On the Proximal Functions of Adaptive Methods and Beyond

Wenjie Li, Guang Cheng

May 2020

Introduction

Introduction i

Since the creation of AdaGrad (Duchi et al., 2011), adaptive algorithms have become a heated topic. However, no real progress has been made.

Introduction ii

Algorithm 1 Generic Adaptive Optimization Algorithm

Input: $x_0 \in \mathcal{F}$, sequence of step sizes $\{\alpha_t\}_{t=1}^{T}$
Moment functions: $\{\phi_t, \psi_t\}_{t=1}^{T}$
Initialize $m_0 = 0, V_0 = 0$
for $t = 1$ **to** T **do**
    $g_t = \nabla f_t(x_t)$
    $m_t = \phi_t(g_1, g_2, \ldots, g_t), V_t = \psi_t(g_1, g_2, \ldots, g_t)$
    $x_{t+1} = \Pi_{\mathcal{F}, \sqrt{V_t}}(x_t - \alpha_t m_t / \sqrt{V_t})$
end for

m_t – First moment
V_t – Second moment

Most algorithms design m_t as follows.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \text{(e.g. } \beta_1 = 0.9) \tag{1}$$

How about V_t?

Tabelle 1: Comparisons of different designs of the second moment

	AdaGrad (Duchi et al., 2011)	Adam (Kingma & Ba, 2015)
V_t	$\mathrm{diag}(\sum_{i=1}^{t} g_i^2/t)$	$\mathrm{diag}((1-\beta_2)v_{t-1}+\beta_2 g_t^2)$
	NosAdam (Huang et al., 2019)	AdaX (Li et al., 2020)
V_t	$\mathrm{diag}((1-\beta_{2t})v_{t-1}+\beta_{2t} g_t^2)$	$\mathrm{diag}((1+\beta_2)v_{t-1}+\beta_2 g_t^2)/C_t$

A lot more other algorithms also exist such as AdaDelta (Zeiler, 2012), AMSGrad (Reddi et al., 2018), AdaShift (Zhou et al., 2019), AdaBound (Luo et al., 2019), Radam (Liu et al., 2019)...

Introduction v

Why? Real quotes from the paper of NosAdam (Huang et al., 2019).

The true role of V_t, however, remains a mystery...

This blind shooting process should come to an end.

Background

The composite mirror descent rule

$$x_{t+1} = \operatorname{argmin}\{\alpha_t \langle g_t, x\rangle + \alpha_t \phi(x) + B_{\psi_t}(x, x_t)\}, \tag{2}$$

g_t-gradient, α_t-step size, $\phi(x)$-regularization term

Also, ψ_t is a strongly convex and differentiable function, named as the *proximal function*.

$B_{\psi_t}(x, x_t)$ is the Bregman divergence associated with ψ_t defined as

$$B_{\psi_t}(x, y) = \psi_t(x) - \psi_t(y) - \langle \nabla \psi_t(y), x - y\rangle. \tag{3}$$

Background ii

Examples when $\phi(x) = 0$

- When $\psi_t(x) = \langle x, x \rangle$, $B_{\psi_t}(x, x_t) = \langle x - x_t, x - x_t \rangle$ and

$$x_{t+1} = x_t - \frac{\alpha_t}{2} g_t. \text{ (SGD)}$$

- When $\psi_t(x) = \langle x, H_t x \rangle$, $B_{\psi_t}(x, x_t) = \langle x - x_t, H_t(x - x_t) \rangle$ and

$$x_{t+1} = x_t - \frac{\alpha_t}{2} H_t^{-1} g_t. \text{ (Adaptive)}$$

Background iii

Definition of regret $R(T)$ in the online learning framework

$$R(T) = \sum_{t=1}^{T} f_t(x_t) - f_t(x^*) + \phi(x_t) - \phi(x^*). \tag{4}$$

where $\{f_t\}_{t=1}^{T}$ are a sequence of loss functions and $\phi(x)$ is the regularization term.

Goal: Find algorithm with sublinear regret $R(T) = o(T)$

Adaptive Proximal Functions

Theorem

Let the sequence $\{x_t\}$ be defined by the composite mirror update rule (2) and for any x^*, denote $D_{t,\infty}^2 = \|x_t - x^*\|_\infty^2$. When $\psi_t(x) = \langle x, H_t x \rangle$, where $H_t = diag(h_{t,1}, h_{t,2}, \cdots, h_{t,d})$, assume without loss of generality that $\phi(x_1) = 0, H_0 = 0$, then

$$\sum_{t=1}^{T} f_t(x_t) - f_t(x^*) + \phi(x_t) - \phi(x^*) \leq \sum_{t=1}^{T} \sum_{i=1}^{d} \left(\frac{h_{t,i}}{\alpha_t} - \frac{h_{t-1,i}}{\alpha_{t-1}}\right) D_{t,\infty}^2 + \sum_{t=1}^{T} \sum_{i=1}^{d} \frac{\alpha_t g_{t,i}^2}{2 h_{t,i}} \tag{5}$$

Adaptive Proximal Functions (Diagonal) ii

Duchi et al. (2011) believed that the second term occupied the major part of the regret bound and proposed to find the hind-sight solution of the optimization problem below.

$$h = \operatorname{argmin} \sum_{t=1}^{T} \sum_{i=1}^{d} \frac{g_{t,i}^2}{h_i} \text{ s.t. } h \succeq 0, \langle 1, h \rangle \leq c \tag{6}$$

The solution is $h \propto \sqrt{g_1^2 + g_2^2 + \cdots g_T^2}$. They therefore defined their

$$h_t/\alpha_t \propto \sqrt{g_1^2 + g_2^2 + \cdots g_t^2}$$

which leads to the global average design when $\alpha_t = \alpha/\sqrt{t}$.

Adaptive Proximal Functions (Diagonal) iii

However, the above intuition is not straight-forward. Take a look at the regret again.

$$\sum_{t=1}^{T} f_t(x_t) - f_t(x^*) + \phi(x_t) - \phi(x^*) \leq \sum_{t=1}^{T}\sum_{i=1}^{d}(\frac{h_{t,i}}{\alpha_t} - \frac{h_{t-1,i}}{\alpha_{t-1}})D_{t,\infty}^2 + \sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\alpha_t g_{t,i}^2}{2h_{t,i}} \tag{7}$$

- The first term is $O(\sqrt{T})$ when $\alpha_t = \alpha/\sqrt{t}$. All the optimization algorithms (SGD, AdaGrad...) have regret bound of $O(\sqrt{T})$.
- Why directly changing the T to small t provides the best solution at time t?

However, it's impossible to find the best solutions as future information is always unknown.

Adaptive Proximal Functions (Diagonal) iv

Nevertheless, we can find the best solution at the moment.

We propose to find a *greedy* choice of h_t such that it minimizes the marginal increment of the regret bound. (Inspired by greedy algorithms)

At each time step t, $h_1, h_2, \cdots h_{t-1}$ have been determined. The regret bound is only related to the choice of h_T. Therefore the *marginal regret bound minimization* problem is to find h_T that

$$h_t = \text{argmin}_{h_t} \sum_{i=1}^{d} (D_{t,\infty}^2 \frac{h_{t,i}}{\alpha_t} + \frac{\alpha_t g_{t,i}^2}{2h_{t,i}}), \ \text{s.t.} \frac{h_{t,i}}{\alpha_t} \geq \frac{h_{t-1,i}}{\alpha_{t-1}} \tag{8}$$

Adaptive Proximal Functions (Diagonal) v

The solution is

$$h_t = \max\{\sqrt{\frac{t-1}{t}}h_{t-1}, \frac{\alpha_t}{\sqrt{2}D_{t,\infty}}|g_t|\}. \tag{9}$$

Let $c_t = \alpha_t^2/2D_{t,\infty}^2$, this is our proposed best choice of h_t at the moment.

Adaptive Proximal Functions (Full)

For full matrix adaptive proximal functions, a similar result can be proved.

$$H_t = \max\{\sqrt{\frac{t-1}{t}} H_{t-1}, \frac{\alpha_t}{\sqrt{2} D_{t,\infty}} (g_t g_t^T)^{1/2}\}. \tag{10}$$

where the max operation here is to compare the traces of the two matrices

Our Algorithm

Our algorithm i

Algorithm 2 AMX Algorithm with Momentum (Diagonal)

Input: $x \in \mathcal{F}$, $\{\alpha_t\}_{t=1}^{T}, \{c_t\}_{t=1}^{T}, \{\beta_{1t}\}_{t=1}^{T}$
Initialize $m_0 = 0, h_0 = 0$
for $t = 1$ **to** T **do**
  $g_t = \nabla f_t(x_t)$
  $m_t = \beta_{1t} m_{t-1} + (1 - \beta_{1t}) g_t$
  $h_t = \sqrt{\max(\frac{t-1}{t} h_{t-1}^2, c_t g_t^2)} + \epsilon$
  $H_t = \mathrm{diag}(h_{t,1}, h_{t,2}, \cdots, h_{t,d})$
  $x_{t+1} = \Pi_{\mathcal{F}, H_t}(x_t - \alpha_t m_t / h_t)$
end for

Our algorithm ii

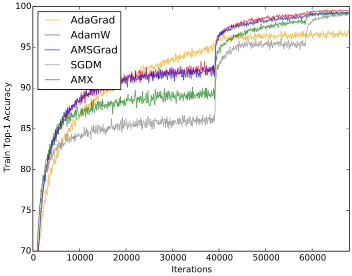
How to get c_t's? - Not easy because $D_{t,\infty}$'s are unknown

We use $c_t = 1$ in our implementation because both α_t and $D_{t,\infty}$ are decreasing and it is easy to implement.

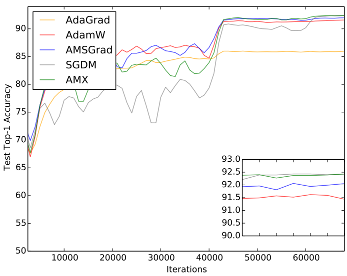
In the future, researchers can simply modify c_t's to find better algorithms, instead of shooting blindly. Future Directions

Experiments

Experiments i



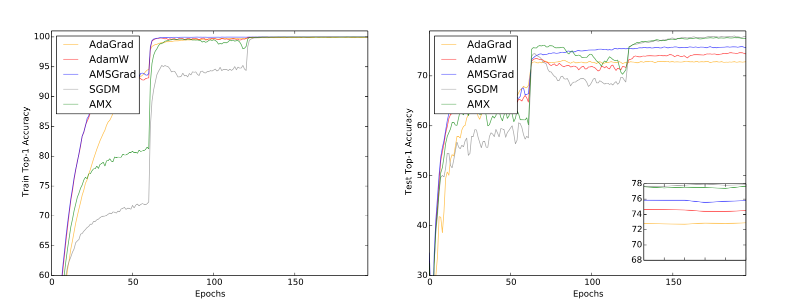
(a) CIFAR-10 Training Acc.



(b) CIFAR-10 Testing Acc.

Abbildung 1: Training and Testing Top-1 accuracy on CIFAR-10

Experiments ii

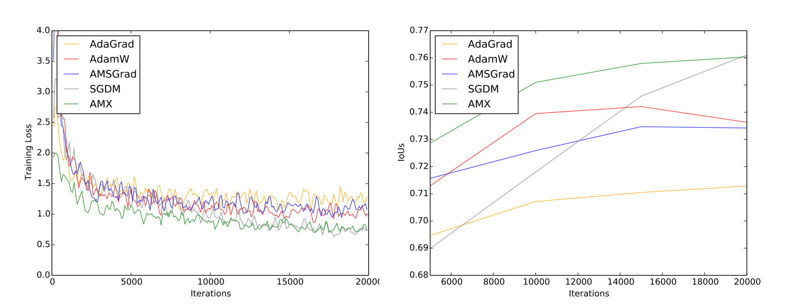


(a) CIFAR-100 Training Acc.

(b) CIFAR-100 Testing Acc.

Abbildung 2: Training and Testing Top-1 accuracy CIFAR-100.

Experiments iii

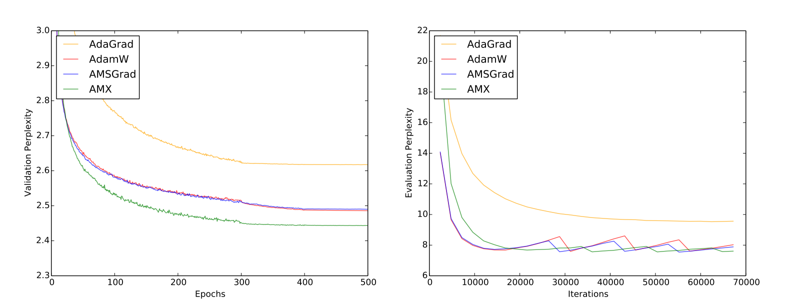


(a) VOC Training Loss Curve

(b) VOC Testing IoU

Abbildung 3: Training Loss and Testing IoU curves on the VOC2012 Segmentation dataset.

Experiments iv



(a) PTB Test Perplexity

(b) IWSLT'14 Test Perplexity

Abbildung 4: (a) Validation perplexity curve on the Penn Tree Bank (PTB) dataset. (b). Validation Perplexity curve on the IWSLT'14 DE-EN machine translation dataset.

This is it ☺

Literatur

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research (JMLR)*, pp. 12:2121–2159, 2011.

Haiwen Huang, Chang Wang, and Bin Dong. Nostalgic adam: Weighting more of the past gradients when designing the adaptive learning rate. *arXiv preprint arXiv: 1805.07557*, 2019.

Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

Wenjie Li, Zhaoyang Zhang, Xinjiang Wang, and Ping Luo. Adax: Adaptive gradient descent with exponential long term memory. *arXiv preprint arXiv:2004.09740*, 2020.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *Proceedings of 7th International Conference on Learning Representations*, 2019.

Sashank J. Reddi, Stayen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

Matthew D. Zeiler. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Zhiming Zhou, Qingru Zhang, Guansong Lu, Hongwei Wang, Weinan Zhang, and Yong Yu. Adashift: Decorrelation and convergence of adaptive learning rate methods. *Proceedings of 7th International Conference on Learning Representations (ICLR)*, 2019.