Supplementary Material for "Efficient Full-Matrix Adaptive Regularization"

Naman Agarwal [1] **Brian Bullins** [1 2] **Xinyi Chen** [1] **Elad Hazan** [1 2] **Karan Singh** [1 2] **Cyril Zhang** [1 2] **Yi Zhang** [1 2]

A. Full adaptive convergence analysis

In this section, we give the details on the theoretical treatment of GGT outlined in Section 4. The overall goal is to develop a theory for adaptive regularization in non-convex stochastic optimization. After formalizing the setting, we will define a version of GGT that uses a hard gradient memory window. This will allow us to transfer any insight on the advantage of adaptivity in the convex case to the non-convex case, giving rise to the main theorem. We will conclude this section by with an example illustrating the advantage of adaptive optimizers in the presence of sparse gradients.

A.1. Setting: stochastic non-convex optimization

Theorem A.2 will provide a bound on the number of stochastic gradient calls required by GGT to achieve a first-order critical point. In particular, the theorem shows that GGT can converge to an approximate first-order critical point faster than SGD, with convergence rate controlled by the *adaptive ratio* μ, defined in (1).

We consider the standard setting of stochastic optimization of a differentiable non-convex function $f(\cdot)$, equipped with a bounded-variance stochastic gradient oracle defined as follows.

Definition A.1 (stochastic gradient oracle)**.** *Given a function $f : D \to \mathbb{R}$ we call an oracle O_f, a σ-bounded stochastic gradient oracle if for any x, O_f returns a a random vector $\widetilde{\nabla} f(x)$ such that*

$$\mathbb{E}\left[\widetilde{\nabla} f(x)\right] = \nabla f(x) \quad \textit{and} \quad \mathbb{E}\left[\|\widetilde{\nabla} f(x) - \nabla f(x)\|^2\right] \leq \sigma^2.$$

The objective, as is standard in non-convex optimization, is to find a first-order critical point, i.e. a point x for which $\|\nabla f(x)\| \leq \varepsilon$. We will also assume that f has a Lipschitz gradient; i.e. $\|\nabla^2 f(x)\|_2 \leq L$.

Our algorithm makes a reduction to the case of stochastic convex optimization. The setting formally is that, given a smooth convex function and a σ-bounded stochastic gradient oracle, the algorithm's aim is to minimize the convex function f. Given any algorithm $\mathcal{A}$ we can now define the *adaptive ratio* of the algorithm, referred to as μ, as

$$\mu \stackrel{\text{def}}{=} \frac{f(x_{\mathcal{A}}) - f(x^*)}{\|x_1 - x^*\|_2 \cdot \frac{\sigma}{\sqrt{T}}} \tag{1}$$

where $x_{\mathcal{A}}$ is the output of the algorithm $\mathcal{A}$ and $x^* \in \operatorname{argmin}_x f(x)$, with a total of at most T calls to the stochastic gradient oracle. μ captures the advantage in convergence rate obtained by the algorithm as compared to the error obtained by vanilla SGD, noting that the denominator is a bound on the error obtained by SGD in the same setting.

A popular algorithm for stochastic (and in general online) convex optimization is AdaGrad (Duchi et al., 2011). Due to adaptive regularization, AdaGrad can often be advantageous over SGD. We quantify this advantage by the notion of μ defined above. The bounds of (Duchi et al., 2011) imply that μ can be as small as $\frac{1}{\sqrt{d}}$, depending on the geometry of the optimization problem. An example of this was provided by (Duchi et al., 2011) for both the diagonal and the full version of Adagrad. At the end of this section, we provide a different example which shows the same phenomenon even in the case of strongly convex functions.

[1]Google AI Princeton [2]Department of Computer Science, Princeton University. Correspondence to: Cyril Zhang <cyril.zhang@cs.princeton.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

In the rest of this section we describe Algorithm 2, which uses AdaGrad (Algorithm 1) as a subroutine during each window. In this regard, while stating the bounds for our algorithms, we use μ as an upper bound on the advantage of AdaGrad in each iteration.

A.2. A suitable abstraction for GGT

As mentioned in Section 4, our analysis uses a slightly idealized version of GGT, which replaces the gradient memory mechanism (governed by w and β_2) with a *hard* window; i.e., the gradient buffer is *reset* every w steps. This simple modification enables us to develop a more informative theory, in which we benefit directly from the familiar theory of AdaGrad for convex optimization, while capturing the necessity of forgetting past gradient information in adaptive non-convex optimization.

First, for clarity, we restate the definition of the full-matrix AdaGrad algorithm, introduced by (Duchi et al., 2011), which accumulates the second-moment matrix of all past gradients:

Algorithm 1 AdaGrad for convex optimization (Duchi et al., 2011)

Input: initializer x_1, window length w, stochastic gradient oracle $\widetilde{\nabla} f(\cdot)$, $\varepsilon, \eta > 0$.
for $t = 1, \dots, w$ **do**
  Receive stochastic gradient $\widetilde{\nabla} f(x_t)$.
  Let $\mathbf{G}_t = [g_t \; g_{t-1} \dots g_1]$, where $g_t := \widetilde{\nabla} f(x_t)$.
  Update $x_{t+1} \leftarrow x_t - \eta \cdot \left[\varepsilon \mathbf{I} + (\mathbf{G}_t \mathbf{G}_t^\top)^{1/2}\right]^{-1} g_t$.
end for
Output: Average iterate $\frac{1}{w}\left(\sum_{t=1}^{w} x_t\right)$.

The final algorithm we analyze simply runs AdaGrad between restarts.

Algorithm 2 GGT with a hard gradient window

Input: initializer x_1, time horizon T, window length w, $\lambda > 0$.
for $t = 1$ to T: **do**
  Let $f_t(x) = f(x) + \lambda \|x - x_t\|^2$.
  Update x_{t+1} to be the output of Algorithm 1 on $f_t(x)$, starting at x_t, for w steps.
end for
Output: Best iterate x_{t^*}, where $t^* := \operatorname{argmin}_{t \le T+1} \|\nabla f(x_t)\|$.

The remaining discrepancies between Algorithm 2 and Algorithm 1 from the main paper are standard. We provide some references below.

- **Absence of first-moment estimation.** Although it is customary to use nonzero β_1 (otherwise known as momentum) when applying Adam in practice, it is orthogonal to the effect of adaptive regularization in all established theory. In fact, the convergence rates given by (Kingma & Ba, 2014) (and fixed by (Reddi et al., 2018)) contain only factors of $1/(1-\beta_1)$, and are thus strongest when $\beta_1 = 0$.

- **Model averaging.** Theoretical guarantees in online and stochastic convex optimization are most naturally stated on the average iterate; see (Polyak & Juditsky, 1992; Duchi et al., 2011). Thus, we adopt the convention that Algorithm 1 returns the average iterate. We note that model averaging is a common regularization technique in practical non-convex settings, though not the default choice for adaptive optimizers in practice.

- **ℓ_2 regularization.** The addition of the $\lambda\|x - x_t\|^2$ term in Algorithm 2 is an artifact we introduce to obtain a tight analysis for hard-window GGT. It ensures that iterates in each window do not move too far, and allows us to analyze each window as a fixed convex program, so that we can use the convex theory of AdaGrad directly. The soft-window analogue would simply to be decrease the learning rate. Interestingly, a similar technique directly appears in the algorithm proposed by (Allen-Zhu, 2017). Finally, we note that from a σ-bounded stochastic gradient oracle for f, it is trivial to construct one for f_t, by adding $-2\lambda x_t$ (deterministically).

A.3. Main theorem and proof

Theorem A.2. *Consider a non-convex function f, such that for all x, $\|\nabla^2 f(x)\|_2 \leq L$ and a point x_1 such that $f(x_1) - \min_{x^* \in \mathcal{K}} f(x^*) \leq M$. Further, suppose we have access to a σ-bounded stochastic gradient oracle O_f. Suppose for any $\lambda \geq \frac{L}{2}$, Algorithm 2 is run with $T = \frac{4M(L+2\lambda)}{\varepsilon^2}$ and $w = \frac{16\mu^2\sigma^2(L+2\lambda)}{\varepsilon^2(2\lambda - L)}$. Then the point x' returned by Algorithm 2 is such that*

$$\mathbb{E}\|\nabla f(x')\| \leq \varepsilon,$$

where $\mu = \max_{t\in[T]} \mu_t$ and μ_t is the adaptive ratio when run on f_t (as defined in (1)*). Further, note that choosing $\lambda = 3L/2$, the total number of stochastic gradient calls to the oracle O_f, made by the algorithm is bounded by $T \cdot w = \frac{512LM\mu^2\sigma^2}{\varepsilon^4}$.*

For the setting of Theorem A.2, the best known bound on the number of oracle calls to the stochastic gradient oracle in the case of the vanilla SGD algorithm is $O(\frac{LM\sigma^2}{\varepsilon^4})$. Note that due to the presence of μ^2 in the bound provided in Theorem A.2 reflects the advantage of Algorithm 2 over SGD. This advantage as we argue in the following section can be as large as up to a factor of $1/d$, a significant improvement over SGD.

Before proving Theorem A.2, we state an oracle complexity bound for AdaGrad (Algorithm 1) for strongly convex functions.

Lemma A.3. *Suppose f is a λ-strongly convex function equipped with a σ-bounded stochastic gradient oracle. Given an initial point x_1, Algorithm 1 when run for w steps is guaranteed to output a point x' such that*

$$\mathbb{E}[f(x')] - \min_x f(x) \leq \frac{\mu^2\sigma^2\sqrt{2(f(x_1) - \min_x f(x))}}{\sqrt{\lambda w}},$$

where μ is the adaptive ratio of AdaGrad on f as defined in (1)*.*

Using this lemma we first prove Theorem A.2 and then finish the section by providing a proof of Lemma A.3.

Proof of Theorem A.2. We begin by proving the following useful property regarding the function f_t for any t and any η:

$$\begin{aligned} f_t(x_t) - \min_x f_t(x) &\geq f(x_t) - f_t(x_t - \eta\nabla f(x_t)) \\ &= f(x_t) - f(x_t - \eta\nabla f(x_t)) - \lambda\eta^2\|\nabla f(x_t)\|^2 \\ &\geq \eta\|\nabla f(x_t)\|^2 - \frac{L\eta^2}{2}\|\nabla f(x_t)\|^2 - \lambda\eta^2\|\nabla f(x_t)\|^2. \end{aligned}$$

Setting $\eta = \frac{1}{L+2\lambda}$, we get that

$$f_t(x_t) - \min_x f_t(x) \geq \frac{\|\nabla f(x_t)\|^2}{2(L+2\lambda)}. \tag{2}$$

We will now prove the theorem by contradiction. Suppose for all the t, $\|\nabla f(x_t)\|^2 > \varepsilon^2$. We now have that

$$\begin{aligned} f(x_t) - f(x_{t+1}) &\geq f_t(x_t) - f_t(x_{t+1}) \\ &= f_t(x_t) - \min_x f_t(x) - (f_t(x_{t+1}) - \min_x f_t(x)) \\ &\geq f_t(x_t) - \min_x f_t(x) - \frac{\sqrt{f_t(x_t) - \min_x f_t(x)}\sqrt{\varepsilon^2}}{2\sqrt{2(L+2\lambda)}} \\ &\geq f_t(x_t) - \min_x f_t(x) - \frac{\sqrt{f_t(x_t) - \min_x f_t(x)}\sqrt{\|\nabla f(x_t)\|^2}}{2\sqrt{2(L+2\lambda)}} \\ &\geq \frac{f_t(x_t) - \min_x f_t(x)}{2} \geq \frac{\|\nabla f_t(x_t)\|^2}{4(L+2\lambda)} > \frac{\varepsilon^2}{4(L+2\lambda)}. \end{aligned} \tag{3}$$

where the first inequality follows from noting that $f(x) \leq f_t(x)$ for all $x \in \mathbb{R}^d$, and that $f_t(x_t) = f(x_t)$. The second inequality follows from Lemma A.3, by noting that f_t is $2\lambda - L$ strongly convex and the choice of w. The third inequality follows from the counterfactual assumption $\|\nabla f(x_t)\|^2 > \varepsilon^2$, and the last set of inequalities follow from (2).

Summing (3) over all $t \in [T]$ gives us that

$$f(x_1) - f(x_{T+1}) > \frac{T\varepsilon^2}{4(L+2\lambda)} = M,$$

which is a contradiction and hence proves the theorem. The number of stochastic gradient oracle calls when $\lambda = 3L/2$ is bounded by

$$T \cdot w \leq \frac{4M(L+2\lambda)}{\varepsilon^2} \cdot \frac{16\mu^2\sigma^2(L+2\lambda)}{\varepsilon^2(2\lambda - L)} \leq \frac{512M\mu^2\sigma^2 L}{\varepsilon^4}.$$

□

Proof of Lemma A.3. We have

$$\mathbb{E}\left[f(x')\right] - f(x^*) = \mathbb{E}\left[\frac{\mu\sigma}{\sqrt{w}}\|x_1 - x^*\|\right] \leq \mathbb{E}\left[\frac{\mu\sigma\sqrt{2(f(x_1) - f(x^*))}}{\sqrt{w\lambda}}\right],$$

where the equality follows from the definition of μ and the inequality follows from strong convexity. □

A.4. Example: the advantage of adaptivity

Here we provide a strongly convex function (in fact a simple quadratic) and a sketch of the proof of the fact that depending on the starting point adaptive advantage i.e. μ of AdaGrad can be up to a factor of $\sqrt{d}$.

Consider the function $\|x\|^2$ in $\mathbb{R}^d$ and consider the starting point x_0. Let the stochastic gradient oracle O_f be such that before the experiment the oracle samples a random orthonormal basis $V = \{v_1 \dots v_d\}$ and when queried at a point x returns the vector

$$\widetilde{\nabla} f(x) = \nabla f(x) + a_t z_t$$

where $a_t = \pm 1$ with probability $1/2$ and z_t is a vector picked from the set V uniformly randomly. It is easy to verify that O_f is a σ-bounded stochastic gradient oracle. We now provide an analysis of AdaGrad with the above oracle for f.

Firstly note that we can without loss of generality, assume that the basis chosen is the canonical basis $\{e_i\}$. This can be seen by performing a simple rotation which does not affect the function $\|x\|^2$. Further under this setting note that AdaGrad is equivalent to running a one dimensional SGD algorithm in each coordinate independently. The following bound now follows directly from the well known analysis of SGD on smooth functions (see Theorem 6.3 in (Bubeck et al., 2015) for a concrete reference).

$$\forall\ i \in [d] \quad (x'[i])^2 \lesssim \frac{|x_1[i]|\sqrt{\sum_{t=1}^T (\sigma_t[i])^2}}{T} = |x_1[i]| \cdot \sqrt{\frac{1}{Td}},$$

where $\sigma_t[i] = 1/d$ is the variance of the noise in the stochastic gradient seen at time t along coordinate i and x' is the output of AdaGrad. Note that in the above we have ignored the *bias* term which scales as $1/T$ (refer to Theorem 6.3 in (Bubeck et al., 2015)). This implies that the overall error for AdaGrad scales as

$$\|x'\|^2 \lesssim \|x_1\|_1 \cdot \sqrt{\frac{1}{Td}}.$$

Therefore the advantage of adaptivity μ is bounded by

$$\mu \leq \frac{\|x_1\|_1\sqrt{\frac{1}{Td}}}{\|x_1\|_2\sqrt{\frac{1}{T}}} = \frac{\|x_1\|_1}{\|x_1\|_2\sqrt{d}}.$$

This follows by noting that the variance of the noise in the stochastic gradient measured in the ℓ_2 is 1. The above expression implies that μ can be as small as $O(\frac{1}{\sqrt{d}})$ in particular if the starting point x_1 is sparse or nearly sparse and therefore $\|x_1\|_1 \sim \|x_1\|_2$.

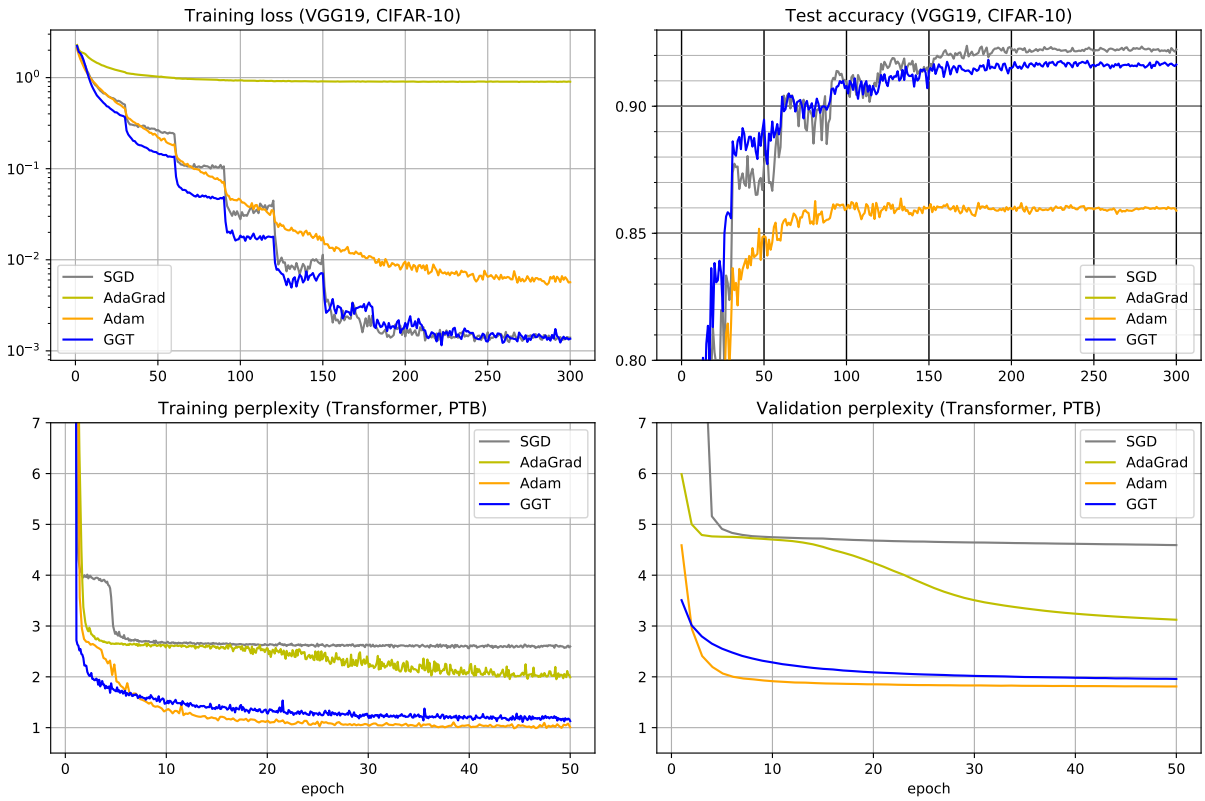

Figure 1. Plot of experiments from Sections 3.2 and 3.3, with wall clock time on horizontal axis instead of epoch count (as in Figure 3). *Top:* CIFAR-10 classification with a 19-layer vanilla CNN. *Bottom:* PTB character-level language modeling a Transformer network.

B. Additional experiments

B.1. Experiments on additional architectures

We present some additional large-scale empirical studies in Figure 1.

To demonstrate a vision task with a harder optimization landscape, we use GGT to train a 19-layer "vanilla" convolutional network (VGGNet, (Simonyan & Zisserman, 2014)), without residual connections or batch normalization, on the same CIFAR-10 classification task. Here, we recover the same insights as found by (Wilson et al., 2017), in which diagonal-matrix adaptive methods can fail to train a network dramatically. Here, unlike diagonal-matrix adaptive optimizers, GGT stays on par with SGD throughout training, with a $\sim 1\%$ gap remaining in generalization at the end. We use a standard fixed halving learning rate schedule; it is clear here that in the initial epochs after decaying the learning rate, GGT trains the most rapidly. We leave a careful investigation of leveraging this phenomenon, and tuning GGT's learning rate schedule, to future work.

A recent significant advancement on many NLP tasks, including language modeling, is the introduction of attention-based models. We investigate the behavior of GGT on a Transformer network (Vaswani et al., 2017), on the same Penn Treebank character-level language modeling task. Here, after an initial lead, GGT is outperformed by Adam in training and validation loss. The value of using gradient correlations to assist in the training of attention models seems to be limited.

References

Allen-Zhu, Z. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1200–1205. ACM, 2017.

Bubeck, S. et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8

(3-4):231–357, 2015.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Polyak, B. T. and Juditsky, A. B. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of Adam and beyond. In *International Conference on Learning Representations*, 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pp. 4151–4161, 2017.