A Scale-Free MADGRAD Regret Bound

Shashank Manjunath
Boston University
manjuns@bu.edu

Abstract

We introduce a new convergence bound for the MADGRAD optimization algorithm which requires no assumption of the boundedness of gradients. MADGRAD, an optimization algorithm in the family of AdaGrad adaptive gradient methods, shows promising results on deep learning optimization problems, outperforming Adam and providing a provable convergence bound (Defazio and Jelassi). We additionally show emprical results of MADGRAD and Modernized Dual Averaging (MDA) (Jelassi and Defazio [2020]) in comparison to Adam and SGD on the CIFAR10 dataset (Krizhevsky [2009]).

1 Introduction

Significant literature exists applyting dual averaging algorithms to deep learning style algorithms, most notably in AdaGrad (Duchi et al.). However, analysis of these algorithms using Follow-the-Regularized Leader (FTRL) analysis techniques is more rare. Many convergence proofs require assumptions about the boundedness of the domain or the boundedness of the gradients of the algorithm. This paper focuses on implementing two dual averaging algorithms, Modernized Dual Averaging (MDA) (Jelassi and Defazio [2020]) and MADGRAD (Defazio and Jelassi), which use Follow the Regularized Leader (FTRL) style algorithms in order to optimize deep learning algorithms, but do not use FTRL style analysis when proving a convergence bound. We first present a simple motivation for each algorithm, then present our implementation results for these algorithms in PyTorch (Paszke et al. [2019]) on the CIFAR10 dataset (Krizhevsky [2009]). We then prove an alternate, scale-free regret bound for the MADGRAD algorithm in Section 4.

2 Algorithm Details

2.1 Algorithm Comparison

MDA is defined as follows:

```
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 $\gamma_k = \gamma\sqrt{k+1}$
    Sample $\xi_k$ $P$ 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 $x_T$
```

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

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

This algorithm uses an L_2 based regularizer function. The averaging technique used to create x_{k+1}, $x_{k+1} = (1 - c_{k+1})x_k + c_{k+1} z_{k+1}$, allows use of the final iterate, x_T, as the algorithm parameters. Removing this and directly using the z_{k+1} iterates requires using the averaged iterates as algorithm parameters, i.e. $x_T = \frac{1}{T-1} \sum_{i=1}^{T-1} x_i$. This can become infeasible for a large number of rounds or models which require a large number of parameters such as deep neural networks.

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 $\epsilon$.
Initialize $s_0 = 0$ and $\nu_0 = 0$
for $k = 0, \cdots, T-1$ do
    Sample $\xi_k$ $P$ and compute stochastic gradient $g_k = \nabla f(x_k, \xi_k)$.
    Set $\lambda_k = \gamma_k = \gamma\sqrt{k+1}$
    $s_{k+1} = s_k + \lambda_k g_k$
    $\nu_{k+1} = \nu_k + \lambda_k (g_k \circ g_k)$
    $z_{k+1} = x_0 - \frac{1}{\sqrt[3]{\nu_{k+1}}+\epsilon} \circ s_{k+1}$
    $x_{k+1} = (1 - c_{k+1})x_k + c_{k+1} z_{k+1}$
end
return $x_T$
```

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

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

where $A_t = \text{diag}\left(\sqrt[3]{\sum_{i=1}^{k} \lambda_k (g_k \circ g_k)} \right)$.

2.2 MADGRAD Cube-Root Denominator

Unlike Adagrad (Duchi et al.) and many other optimization algorithms, MADGRAD uses a cube root in the denominator. This is discussed in (Defazio and Jelassi) and can be motivated by a small modification to Adagrad. In Adagrad, the s_{k+1} iterate sequence is motivated by the following minimization problem over a D-dimensional vector:

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

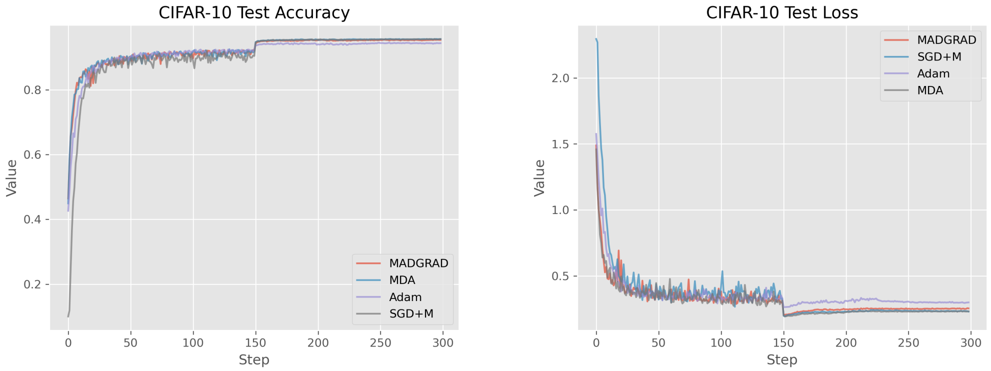
This is solved by $s_d \propto \sqrt{\sum_{i=0}^{k} g_{i,d}^2}$. However, consider minimizing the L_2 norm squared of s:

$$\min_s \sum_{i=1}^{k} \sum_{d=0}^{D} \frac{g_{i,d}^2}{s_d}, \|s\|_2^2 \leq c, \forall d : s_d > 0$$

Solving this problem yields $s_d \propto \sqrt[3]{\sum_{i=0}^{k} g_{i,d}^2}$

3 Algorithm Implementation

We have successfully replicated results on the CIFAR10 dataset for the MDA, MADGRAD, Adam, and Stochastic Gradient Descent with Momentum (SGD+M) algorithms. We show our test accuracy and test loss results in the plot 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

Our experiment setup and optimizer implementations for MDA and MADGRAD can be found at `https://github.com/shashankmanjunath/ftrl_deep_learning`. We provide algorithm hyperparameters in A.1.

Comparing algorithm performance shows that MDA and MADGRAD achieve performance on par with tuned SGD+M, and outperform tuned Adam. However, MDA and MADGRAD also require significant parameter tuning in order to perform appropriately, and therefore do not necessarily provide a significant advantage over SGD+M.

4 Theory

When proving the convergence bound for MADGRAD, the authors require an alternative definition of MADGRAD than the one presented in the paper and implemented. In the original MADGRAD algorithm presented in the paper, the z_{k+1} is given by:

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

where $\circ$ indicates the Hadamard product. ϵ is included for numerical stability in the early iterations of the algorithm, as the v_{k+1} parameter can be 0. However, in the convergence proof, the z_{k+1} parameter is given by:

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

Note the extra $\lambda_{k+1} G^2$ in the denominator, which is used to create the following upper bound leveraged in the overall convergence proof proposed by the original authors:

$$\sum_{t=0}^{k} \frac{\lambda_t^2 g_t^2}{\sqrt[3]{\lambda_t G^2 + \sum_{i=0}^{t-1} \lambda_i g_i^2}} \leq \frac{3}{2} \lambda_k \left(\sum_{i=1}^{k} \lambda_i g_i^2 \right)^{\frac{2}{3}}$$

This extra $\lambda_t G^2$ 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.

4.1 Proof of Scale-Free Regret Bound for MADGRAD

Consider the MADGRAD algorithm. This algorithm implements FTRL with the regularizer:

$$\psi_t(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|_{A_t}$$

where $A_t = \text{diag}(\alpha_t)$, and $\alpha_t = \sqrt[3]{\sum_{i=1}^{t-1} \lambda_i g_i^2}$. Note that $\psi_t(\mathbf{x})$ is strongly-convex with respect to the norm $\|\cdot\|_{A_t}$. Let us denote the Bregman divergence of a function f over two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$ by $B_f(\mathbf{x}; \mathbf{y})$ and let f^* indicate the Fenchel conjugate of f. In order to prove the scale-free bound, we make the reduction that $c_t = 1$ for all rounds. This effectively removes the momentum operation, and leaves us with only FTRL iterates. Let us first make a useful proposition. Proving a scale-free bound for the whole algorithm will require modification of Lemma 1 in order to handle the modified update. Lemmas of this form are proved in (Nesterov and Shikhman [2015]); however, we were not able to prove a specific alternate lemma which would enable a robust convergence bound. Note that we make no assumption on the boundedness of our convex set K used as the feasible set for our algorithm. Additionally, we make no assumptions about the boundedness of subgradients g_i.

Proposition 1 (Proposition 2 in Orabona et al. [2014]: Properties of Fenchel Conjugates of Strongly Convex Functions)**.** *Let $K \subseteq \mathbb{R}^D$ be a non-empty closed convex set. Let $\lambda \geq 0$, and let $f : K \to R$ be a lower semi-continuous function that is λ-strongly convex with respect to $\|\cdot\|$. The Fenchel conjugate of f satisfies:*

1. *f^* is finite everywhere and differentiable*
2. *$\nabla f^*(\ell) = argmin_{w \in K}(f(w) - \langle \ell, w\rangle)$*
3. *For any $\ell \in V^*$, $f^*(\ell) + f(\nabla f^*(\ell)) = \langle \ell, \nabla f^*(\ell)\rangle$*
4. *f^* is $\frac{1}{\lambda}$-strongly smooth, i.e. for any $x, y \in V^*$, $B_{f^*}(x, y) \leq \frac{1}{2\lambda}\|x - y\|_*$*
5. *f^* has $\frac{1}{\lambda}$-Lipschitz continuous gradients, i.e. $\|\nabla f^*(x) - \nabla f^*(y)\| \leq \frac{1}{\lambda}\|x - y\|_*$ for any $x, y \in V^*$*

Let us now define some useful lemmas.

Lemma 1 (Lemma 1 in Orabona et al. [2014])**.** *Let $\{\psi_t\}_{t=1}^{\infty}$ be a sequence of lower semi-continuos 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 $\mathbf{u} \in S$,*

$$Regret_T(\mathbf{u}) \leq \sum_{t=1}^{T}\langle g_t, \mathbf{u} - \mathbf{x}_t\rangle \leq \psi_T(\mathbf{u}) + \psi_1^*(0) + \sum_{t=1}^{T} B_{\psi_t^*}(-\theta_t, -\theta_{t-1}) - \psi_t^*(-\theta_t) + \psi_{t+1}^*(-\theta_t)$$

Proof. Given in (Orabona et al. [2014]) □

Lemma 2. *Let $a_1, a_2, \cdots, a_t$ be non-negative real numbers. If $a_1 > 0$, then*

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

Proof. Given in A.2

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

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

Proof. Given in A.3

Let us now state the overall convergence bound for our version of MADGRAD. Our proof of this technique broadly follows the technique set forth for Scale-Free Online Linear Optimization FTRL given in (Orabona and Pal [2015]) applied in a per-coordinate manner. We additionally only handle the constant learning rate case, which ensures that $\lambda_t \leq \lambda_{t+1}$

Theorem 1. *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:*

$$\textit{Regret}_T(\mathbf{u}) \leq \sum_{d=1}^{D} \frac{(\mathbf{u}_d - x_{0,d})}{2\sqrt[3]{\sum_{i=1}^{T} \lambda_i g_{i,d}^2}} + \frac{3}{2}\left(\sum_{i=1}^{T} \lambda_i g_{i,d}^2\right)^{\frac{2}{3}} + 2\sqrt{T-1}\left(\sum_{i=1}^{T-1} \lambda_i g_{i,d}^2\right)^{\frac{2}{3}} (1 + \min_{t \leq T}(\sqrt{\lambda_t}|g_{t,d}|)^{\frac{4}{3}}) \max_{t \leq T} \sqrt{\lambda_t}|g_{td}|$$

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

Proof. Note that $\psi_t(x) = \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|_{A_t}^2$, where $A_t = \text{diag}(\alpha_t)$. For this regularizer, $\alpha_t = \sqrt[3]{\sum_{i=1}^{t-1} \lambda_i g_i^2} \in \mathbb{R}^D$. Let $L_t = \sum_{i=1}^{t} \lambda_i g_i$.Let us perform this analysis per-coordinate.

$$\psi_{t,d}(\mathbf{x}) = \frac{1}{2}(\mathbf{x}_d - \mathbf{x}_{0,d})\sqrt[3]{\sum_{i=1}^{t-1} \lambda_i g_{id}^2}(\mathbf{x}_d - \mathbf{x}_{0,d}) = \frac{1}{2}\sqrt[3]{\sum_{i=1}^{t-1} \lambda_i g_{id}^2}(\mathbf{x}_d - \mathbf{x}_{0,d})^2.$$

Let $\eta_{t,d} = \frac{1}{\sqrt[3]{\sum_{i=1}^{t-1} \lambda_i g_{id}^2}}$. Therefore, we have:

$$\psi_{t,d}(\mathbf{x}) = \frac{1}{\eta_{t,d}}\psi_d(\mathbf{x})$$
$$\psi_d(\mathbf{x}) = \frac{1}{2}(\mathbf{x}_d - \mathbf{x}_{0,d})$$

Since A_t is a diagonal matrix, $\psi_t(\mathbf{x})$ is $\min_{d \leq D} \sqrt[3]{\sum_{i=1}^{t-1} \lambda_i g_{i,d}^2}$-strongly convex. We can tighten this bound by analyzing $\psi_{t,d}(\mathbf{x})$ and establishing a per-coordinate strong convexity bound. Recall that, since $\psi_d(\mathbf{x}) : \mathbb{R} \to \mathbb{R}$, we can find the strong convexity constant by finding the lower bound of the second derivative of $\psi_d(\mathbf{x})$.

$$\frac{d\psi_d(\mathbf{x})}{d\mathbf{x}_d} = \mathbf{x}_d - \mathbf{x}_{0,d}$$
$$\frac{d^2\psi_d(\mathbf{x})}{d\mathbf{x}_d^2} = 1$$

Therefore, $\psi_d(\mathbf{x})$ is 1-strongly convex with respect to $|\cdot|$, as we are dealing with real numbers in the per-coordinate case. By Lemma 1, we have:

$$\text{Regret}_T(\mathbf{u}) \le \psi_{T+1}(\mathbf{u}) + \psi_1^*(\mathbf{0}) + \sum_{t=1}^{T} B_{\psi_t^*}(-L_t, -Lt-1) - \psi_t^*(-L_t) + \psi_{t-1}^*(-L_t)$$

Writing this in a per-coordinate manner,

$$\begin{aligned}\text{Regret}_T(\mathbf{u}) &\le \sum_{d=1}^{D} \psi_{T+1,d}(\mathbf{u}) + \psi_{1,d}^*(\mathbf{0}) + \sum_{t=1}^{T} B_{\psi_{t,d}^*}(-L_t, -Lt-1) - \psi_{t,d}^*(-L_t) + \psi_{t-1,d}^*(-L_t)\\ &\le \sum_{d=1}^{D} \frac{1}{\eta_{T+1}}\psi_d(\mathbf{u}) + \frac{1}{\eta_1}\psi_d^*(\mathbf{0}) + \sum_{t=1}^{T} B_{\psi_{t,d}^*}(-L_t, -Lt-1) - \psi_{t,d}^*(-L_t) + \psi_{t-1,d}^*(-L_t)\end{aligned}$$

Let us proceed by bounding $B_{\psi_{t,d}^*}(-L_t, -Lt-1) - \psi_{t,d}^*(-L_t) + \psi_{t-1,d}^*(-L_t)$ in two ways.

1. By Proposition 1 item 4, we know that $B_{\psi_{t,d}^*}(-L_t, -L_{t-1}) \le \frac{\eta_t \lambda_t g_{t,d}^2}{2\mu_{t,d}} = \frac{\eta_t \lambda_t g_{t,d}^2}{2}$. Therefore, by Lemma 1, we know that:

$$B_{\psi_{t,d}^*}(-L_t, -L_{t-1}) - \psi_{t,d}^*(-L_t) + \psi_{t+1,d}^*(-L_t) \le B_{\psi_{t,d}^*}(-L_t, -L_{t-1}) \le \frac{\eta_t \lambda_t g_{t,d}^2}{2}$$

since $\psi_{t,d}^* \ge \psi_{t+1,d}^*$.

2. Similarly,

$$\begin{aligned}B_{\psi_{t,d}^*}(-L_t, -L_{t-1}) - \psi_{t,d}^*(-L_t) + \psi_{t+1,d}^*(-L_t) &= B_{\psi_{t+1,d}^*}(-L_t, -L_{t-1}) + \psi_{t+1,d}^*(-L_{t-1})\\ &\quad - \psi_{t,d}^*(-L_{t-1}) + \langle \nabla\psi_{t,d}^*(-L_{t-1}) - \nabla\psi_{t+1,d}^*(-L_{t-1}, g_{t,d}))\rangle\\ &\le \frac{1}{2}\eta_{t+1,d}\lambda_t g_{t,d}^2 + |\nabla\psi_{t,d}^*(-L_{t-1}) - \nabla\psi_{t+1,d}^*(-L_{t-1})||g_{t,d}|\\ &\le \frac{1}{2}\eta_{t+1,d}\lambda_t g_{t,d}^2 + |\nabla\psi_d^*(-\eta_{t,d}L_{t-1}) - \nabla\psi_d^*(-\eta_{t+1}L_{t-1})||g_{t,d}|\\ &\le \frac{1}{2}\eta_{t+1,d}\lambda_t g_{t,d}^2 + |L_{t-1}|(\eta_{t,d} - \eta_{t+1,d})|g_{t,d}|\end{aligned}$$

Recall that $\eta_{t,d} = \frac{1}{\sqrt[3]{\sum\limits_{i=1}^{t-1} \lambda_i g_{id}^2}}$. Therefore,

$$\begin{aligned}|L_{t-1}|(\eta_{t,d} - \eta_{t+1,d}) \le |L_{t-1}|\eta_{t,d} &= \frac{\sum_{i=1}^{t-1} \lambda_i g_{i,d}}{\sqrt[3]{\sum_{i=1}^{t-1} \lambda_i g_{i,d}^2}}\\ &\le \frac{\sqrt{(\sum_{i=1}^{t-1} \sqrt{\lambda_i} g_{i,d}\sqrt{\lambda_i})^2}}{\sqrt[3]{\sum_{i=1}^{t-1} \lambda_i g_{i,d}^2}}\\ &\le \frac{\sqrt{(\sum_{i=1}^{t-1} \lambda_i)(\sum_{i=1}^{t-1} \lambda_i g_{i,d}^2)}}{\sqrt[3]{\sum_{i=1}^{t-1} \lambda_i g_{i,d}^2}}\end{aligned}$$

By Callebaut's inequality. Therefore, we have:

$$|L_{t-1}|(\eta_{t,d} - \eta_{t+1,d}) \leq \left(\sqrt{\sum_{i=1}^{t-1} \lambda_i}\right) \left(\sum_{i=1}^{t-1} \lambda_i g_{i,d}^2\right)^{\frac{2}{3}}$$

Combining these two bounds, we have:

$$B_{\psi_{t,d}^*}(-L_t, -L_{t-1}) - \psi_{t,d}^*(-L_t) + \psi_{t+1,d}^*(-L_t) \leq \frac{\eta_t \lambda_t g_{t,d}^2}{2} + \left(\sqrt{\sum_{i=1}^{t-1} \lambda_i}\right) \left(\sum_{i=1}^{t-1} \lambda_i g_{i,d}^2\right)^{\frac{2}{3}}$$

Note that

$$\sqrt{\sum_{i=1}^{t-1} \lambda_i} \leq \sqrt{(t-1)\lambda_t} = \sqrt{t-1}\sqrt{\lambda_t}$$

Let $H = \left(\sum\limits_{i=1}^{t-1} \lambda_i g_{i,d}^2\right)^{\frac{2}{3}} \sqrt{t-1}$. Therefore, we have:

$$B_{\psi_{t,d}^*}(-L_t, -L_{t-1}) - \psi_{t,d}^*(-L_t) + \psi_{t+1,d}^*(-L_t) \leq \frac{\eta_{t+1,d} \lambda_t g_{t,d}^2}{2} + H\sqrt{\lambda_t}|g_{t,d}|$$

Therefore, we have an overall regret bound of

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

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

Let us bound this regret in three groups.

1. $\frac{1}{2}\sum\limits_{t=1}^{T} \min\left\{\frac{\eta_t \lambda_t g_{t,d}^2}{2}, 2H\sqrt{\lambda_t}|g_{t,d}|\right\}$

   We bound this using Lemma 3. Let $B = \min_{t \leq T} a_t^{\frac{4}{3}}$ setting $\alpha = 1 + B$.

   $$\frac{1}{2}\sum_{t=1}^{T} \min\left\{\frac{\eta_t \lambda_t g_{t,d}^2}{2}, 2H\sqrt{\lambda_t}|g_{t,d}|\right\} \leq 2H(1+B) \max_{t \leq T} \sqrt{\lambda_t}|g_{t,d}| + 2\sqrt[3]{1 + (1+B)^2}\sqrt{\sum_{t=1}^{T} \lambda_t^{\frac{3}{2}} g_{t,d}^3}$$

2. $\sum\limits_{t=1}^{T} \frac{\lambda_t g_{t,d}^2}{\sqrt[3]{\sum\limits_{i=1}^{t} \lambda_i g_{i,d}^2}}$. We bound this by Lemma 2.

   $$\sum_{t=1}^{T} \frac{\lambda_t g_{t,d}^2}{\sqrt[3]{\sum\limits_{i=1}^{t} \lambda_i g_{i,d}^2}} \leq \frac{3}{2}\left(\sum_{t=1}^{T} \lambda_t g_{t,d}^2\right)^{\frac{2}{3}}$$

3. $\frac{1}{\eta_{T+1}}\psi_d(\mathbf{u}) + \frac{1}{\eta_1}\psi_d^*(0)$.

   Note that $\psi_d(\mathbf{u}) = \frac{1}{2}(\mathbf{u}_d - x_{0,d})$. Therefore,

$$\frac{1}{\eta_{T+1,d}}\psi_d(\mathbf{u}) = \frac{(\mathbf{u}_d - x_{0,d})^2}{2\sqrt[3]{\sum\limits_{t=1}^{T} \lambda_i g_{i,d}^2}}$$

Now let us analyze $\psi_d^*(0)$. By Proposition 1 item 2,

$$\psi_d^*(0) = \sup_{x \in K}(\langle x, 0\rangle - \psi_d(x)) = \sup_{x \in K}(-\frac{1}{2}(\mathbf{x}_d - x_{0,d})) \leq 0$$

Therefore, $\frac{1}{\eta_{T+1}}\psi_d(\mathbf{u}) + \frac{1}{\eta_1}\psi_d^*(0) \leq \frac{1}{\eta_{T+1}}\psi_d(\mathbf{u})$

Substituting these three upper bounds back into (1) gives the desired bound.

□

5 Conclusion

In this work, we have tested implementations of the MADGRAD and MDA algorithms, and proved a convergence bound for non-momentumized MADGRAD. Future work includes proving a convergence bound for momentumized MADGRAD.

References

Aaron Defazio and Samy Jelassi. Adaptivity without Compromise: A Momentumized, Adaptive, Dual Averaged Gradient Method for Stochastic Optimization. page 33.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. page 13.

Samy Jelassi and Aaron Defazio. Dual Averaging is Surprisingly Effective for Deep Learning Optimization. *arXiv:2010.10502 [cs, math, stat]*, October 2020. URL `http://arxiv.org/abs/2010.10502`. arXiv: 2010.10502.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Yu. Nesterov and V. Shikhman. Quasi-monotone Subgradient Methods for Nonsmooth Convex Minimization. *Journal of Optimization Theory and Applications*, 165(3):917–940, June 2015. ISSN 0022-3239, 1573-2878. doi: 10.1007/s10957-014-0677-5. URL `http://link.springer.com/10.1007/s10957-014-0677-5`.

Francesco Orabona and David Pal. Scale-Free Algorithms for Online Linear Optimization. *arXiv:1502.05744 [cs, math]*, July 2015. URL `http://arxiv.org/abs/1502.05744`. arXiv: 1502.05744.

Francesco Orabona, Koby Crammer, and Nicolò Cesa-Bianchi. A Generalized Online Mirror Descent with Applications to Classification and Regression. *arXiv:1304.2994 [cs]*, July 2014. URL `http://arxiv.org/abs/1304.2994`. arXiv: 1304.2994.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

A Appendix

A.1 Algorithm Parameters

We set our parameters as described in Defazio and Jelassi, following standard practice. Our data augmentation pipeline includes random horizontal flipping, random cropping to 32x32, then normalization by centering around (0.5, 0.5, 0.5).

Hyperparameter	**Value**
Architecture	PreAct ResNet152
Epochs	300
GPUs	1x A100
Batch Size per GPU	128
Learning Rate Schedule	150-225 tenthing

Method	Learning Rate	Decay
MADGRAD	2.5e-4	0.0001
MDA	2.5e-4	0.0001
Adam	2.5e-4	0.0001
SGD+M	0.1	0.0001

A.2 Proof of Lemma 2

Lemma 2. *Let $a_1, a_2, \cdots, a_t$ be non-negative real numbers. If $a_1 > 0$, then*

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

Proof. Note that if $0 \leq x \leq 1$,

$$\frac{2}{3} x \leq 1 - (1 - x)^{\frac{2}{3}}$$

Let $L_t = \sum\limits_{i=1}^{t} \ell_i$, and let $x = \frac{\ell_t}{L_t}$. Let $\ell_0 = 0$.

$$\begin{aligned}
\frac{2}{3} \frac{\ell_t}{L_t} &\leq 1 - (1 - \frac{\ell_t}{L_t})^{\frac{2}{3}} = 1 - (\frac{L_{t-1}}{L_t})^{\frac{2}{3}} \\
\frac{2}{3} \frac{\ell_t}{L_t} L_t^{\frac{2}{3}} &\leq L_t^{\frac{2}{3}} - L_{t-1}^{\frac{2}{3}} \\
\frac{2}{3} \frac{\ell_t}{\sqrt[3]{L_t}} &\leq L_t^{\frac{2}{3}} - L_{t-1}^{\frac{2}{3}} \\
\therefore \frac{2}{3} \sum_{t=1}^{T} \frac{\ell_t}{\sqrt[3]{L_t}} &\leq \sum_{t=1}^{T} L_t^{\frac{2}{3}} - L_{t-1}^{\frac{2}{3}} \\
\sum_{t=1}^{T} \frac{\ell_t}{\sqrt[3]{L_t}} &\leq \frac{3}{2} L_T^{\frac{3}{2}} \\
\sum_{t=1}^{T} \frac{\ell_t}{\sqrt[3]{L_t}} &\leq \frac{3}{2} \left(\sum_{t=1}^{T} \ell_t \right)^{\frac{2}{3}}
\end{aligned}$$

Letting $\ell_i = a_i \forall i$ yields the lemma.

□

A.3 Proof of Lemma 3

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

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

Proof. We will prove this bound by proving each individual case in the minimum, then summing them.

Case 1. Consider $a_t \leq \alpha^3 \left(\sum\limits_{s=1}^{t-1} a_s^2\right)^{\frac{2}{3}}$.

$$\begin{aligned}
\min\left\{ \frac{a_t^2}{\sqrt[3]{\sum\limits_{s=1}^{t-1} a_s^2}}, C a_t \right\} &\leq \frac{\alpha_t^2}{\sqrt[3]{\sum\limits_{s=1}^{t-1} a_s^2}} = \frac{a_t^2}{\sqrt[3]{\frac{1}{1+\alpha^2}\left(\alpha^2 \sum\limits_{s=1}^{t-1} a_s^2 + \sum\limits_{s=1}^{t-1} a_s^2\right)}} \\
&\leq \frac{a_t^2 \sqrt[3]{1+\alpha^2}}{\sqrt[3]{a_t^2 + \sum\limits_{s=1}^{t-1} a_s^2}} = \frac{a_t^2 \sqrt[3]{1+\alpha^2}}{\sqrt[3]{\sum\limits_{s=1}^{t} a_s^2}}
\end{aligned}$$

Note that $\frac{x^2}{\sqrt[3]{x^2+y^2}} \leq 2(\sqrt{x^3+y^3} - \sqrt{y^3})$. Using this inequality,

$$\sqrt[3]{1+\alpha^2} \frac{a_t^2}{\sqrt[3]{\sum\limits_{s=1}^{t} a_s^2}} \leq 2\sqrt[3]{1+\alpha^2}\left(\sqrt{\sum_{s=1}^{t} a_s^3} - \sqrt{\sum_{s=1}^{t-1} a_s^3}\right)$$

Case 2 Consider $a_t^2 \geq \alpha^3 \left(\sum\limits_{s=1}^{t-1} a_s^2\right)^{\frac{2}{3}}$. Note that this implies that $a_t \geq \alpha^{\frac{3}{2}} \sqrt[3]{\sum\limits_{s=1}^{t-1} a_s^2}$. Additionally, let $A = \left(\sum\limits_{s=1}^{t-1} a_s^2\right)^{\frac{2}{3}}$.

$$\begin{aligned}
\min\left\{ \frac{a_t^2}{\sqrt[3]{\sum\limits_{s=1}^{t-1} a_s^2}}, C a_t \right\} &\leq C a_t = C a_t \left(\frac{\alpha - A}{\alpha - A}\right) \\
&\leq \frac{C}{\alpha - A}\left(\alpha a_t - A a_t\right) = \frac{C\alpha}{\alpha - A}\left(a_t - \alpha^{\frac{1}{2}} A \sqrt[3]{\sum_{s=1}^{t-1} a_s^2}\right) \\
&\leq \frac{C\alpha}{\alpha - A}\left(a_t - \left(\sum_{s=1}^{t-1} a_s^2\right)^{\frac{2}{3}} \left(\sum_{s=1}^{t-1} a_s^2\right)^{\frac{1}{3}}\right) \\
&\leq \frac{C\alpha}{\alpha - A}\left(a_t - \sqrt{\sum_{s=1}^{t-1} a_s^2}\right)
\end{aligned}$$

Let $M_t = \max\{a_t, \cdots, a_t\}$. Note that in this case, $a_t = M_t$, and $\sqrt{\sum\limits_{s=1}^{t-1} a_s^2} \geq M_{t-1}$. Therefore,

$$\min\left\{\frac{a_t^2}{\sqrt[3]{\sum\limits_{s=1}^{t-1} a_s^2}}, Ca_t\right\} \leq \frac{C\alpha}{\alpha - A}\left(M_t - M_{t-1}\right)$$

Further note that since $a_t \geq 0 \forall t$, $A = (\sum\limits_{s=1}^{T} a_s^2)^{\frac{2}{3}} \geq \min_{t=1,\cdots,T} a_t^{\frac{4}{3}}$. Let $B = \min_{t=1,\cdots,T} a_t^{\frac{4}{3}}$, Therefore,

$$\min\left\{\frac{a_t^2}{\sqrt[3]{\sum\limits_{s=1}^{t-1} a_s^2}}, Ca_t\right\} \leq \frac{C\alpha}{\alpha - B}\left(M_t - M_{t-1}\right)$$

Therefore, combining the two cases, we have:

$$\min\left\{\frac{a_t^2}{\sqrt[3]{\sum\limits_{s=1}^{t-1} a_s^2}}, Ca_t\right\} \leq \frac{C\alpha}{\alpha - B}\left(M_t - M_{t-1}\right) + 2\sqrt[3]{1+\alpha^2}\left(\sqrt{\sum_{s=1}^{t} a_s^3} - \sqrt{\sum_{s=1}^{t-1} a_s^3}\right)$$

Therefore, summing from $t = 1$ to T,

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

□