A Scale-Free MADGRAD Regret Bound

Shashank Manjunath
Boston University
manjuns@bu.edu

December 8, 2021

Introduction

- This project is concerned with two dual averaging algorithms applied to deep learning - Modernized Dual Averaging (MDA)[1] and Momentumized, Adaptive, Dual averaged GRADient (MADGRAD)[2]
- These algorithms use Follow-the-Regularized-Leader (FTRL) style algorithms aimed to optimize deep learning techniques
- We will first discuss the algorithms in detail and their implementations and performance on the CIFAR10 dataset[3]
- We will then prove an alternate, scale-free regret bound for the MADGRAD algorithm.

Algorithm 1: Modernized Dual Averaging

Input: $x_0 \in \mathbb{R}^n$ initial point, $\gamma_k \geq 0$ stepsize sequence, c_k momentum parameter sequence. Initialize $s_{-1} = 0$

for $k = 0, \cdots, T-1$ **do**

Set the scaling coefficient $\beta_k = \sqrt{k+1}$ and stepsize $\lambda_k = \gamma\sqrt{k+1}$

Sample ξ_k and compute stochastic gradient $g_k = \nabla f(x_k, \xi_k)$.

$s_k = s_{k-1} + \lambda_k g_k$

$z_{k+1} = x_0 - \frac{s_k}{\beta_k}$

$x_{k+1} = (1 - c_{k+1})x_k + c_{k+1} z_{k+1}$

end

return x_T

Modernized Dual Averaging

- Note that MDA implements FTRL on the z_{k+1} iterates with the following update:

$$z_{k+1} = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \langle \sum_{i=1}^{k} \lambda_i g_i, x \rangle + \frac{1}{2\sqrt{k+1}} \|x - x_0\|_2 \right\}$$

- Algorithm uses an L_2 based regularizer
- Averaging technique $x_{k+1} = (1 - c_{k+1})x_k + c_{k+1} z_{k+1}$ allows use of the final iterate
- Disabling this (setting $c_{k+1} = 1$) implements a pure FTRL update, but requires that averaged iterates are used in the final model

MADGRAD

- MADGRAD implements a similar algorithm with a slightly different regularizer. We denote element-wise multiplication of vectors (Hadamard product) by $\circ$.

Algorithm 2: MADGRAD

Input: $x_0 \in \mathbb{R}^n$ initial point, $\gamma_k \geq 0$ stepsize sequence, c_k momentum parameter sequence, epsilon ϵ.

Initialize $s_0 = 0$ and $\nu_0 = 0$

for $k = 0, \cdots, T-1$ **do**

Sample ξ_k and compute stochastic gradient $g_k = \nabla f(x_k, \xi_k)$.

Set $\lambda_k = \gamma_k \sqrt{k+1}$

$s_{k+1} = s_k + \lambda_k g_k$

$\nu_{k+1} = \nu_k + \lambda_k (g_k \circ g_k)$

$z_{k+1} = x_0 - \frac{1}{\sqrt[3]{\nu_{k+1}} + \epsilon} \circ s_{k+1}$

$x_{k+1} = (1 - c_{k+1}) x_k + c_{k+1} z_{k+1}$

end

return x_T

MADGRAD

- This algorithm implements FTRL on the z_{k+1} iterates with the following update:

$$z_{k+1} = \operatorname*{argmin}_{x \in \mathbb{R}^n} \left\{ \langle \sum_{i=1}^{k} \lambda_i g_i, x \rangle + \frac{1}{2} \|x - x_0\|_{A_k}^2 \right\}$$

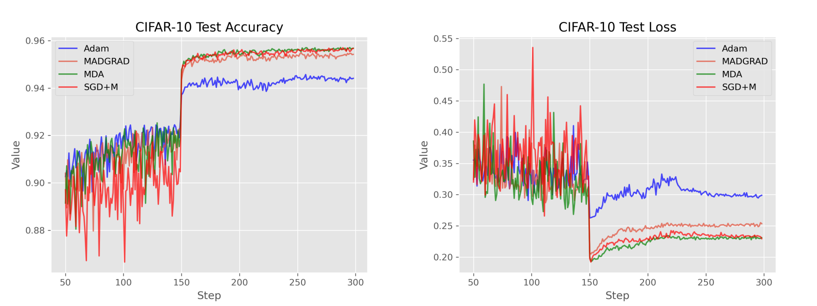
- $A_t = \operatorname{diag}\left(\sqrt[3]{\sum_{i=1}^{t} \lambda_i (g_i \circ g_i)} \right)$
- Unlike Adagrad[4], MADGRAD uses a cube root in the denominator.
- In Adagrad, the s_{k+1} iterate sequence is motivated by the following minimization problem over a D-dimensional vector:

$$\min_s \sum_{i=1}^{k} \sum_{d=0}^{D} \frac{g_{i,d}^2}{s_d}, \|s\|_1 \leq c, \forall d : s_d > 0$$

- Constraining with the L_2 norm and calculating the solution to this problem yields the cube-root denominator

Algorithm Implementation

- Results on the CIFAR10 dataset for the MDA, MADGRAD, Adam, and Stochastic Gradient Descent with Momentum (SGD+M) algorithms shown below



(a) Test Accuracy of Optimizers on CIFAR-10

(b) Test Loss of Optimizers on CIFAR-10

Figure 1: Comparison of optimizer performance on CIFAR-10 dataset

MADGRAD Theory

- In the original MADGRAD algorithm presented in the paper, the z_{k+1} is given by:

$$z_{k+1} = x_0 - \frac{1}{\sqrt[3]{v_{k+1}} + \epsilon} \circ s_{k+1}$$

- In the convergence proof, the z_{k+1} parameter is given by:

$$z_{k+1} = x_0 - \frac{1}{\sqrt[3]{\lambda_{k+1} G^2 + v_{k+1}}} \circ s_{k+1}$$

- This extra $\lambda_{k+1} G^2$ prevents the algorithm from being *scale-free*, or an algorithm that is invariant to the scaling of losses by a constant factor
- Therefore, we aim to construct a convergence proof which maintains the scale-free nature of the algorithm, and does not require assumptions about the boundedness of the subgradients
- In particular, we avoid the asumption that $\|g_i\|_\infty \leq G$
- We make two reductions in order to prove our bound: we assume that $c_k = 1$ to only use FTRL updates, and we assume a constant learning rate

Original MADGRAD Per-Coordinate Convergence Bound

$$\mathbb{E}[f(x_k) - f(x_*)] \leq \frac{3}{\gamma}\frac{1}{(k+1)^{3/2}}\sum_{d=0}^{D}\left(\mathbb{E}\left[\lambda_k\left(\sum_{i=0}^{k}\lambda_i g_{i,d}^2\right)^{2/3}\right]\right)$$
$$+\frac{3}{\gamma}\frac{1}{(k+1)^{3/2}}\sum_{d=0}^{D}(x_{0,d} - x_{*,d})^2\,\mathbb{E}\left(\lambda_{k+1}G^2 + \sum_{i=1}^{k}\lambda_i g_{i,d}^2\right)^{1/3}$$

Lemma (Lemma 1 in (Orabona and Pàl, 2015) [5])

Let $\{\psi_t\}_{t=1}^{\infty}$ *be a sequence of lower semi-continuous functions defined on a common convex domain* $S \subseteq \mathbb{R}^n$ *and such that each* ψ_t *is* μ_t*-strongly convex with respect to the norm* $\|\cdot\|_t$*. Let* $\|\cdot\|_{t,*}$ *be the dual norm of* $\|\cdot\|_t$*, for* $t = 1, 2, \cdots, T$*. Then, for any* $\mathrm{u} \in S$*, the FTRL algorithm yields:*

$$Regret_T(\mathrm{u}) \leq \sum_{t=1}^{T} \langle g_t, \mathrm{u} - \mathrm{x}_t \rangle \leq \psi_T(\mathrm{u}) + \psi_1^*(0)$$

$$+ \sum_{t=1}^{T} B_{\psi_t^*}(-\theta_t, -\theta_{t-1}) - \psi_t^*(-\theta_t) + \psi_{t+1}^*(-\theta_t)$$

Lemma

Let $a_1, a_2, \cdots, a_t$ be non-negative real numbers. If $a_1 > 0$, then

$$\sum_{t=1}^{T} \frac{a_t}{\sqrt[3]{\sum_{s=1}^{t} a_s}} \leq \frac{3}{2} \left(\sum_{t=1}^{T} a_t \right)^{\frac{2}{3}}$$

Lemma 3

Lemma

Let $C, a_1, a_2, \cdots, a_T \geq 0$, *and* $\alpha \geq 1, \alpha \neq \min\limits_{t=1,2,\cdots,T} a_t^{\frac{4}{3}}$. *Then,*

$$\sum_{t=1}^{T} \min\left\{\frac{a_t^2}{\sqrt[3]{\sum\limits_{s=1}^{t-1} a_s^2}}, Ca_t\right\} \leq \frac{C\alpha}{\alpha - \min\limits_{t=1,2,\cdots,T} a_t^{\frac{4}{3}}} \max_{t=1,2,\cdots,T} a_t + 2\sqrt[3]{1+\alpha^2}\sqrt{\sum_{s=1}^{T} a_s^3}$$

Alternative MADGRAD Bound

Theorem

Suppose $K \subseteq \mathbb{R}^D$ is a non-empty closed convex subset. Suppose that a regularizer $\psi_t : K \to \mathbb{R}$ is a non-negative lower semi-continuous function that is strongly convex with respect to a norm $\|\cdot\|_{A_t}$. The regret of non-momentumized MADGRAD satisfies:

$$Regret_T(\mathrm{u}) \leq \sum_{d=1}^{D} \frac{(\mathrm{u}_d - x_{0,d})^2}{2\sqrt[3]{\sum_{i=1}^{T} \lambda_i g_{i,d}^2}} + \frac{3}{2}\left(\sum_{i=1}^{T} \lambda_i g_{i,d}^2\right)^{\frac{2}{3}}$$

$$+2\sqrt{T-1}\left(\sum_{i=1}^{T-1} \lambda_i g_{i,d}^2\right)^{\frac{2}{3}} \left(1 + \min_{t\leq T}(\sqrt{\lambda_t}|g_{t,d}|)^{\frac{4}{3}}\right) \max_{t\leq T} \sqrt{\lambda_t}|g_{td}|$$

$$+ 2\sqrt[3]{1 + (1 + \min_{t\leq T}(\sqrt{\lambda_t}|g_{t,d}|)^{\frac{4}{3}})^2}\sqrt{\sum_{t-1}^{T} \lambda_t^{\frac{3}{2}} |g_{t,d}|^3}$$

Proof Intuition

- Our proof technique follows (Orabona and Pàl, 2015) for Scale-free Online Linear Opimization FTRL (SOLO FTRL)
- Our regularizer, $\psi_t(x) = \frac{1}{2}\|\mathrm{x} - \mathrm{x}_0\|_{A_t}^2$, is defined by a diagonal matrix, $A_t = \mathrm{diag}\left(\sqrt[3]{\sum_{i=1}^{t-1} \lambda_i g_i^2}\right)$
- $\psi_t(\mathrm{x})$ is $\min_{d \leq D} \sqrt[3]{\sum_{i=1}^{t-1} \lambda_i g_{i,d}^2}$-strongly convex
- Let $\eta_{t,d} = \frac{1}{\sqrt[3]{\sum_{i=1}^{t-1} \lambda_i g_{i,d}^2}}$
- We tighten this bound by analyzing in a per-parameter fashion by defining our regularizer in a per-coordinate manner: $\psi_{t,d}(x) = \frac{1}{\eta_{t,d}}\psi_d(\mathrm{x}) = \frac{1}{2\eta_{t,d}}(\mathrm{x}_d - \mathrm{x}_{0,d})^2$
- Since $\psi_d(\mathrm{x}) : \mathbb{R} \to \mathbb{R}$, we can find the strong convexity constant by finding the lower bound of the second derivative of $\psi_d(\mathrm{x})$, which is 1.
- We use this per-coordinate regularizer in order to prove an entirely per-coordinate bound

Proof Technique

- We start with Lemma 1, and upper bound the $B_{\psi_t^*}(-\theta_t, -\theta_{t-1}) - \psi_t^*(-\theta_t) + \psi_{t+1}^*(-\theta_t) \leq B_{\psi_t^*}(-\theta_t, -\theta_{t-1})$ in two ways using the properties of Fenchel conjugates, with the upper bound being the minimum of these two terms. Note that $H = \left(\sum_{i=1}^{T-1} \lambda_i g_{i,d}^2\right)^{\frac{2}{3}} \sqrt{T-1}$

$$\begin{aligned}
\text{Regret}_T(\mathrm{u}) &\leq \sum_{d=1}^{D} \frac{1}{\eta_{T+1}} \psi_d(\mathrm{u}) \frac{1}{\eta_1} \psi_d^*(0) \\
&\quad + \sum_{t=1}^{T} \min\left\{ \frac{\eta_t \lambda_t g_{t,d}^2}{2}, \frac{\eta_{t+1,d} \lambda_t g_{t,d}^2}{2} + H\sqrt{\lambda_t} |g_{t,d}| \right\} \\
\therefore \text{Regret}_T(\mathrm{u}) &\leq \frac{1}{\eta_{T+1}} \psi_d(\mathrm{u}) + \frac{1}{\eta_1} \psi_d^*(0) \\
&\quad + \frac{1}{2} \sum_{t=1}^{T} \eta_{t+1} \lambda_t g_{t,d}^2 + \frac{1}{2} \sum_{t=1}^{T} \min\left\{ \frac{\eta_t \lambda_t g_{t,d}^2}{2}, 2H\sqrt{\lambda_t} |g_{t,d}| \right\}
\end{aligned}$$

- We then use Lemma 3 to upper bound the minimum
- Lastly, we use Lemma 2 to upper bound the $\frac{1}{2} \sum_{t=1}^{T} \eta_{t+1} \lambda_t g_{t,d}^2$

Conclusion

- In this work, we implement and analyze the performance of MDA and MADGRAD algorithms on the CIFAR10 dataset
- We then prove an alternate, scale-free regret bound for the MADGRAD algorithm

References

- S. Jelassi and A. Defazio, "Dual Averaging is Surprisingly Effective for Deep Learning Optimization," *arXiv:2010.10502 [cs, math, stat]*, Oct. 2020. **arXiv: 2010.10502.**
- A. Defazio and S. Jelassi, "Adaptivity without Compromise: A Momentumized, Adaptive, Dual Averaged Gradient Method for Stochastic Optimization," p. 33.
- A. Krizhevsky, "Learning multiple layers of features from tiny images," tech. rep., 2009.
- J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," p. 13.
- F. Orabona, K. Crammer, and N. Cesa-Bianchi, "A Generalized Online Mirror Descent with Applications to Classification and Regression," *arXiv:1304.2994 [cs]*, July 2014. **arXiv: 1304.2994.**

Code for PyTorch implementation, LaTeX for presentation and paper can be found at on GitHub at https://github.com/shashankmanjunath/ftrl_deep_learning