in SGD, but without careful tuning of the AdaGrad-Norm's hyper-parameters

- Numerical experiments suggest that the robustness of AdaGrad-Norm extends to state-of-the-art models in deep learning, without sacrificing generalization

See you

at poster section: Pacific Ballroom #56 (Today 6:30-9:00PM).

Practice: ResNet-50 on ImageNet

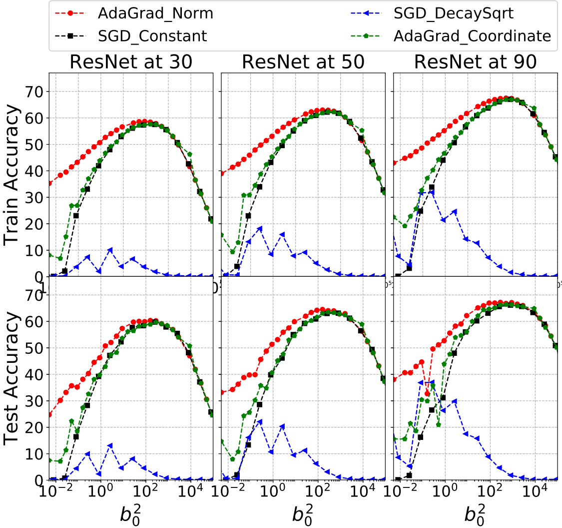

Figure 4: Random initialized x_0 with $\eta = 1$. (AdaGrad-Norm) $\frac{1}{b_j}$; (SGD-Constant) $\frac{1}{b_0}$; (SGD-DecaySqrt) $\frac{1}{b_0\sqrt{j}}$

Theory

Difficulty Proofs of SGD do not straightforwardly extend because b_{k+1} is a random variable correlated with $\nabla F(x_k)$, i.e.,

$$\mathbb{E}_{\xi_j}\left[\frac{\langle \nabla F_j, \nabla F_j - G_j\rangle}{b_{j+1}}\right] \neq \frac{\mathbb{E}_{\xi_j}\left[\langle \nabla F_j, \nabla F_j - G_j\rangle\right]}{b_{j+1}} = \frac{1}{b_{j+1}} \cdot 0;$$

(Cauchy-Schwartz)

$$\mathbb{E}_{\xi_j}\left[\left(\frac{1}{\sqrt{b_j^2 + C^2}} - \frac{1}{b_{j+1}}\right)\langle \nabla F_j, G_j\rangle\right] \leq \mathbb{E}_{\xi_j}\left[\left|\frac{1}{\sqrt{b_j^2 + C^2}} - \frac{1}{b_{j+1}}\right| \|\nabla F_j\| \|G_j\|\right]$$

(Hölder's Inequality)

$$\mathbb{E}\left[\frac{\|\nabla F_k\|^2}{\sqrt{b_{k+1}^2}}\right] \geq \frac{\left(\mathbb{E}\|\nabla F_k\|^{\frac{4}{3}}\right)^{\frac{3}{2}}}{2\sqrt{\mathbb{E}\left[b_{k+1}^2\right]}}$$