doi: 10.7690/bgzdh.2023.11.012

基于 AdaGrad 自适应 DA 方法的最优个体收敛速率

张　旭，韦洪旭
（中国人民解放军陆军炮兵防空兵学院信息工程系，合肥 230031）

摘要：针对 AdaGrad 将自适应矩阵应用到随机梯度下降法中降低工程上超参数搜索的问题，提出一种自适应对偶平均方法。将 AdaGrad 自适应矩阵引入到对偶平均方法框架中，形成自适应的对偶平均方法，并通过凸优化实验验证其可行性和收敛效果。数学推导结果表明：对于非光滑条件下的一般凸函数 AdaDA 方法可以达到与维数相关 $O(1/\sqrt{t})$的最优个体收敛速率，为其提供了理论支撑。

关键词：优化算法；梯度下降；对偶平均方法；AdaGrad；自适应矩阵
中图分类号：TP273　　**文献标志码**：A

Optimal Individual Convergence Rate Based on AdaGrad Adaptive DA Method

Zhang Xu, Wei Hongxu
(*Department of Information Engineering*, *PLA Army Academy of Artillery and Air Defense*, *Hefei* 230031, *China*)

Abstract: An adaptive dual averaging method is proposed to solve the problem of AdaGrad's application of adaptive matrices to the random gradient descent method, which reduces the search for hyperparameters in engineering. The AdaGrad adaptive matrix is introduced into the dual averaging method framework to form an adaptive dual averaging method, and its feasibility and convergence effect are verified through convex optimization experiments. The mathematical derivation results show that the AdaDA method for general convex functions under nonsmooth conditions can achieve an optimal individual convergence rate related to dimension $O(1/\sqrt{t})$, providing theoretical support for it.

Keywords: optimization algorithm; gradient descent; dual average method; AdaGrad; adaptive matrix

0　引言

当前，学者对于深度学习中优化算法领域的研究主流方向之一为自适应方法，用以解决高维问题所伴随超参数搜索的高昂代价。熟知的自适应方法有 AdaGrad[1]、Adadelta[2]、RMSProp[3]、Adam[4]。其中，作为第 1 个自适应优化方法的 AdaGrad 所使用的对角矩阵策略一直以来被认定为行业标准和典范，Adadelta、RMSProp、Adam 也仅仅是对 AdaGrad 针对实验效果进行适当的改进，并未触碰和更改其对角矩阵的核心理论思想。

随机梯度下降法（stochastic gradient descent，SGD）和对偶平均方法（dual average，DA）在特定步长选择下是等价的，这说明 AdaGrad 自适应矩阵应用到对偶平均方法中存在理论的可能性。此外，对偶平均方法较随机梯度下降法具备较高的收敛稳定性，源于其使用历史梯度的加权平均方式代替传统随机梯度下降法的单一当前梯度信息。采用加权平均的对偶平均方法通过对历史梯度信息步长的衰减、缩小，巧妙地突出、放大了当前梯度信息的权重，克服了随机梯度下降法由于步长不断衰减所导致的后期收敛缓慢的固有弊端。鉴于上述算法的异同点，说明自适应对偶平均方法的设计不能只依赖自适应矩阵的简单移植，还需要保留该方法对历史梯度信息权重衰减的特性。AdaDA 方法以此为出发点，替代 AdaGrad 中自适应矩阵的平方根转而使用立方根，成功地保留了对偶平均方法的优势，并达到了对梯度不同维度应用适应步长的自适应效果，实现对偶平均方法在自适应领域的扩展。

此外，收敛性分析是评价算法的重要指标之一。针对一般凸优化问题，Zinkevich[5]提出的 OGD 证明在线学习模式下梯度有界的可微凸函数流可以保证 $O(\sqrt{t})$ 的 Regret bound。在非光滑强凸情形中，Hazan 等[6]在 OGD 基础上得到了在线学习模式下更好的 $O(\log(T))$的 Regret bound。通过 Regret bound 可以进一步得到该算法以所有迭代平均方式作为输出时的收敛速率（即平均收敛速率），二者之间为 $1/t$ 的倍数关系。然而相比平均收敛速率，往往更关注的是单步迭代作为输出时的收敛速率（个体收敛速

收稿日期：2023-07-11；**修回日期**：2023-08-05
基金项目：国家自然科学基金（62076252）
第一作者：张　旭（1994—），男，安徽人。

率)，并且个体解更利于保持机器学习正则化项的结构。同时，对于凸优化问题，从个体解的最优收敛性能够方便获得最优平均收敛性，但反之却不成立。Shamir 等[7]提出 SGD 由平均收敛速率直接得到个体收敛速率的一般技巧，得到 SGD 非光滑条件下强凸问题 $O(\ln t/t)$和一般凸问题 $O(\ln t/\sqrt{t})$的个体收敛速率，相比强凸问题 $O(1/t)$和一般凸问题 $O(1/\sqrt{t})$最优平均收敛速率，仅相差一个对数因子。Nichdas 等[8]证明对数因子必不可少，即 SGD 关于一般凸问题和强凸问题的最优个体收敛速率就是 $O(\ln t/t)$和 $O(\ln t/\sqrt{t})$。2011 年，Duchi 等提出 AdaGrad 算法，可以看作是 SGD 与自适应步长策略的首次结合。John 等[1]证明在线 AdaGrad 在处理一般凸问题时具 $O(\sqrt{t})$有 Regretbound，达到了和 SGD 一样的最优收敛速率，但在处理稀疏数据时 AdaGrad 算法却能获得比 SGD 更小因子的收敛界。笔者提出一种 AdaDA 算法，将 AdaGrad 自适应策略与 DA 方法相结合，对于非光滑条件下一般凸问题，同样可以得到与维数相关的 $O(1/\sqrt{t})$最优个体收敛速率。

1 AdaDA 算法设计分析

通过分析 DA 和 SGD 算法的差别，说明向 DA 中引入 AdaGrad 自适应矩阵的突破点，并作相应改进以保留 DA 的算法优势。首先对符号进行明确，k 表示算法的迭代步骤，g_k 表示凸函数 $f(x)$在 x_k 处的梯度 $\nabla f(x_k)$ 或凸函数 $f(x)$在 x_k 处的次梯度，即 $g_k \in \partial f(x_k)$。

1.1 SGD 和 DA 的比较

以基本型的 DA 方法与 SGD 作比较如下式所示，其中 λ_i=1，$\gamma_i = 1/\sqrt{i+1}$，在 $\beta_{k+1} = \sqrt{k+1}$时 2 种方法达到收敛。

$$\mathrm{SGD}: x_{k+1} = x_0 - \sum_{i=0}^{k} \gamma_i \boldsymbol{g}_i ; \tag{1}$$

$$\mathrm{DA}: x_{k+1} = x_0 - \frac{1}{\beta_{k+1}} \sum_{i=0}^{k} \lambda_i \boldsymbol{g}_i 。 \tag{2}$$

不难发现，式(1)中 SGD 的不同迭代步骤梯度信息权重 $\gamma_i = 1/\sqrt{i+1}$ 为变步长，并随着迭代步骤增加而衰减，而 DA 方法历史梯度和当前梯度权重全部相同，为固定的 $1/\beta_{k+1} = 1/\sqrt{k+1}$。式(2)中 DA 对历史梯度及当前梯度应用的权重恰好为 SGD 在第 k 次迭代中使用的 γ_i 步长。

当 $\lambda_i = (i+1)^{1/2}\gamma_i$ 时，SGD 依旧取步长 γ_i，式(3)中 DA 的形式并未发生改变，DA 可以看作对梯度信息使用 SGD 相同的步长，但对 k 次迭代之前的步长 γ_i 给予 $(i+1)^{1/2}/\sqrt{k+1}$ 的衰减，巧妙地“关注”最新的梯度信息。DA 的步长衰减正是相较 SGD 的特性所在，是本文中方法所考虑保留的。

$$x_{k+1} = x_0 - \frac{1}{\sqrt{k+1}} \sum_{i=0}^{k} (i+1)^{1/2} \gamma_i \boldsymbol{g}_i = x_0 - \gamma_k g_k - \frac{1}{\sqrt{k+1}} \sum_{i=0}^{k-1} (i+1)^{1/2} \gamma_i \boldsymbol{g}_i 。 \tag{3}$$

1.2 自适应矩阵的引入和 AdaDA 算法

AdaGrad 在 SGD 上应用自适应矩阵可以用式(4)表示：

$$x_{k+1} = x_k - \gamma_k \boldsymbol{v}_k^{-1/2} g_k$$
$$\boldsymbol{v}_k = \sum_{i=1}^{t} \boldsymbol{g}_i^2 \Big/ k 。 \tag{4}$$

式中：$\boldsymbol{g}_i^2 = \mathrm{diag}(\boldsymbol{g}_i \boldsymbol{g}_i^{\mathrm{T}})$ 为计算第 i 步梯度 g_i 的外积矩阵对角化后的对角矩阵；$\boldsymbol{v}_k$ 为 $d\times d$ 维(d 是 g 的维数)的自适应矩阵，其元素为 k 步之前所有对角外积矩阵对应位置元素的算术平均值。

AdaGrad 改变了以往算法在应用梯度信息权重上只区分迭代步骤的惯例，而更进一步细化到梯度信息的每一维度信息的差异。这种差异对于稀疏数据更为常见，样本在不同特征(梯度的不同维度)的变化速度存在较大差别，有快有慢，需要自适应的权重调整其变化速度。

根据 1.1 节中的比较分析，取 $\lambda_i = (i+1)^{1/2}\gamma_i$，以 AdaGrad 中第 k 次的步长直接应用于 DA，做最简单的移植后得到自适应的 DA 如式(5)所示，并不能保留 $(i+1)^{1/2}/\sqrt{k+1}$ 的步长衰减。

$$x_{k+1} = x_0 - \sum_{i=0}^{k} (i+1)^{1/2} \gamma_i \boldsymbol{g}_i \Big/ \sqrt{\sum_{i=0}^{k} (i+1)^{1/2} \gamma_i \boldsymbol{g}_i^2} 。 \tag{5}$$

本文中 AdaDA 方法通过式(6)在 DA 中应用 AdaGrad 自适应矩阵：

$$\boldsymbol{x}_{k+1} = x_0 - \sum_{i=0}^{k} (i+1)^{1/2} \gamma_i \boldsymbol{g}_i \Big/ \sqrt[3]{\sum_{i=0}^{k} (i+1)^{1/2} \gamma_i \boldsymbol{g}_i^2} 。 \tag{6}$$

式(6)相较于式(5)的“简单移植”仅仅在分母中使用立方根替代平方根，立方根代替平方根的好处在于 $\sqrt[3]{\sum_{i=0}^{k} (i+1)^{1/2}} \propto (k+1)^{1/2}$，进而达到步长衰减效果。

最后，使用 Nesterov 在文献[9]中提出的 Double Averaging 技巧（类似于指数平均）更好地提高算法的泛化能力，给出本文中的自适应对偶平均方法 AdaDA。

AdaDA 完整迭代过程如下：

$$\boldsymbol{s}_{k+1}=\boldsymbol{s}_k+\lambda_k\boldsymbol{g}_k;$$

$$\boldsymbol{v}_{k+1}=\boldsymbol{v}_k+\lambda_k\boldsymbol{g}_k^2;$$

$$\boldsymbol{z}_{k+1}=\boldsymbol{x}_0-\boldsymbol{s}_{k+1}\Big/\sqrt[3]{\boldsymbol{v}_{k+1}};$$

$$\boldsymbol{x}_{k+1}=(1-c_{k+1})\boldsymbol{x}_k+c_{k+1}\boldsymbol{z}_{k+1}。$$

式中 $\lambda_k=(k+1)^{1/2}\gamma$，$\gamma$ 为固定步长，使用 c_{k+1} 实现 Double Averaging。

AdaDA 算法具体执行步骤：

输入循环次数 T

初始化 $\boldsymbol{x}_0, c_k, \gamma$

1) $\boldsymbol{s}_0=0, \boldsymbol{v}_0=0$；

2) for $k=0,\cdots,T$ do；

3) 计算梯度 $\boldsymbol{g}_k$；

4) $\lambda_k=(k+1)^{1/2}\gamma$；

5) $\boldsymbol{s}_{k+1}=\boldsymbol{s}_k+\lambda_k\boldsymbol{g}_k$；

6) $\boldsymbol{v}_{k+1}=\boldsymbol{v}_k+\lambda_k\boldsymbol{g}_k^2$；

7) $\boldsymbol{z}_{k+1}=\boldsymbol{x}_0-\boldsymbol{s}_{k+1}\Big/\sqrt[3]{\boldsymbol{v}_{k+1}+\varepsilon}$；

8) $\boldsymbol{x}_{k+1}=(1-c_{k+1})\boldsymbol{x}_k+c_{k+1}\boldsymbol{z}_{k+1}$；

9) end for；

10) return x_T。

2　AdaDA 算法个体收敛性分析

在收敛性证明上，借鉴 Duchi 在文献[1]中的技巧，在分母矩阵中添加项 $\lambda_k\boldsymbol{G}^2$，其中 $\boldsymbol{G}$ 为梯度有界假设：$\boldsymbol{g}_i\leqslant\boldsymbol{G},\forall i$。

$$\boldsymbol{s}_{k+1}=\boldsymbol{s}_k+\lambda_k\boldsymbol{g}_k;$$

$$\boldsymbol{v}_{k+1}=\boldsymbol{v}_k+\lambda_k\boldsymbol{g}_k^2;$$

$$\boldsymbol{z}_{k+1}=\boldsymbol{x}_0-\boldsymbol{s}_{k+1}\Big/\sqrt[3]{\lambda_{k+1}\boldsymbol{G}^2+\boldsymbol{v}_{k+1}};$$

$$\boldsymbol{x}_{k+1}=(1-c_{k+1})\boldsymbol{x}_k+c_{k+1}\boldsymbol{z}_{k+1}。$$

另外，参照 Nesterov 在文献[10]中的设置，构造如下的辅助函数 $V_{A_k}(\boldsymbol{s}_k)$：

$$V_{A_k}(\boldsymbol{s}_k)=\max_x\{\langle\boldsymbol{s}_k,\boldsymbol{x}-\boldsymbol{x}_0\rangle-\|\boldsymbol{x}-\boldsymbol{x}_0\|_{A_k}^2\Big/2\}。$$

式中：$\boldsymbol{s}_k=\sum_{i=0}^{k-1}\lambda_i\boldsymbol{g}_i$；$A_k=\mathrm{diag}(a_k)$；$a_k=\sqrt[3]{\lambda_k\boldsymbol{G}^2+\boldsymbol{v}_k}$。

同时，本文中 AdaDA 算法中 $\boldsymbol{z}_k$ 亦可做相似表达：

$$\boldsymbol{z}_k=\mathrm{argmin}_x\left\{-\langle\boldsymbol{s}_k,\boldsymbol{x}\rangle+\|\boldsymbol{x}-\boldsymbol{x}_0\|_{A_k}^2\Big/2\right\}。$$

辅助函数 $V_{A_k}(\boldsymbol{s}_k)$ 具备以下与文献[10]相同的性质（在定理 3 的证明中使用）：

1) $V_{A_{k+1}}(\boldsymbol{s}_k)\leqslant V_{A_K}(\boldsymbol{s}_k)$；

2) $V_{A_{k+1}}(\boldsymbol{s}_k+\boldsymbol{\delta})\leqslant V_{A_K}(\boldsymbol{s}_k)+\langle\boldsymbol{\delta},\nabla V_{A_K}(\boldsymbol{s})\rangle+\dfrac{\|\boldsymbol{\delta}\|_{A_k^{-1}}^2}{2}$；

3) $\nabla V_{A_K}(-\boldsymbol{s}_k)=\boldsymbol{z}_k-\boldsymbol{x}_0$。

2.1　引理 1

对于所有的自然数 k，如果 $\lambda_{k+1}\geqslant\lambda_k$，那么

$$\sum_{t=0}^{k}\frac{\lambda_t^2g_t^2}{(\lambda_tG^2+\sum_{i=0}^{t-1}\lambda_ig_i^2)^{1/3}}\leqslant\frac{3}{2}\lambda_k(\sum_{i=0}^{k}\lambda_ig_i^2)^{2/3}。$$

证明：

当 k=0 时，因为 $g_0^2\leqslant G^2$，

$\dfrac{\lambda_0^2g_0^2}{(\lambda_0G^2)^{1/3}}\leqslant\lambda_0^{5/3}g_0^{2(1-1/3)}=\lambda_0^{5/3}g_0^{2(2/3)}\leqslant\dfrac{3}{2}\lambda_0(\lambda_0g_0^2)^{2/3}$ 满足引理 1；

假设 k−1 时，引理成立。

$$\sum_{t=0}^{k}\frac{\lambda_t^2g_t^2}{(\lambda_tG^2+\sum_{i=0}^{t-1}\lambda_ig_i^2)^{1/3}}=\frac{\lambda_k^2g_k^2}{(\lambda_kG^2+\sum_{i=0}^{k-1}\lambda_ig_i^2)^{1/3}}+$$

$$\sum_{t=0}^{k-1}\frac{\lambda_t^2g_t^2}{(\lambda_tG^2+\sum_{i=0}^{t-1}\lambda_ig_i^2)^{1/3}}\leqslant\frac{\lambda_k^2g_k^2}{(\lambda_kG^2+\sum_{i=0}^{k-1}\lambda_ig_i^2)^{1/3}}+$$

$$\frac{3}{2}\lambda_{k-1}(\sum_{i=0}^{k-1}\lambda_ig_i^2)^{2/3}\leqslant\frac{\lambda_k^2g_k^2}{(\lambda_kG^2+\sum_{i=0}^{k-1}\lambda_ig_i^2)^{1/3}}+$$

$$\frac{3}{2}\lambda_k(\sum_{i=0}^{k-1}\lambda_ig_i^2)^{2/3}。$$

令 $a_k=g_k^2, b_k=\sum_{i=0}^{k}\lambda_ig_i^2$ 代入上式得：

$$\sum_{t=0}^{k}\frac{\lambda_t^2g_t^2}{(\lambda_tG^2+\sum_{i=0}^{t-1}\lambda_ig_i^2)^{1/3}}\leqslant\lambda_k^2a_k(\lambda_kG^2+b_k-$$

$$\lambda_ka_k)^{-1/3}+\frac{3}{2}\lambda_k(b_k-\lambda_ka_k)^{2/3}。$$

由于 $a_k=g_k^2\leqslant G^2$，不等式右侧第 1 项

$$\lambda_k^2a_k(\lambda_kG^2+b_k-\lambda_ka_k)^{-1/3}\leqslant\lambda_k^2a_kb_k^{-1/3};$$

又由凸性可得，不等式右侧第 2 项

$\frac{3}{2}\lambda_k(b_k-\lambda_k a_k)^{2/3}=\frac{3}{2}\lambda_k b_{k-1}^{2/3}\leqslant\frac{3}{2}\lambda_k b_k^{2/3}-\lambda_k^2 a_k b_k^{-1/3}$。

进而得到

$$\sum_{t=0}^{k}\frac{\lambda_t^2 g_t^2}{(\lambda_t G^2+\sum_{i=0}^{t-1}\lambda_i g_i^2)^{1/3}}\leqslant\frac{3}{2}\lambda_k b_k^{2/3}=\frac{3}{2}\lambda_k(\sum_{i=0}^{k}\lambda_i g_i^2)^{2/3}。$$

引理 1 得证。

2.2 引理 2

当 $0<m<1$ 时，对于 $c_k=\frac{m+1}{k+m+n}$，当 $k\geqslant 0$ 时，以下不等式成立：

$$\frac{1-c_k}{c_k}(k+n)^m\leqslant\frac{1}{c_{k-1}}(k+n-1)^m。$$

证明：

对于 $m\in(0,1)$ 的凸函数 $f(x)=x^m$，令 $x=k+n, y=k+n-1$，可以得到：

$$(k+n)^m\leqslant(k+n-1)^m+m(k+n-1)^{m-1}=$$
$$(k+n-1)^m+\frac{m}{k+n-1}(k+n-1)^m=$$
$$\frac{k+n-1+m}{k+n-1}(k+n-1)^m;$$

原式 $\frac{1-c_k}{c_k}(k+n)^m=\frac{1-(m+1)/(k+m+n)}{(m+1)/(k+m+n)}(k+n)^m=(k+n)^m(k+n-1)/(m+1)=(k+m+n-1)(k+n-1)(k+n)^m/(m+1)(k+m+n-1)=(k+n-1)(k+n)^m/(k+m+n-1)c_{k-1}\leqslant(k+n-1)^m/c_{k-1}$。

引理 2 得证。

2.3 定理 3

当 k=0 时，$V_{A_1}(-s_1)\leqslant\lambda_0^2\|\nabla f(x_0)\|_{A_0^{-1}}^2/2$；

当 $k\geqslant 1$ 时，

$$V_{A_{k+1}}(-s_{k+1})\leqslant V_{A_k}(-s_k)+\frac{\lambda_k^2}{2}\|\nabla f(x_k)\|_{A_k^{-1}}^2+$$
$$\lambda_k\langle\nabla f(x_k),x_0-x_*\rangle-\frac{1}{c_k}\lambda_k[f(x_k)-f(x_*)]+$$
$$\lambda_k[f(x_{k-1})-f(x_*)](1-c_k)/c_k。$$

证明：

当 k=0 时，由 $V_{A_k}(s_k)$ 的性质 2 得

$$V_{A_1}(-s_1)\leqslant-\lambda_0\langle\nabla f(x_0),\nabla V_0(s_0)\rangle+\frac{\lambda_0^2}{2}\|\nabla f(x_0)\|_{A_0^{-1}}^2=$$
$$\lambda_k\langle\nabla f(x_k),x_0-x_0\rangle+\frac{\lambda_0^2}{2}\|\nabla f(x_0)\|_{A_0^{-1}}^2=\frac{\lambda_0^2}{2}\|\nabla f(x_0)\|_{A_0^{-1}}^2。$$

当 $k\geqslant 1$ 时，

$$V_{A_{k+1}}(-s_{k+1})\leqslant V_{A_K}(-s_{k+1})\leqslant V_{A_K}(-s_k)-$$
$$\lambda_k\langle\nabla f(x_k),\nabla V_{A_k}(-s_k)\rangle+\frac{\lambda_k^2}{2}\|\nabla f(x_k)\|_{A_K^{-1}}^2=$$
$$V_{A_K}(-s_k)+\lambda_k\langle\nabla f(x_k),x_0-z_k\rangle+\frac{\lambda_k^2}{2}\|\nabla f(x_k)\|_{A_K^{-1}}^2=$$
$$V_{A_K}(-s_k)+\frac{\lambda_k^2}{2}\|\nabla f(x_k)\|_{A_K^{-1}}^2+\lambda_k\left\langle\nabla f(x_k),x_0-x_k+\frac{1-c_k}{c_k}(x_{k-1}-x_k)\right\rangle=V_{A_K}(-s_k)+\frac{\lambda_k^2}{2}\|\nabla f(x_k)\|_{A_K^{-1}}^2+$$
$$\lambda_k\langle\nabla f(x_k),x_0-x_k\rangle+\lambda_k\frac{1-c_k}{c_k}\langle\nabla f(x_k),(x_{k-1}-x_k)\rangle=$$
$$V_{A_K}(-s_k)+\frac{\lambda_k^2}{2}\|\nabla f(x_k)\|_{A_K^{-1}}^2+\lambda_k\langle\nabla f(x_k),x_0-x_*\rangle+$$
$$\lambda_k\langle\nabla f(x_k),x_*-x_k\rangle+\lambda_k\frac{1-c_k}{c_k}\langle\nabla f(x_k),(x_{k-1}-x_k)\rangle。$$

式中：$\langle\nabla f(x_k),x_*-x_k\rangle\leqslant f(x_*)-f(x_k)$；$\langle\nabla f(x_k),(x_{k-1}-x_k)\rangle\leqslant f(x_{k-1})-f(x_k)$。

上式进一步展开得到：

$$V_{A_{k+1}}(-s_{k+1})\leqslant V_{A_K}(-s_k)+\frac{\lambda_k^2}{2}\|\nabla f(x_k)\|_{A_K^{-1}}^2+$$
$$\lambda_k\langle\nabla f(x_k),x_0-x_*\rangle+\lambda_k[f(x_*)-f(x_k)]+$$
$$\lambda_k\frac{1-c_k}{c_k}[f(x_{k-1})-f(x_k)]\leqslant V_{A_K}(-s_k)+$$
$$\frac{\lambda_k^2}{2}\|\nabla f(x_k)\|_{A_K^{-1}}^2+\lambda_k\langle\nabla f(x_k),x_0-x_*\rangle-$$
$$\frac{1}{c_k}\lambda_k[f(x_k)-f(x_*)]+\frac{1-c_k}{c_k}\lambda_k[f(x_{k-1})-f(x_*)]。$$

定理 3 得证。

2.4 定理 4

k 次迭代后，取 $\gamma=\frac{1}{k^{3/4}D^{3/4}G^{1/2}}\|x_0-x_*\|^{3/2}$，$c_k=(3/2)/(k+3/2)$，其中 D 为梯度的维度，那么

$$f(x_k)-f(x_*)\leqslant\frac{6}{k^{1/2}}\|x_0-x_*\|GD^{1/2}。$$

达到与维数相关的 $O(1/\sqrt{k})$ 最优个体收敛速率。

证明：

当 $\lambda_k=\gamma(k+1)^{1/2}$，其中 γ 为常数，取 $c_k=(3/2)/(k+3/2)$，应用引理 2，可以得到 $\lambda_k(1-c_k)/c_k\leqslant\lambda_{k-1}/c_{k-1}$。

进而定理 3 中，当 $k\geqslant 1$ 时：

$$V_{A_{k+1}}(-s_{k+1})\leqslant V_{A_k}(-s_k)+\frac{\lambda_k^2}{2}\|\nabla f(x_k)\|^2_{A_k^{-1}}+$$
$$\lambda_k\langle\nabla f(x_k),x_0-x_*\rangle-\frac{1}{c_k}\lambda_k[f(x_k)-f(x_*)]+$$
$$\frac{1-c_k}{c_k}\lambda_k[f(x_{k-1})-f(x_*)]\leqslant V_{A_k}(-s_k)+$$
$$\frac{\lambda_k^2}{2}\|\nabla f(x_k)\|^2_{A_k^{-1}}+\lambda_k\langle\nabla f(x_k),x_0-x_*\rangle-$$
$$\frac{1}{c_k}\lambda_k[f(x_k)-f(x_*)]+\frac{1}{c_{k-1}}\lambda_{k-1}[f(x_{k-1})-f(x_*)]。$$

上式两端取 $\sum_{t=1}^{k}$ 整理得到

$$\frac{1}{c_k}\lambda_k[f(x_k)-f(x_*)]\leqslant V_{A_1}(-s_1)-V_{A_k}(-s_{k+1})+$$
$$\sum_{t=1}^{k}\frac{\lambda_t^2}{2}\|\nabla f(x_t)\|^2_{A_k^{-1}}+\left\langle\sum_{t=1}^{k}[\lambda_t\nabla f(x_t)],x_0-x_*\right\rangle+$$
$$\frac{1}{c_0}\lambda_0[f(x_0)-f(x_*)]。$$

定理 3 中，当 k=0 时，

$$V_{A_1}(-s_1)\leqslant\lambda_0^2\|\nabla f(x_0)\|^2_{A_0^{-1}}/2$$

$$c_0=1,\frac{1}{c_0}\lambda_0[f(x_0)-f(x_*)]\leqslant\lambda_0\langle\nabla f(x_0),x_0-x_*\rangle。$$

因此：

$$\lambda_k[f(x_k)-f(x_*)]/c_k\leqslant -V_{A_k}(-s_{k+1})+$$
$$\sum_{t=0}^{k}\frac{\lambda_t^2}{2}\|\nabla f(x_t)\|^2_{A_k^{-1}}+\left\langle\sum_{t=1}^{k}[\lambda_t\nabla f(x_t)],x_0-x_*\right\rangle。$$

而根据 $V_{A_k}(\boldsymbol{s}_k)$ 的定义：

$$V_{A_{k+1}}(-s_{k+1})=\max_x\left\{\langle -s_{k+1},x-x_0\rangle-\|x-x_0\|^2_{A_{k+1}}/2\right\}\geqslant$$
$$\langle -s_{k+1},x_*-x_0\rangle-\|x_*-x_0\|^2_{A_{k+1}}/2=\left\langle\sum_{t=}^{k}\lambda_t\nabla f(x_t),x_0-x_*\right\rangle-$$
$$\|x_*-x_0\|^2_{A_{k+1}}/2。$$

继而得到：

$$\frac{1}{c_k}\lambda_k[f(x_k)-f(x_*)]\leqslant\sum_{t=0}^{k}\frac{\lambda_t^2}{2}\|\nabla f(x_t)\|^2_{A_k^{-1}}+\frac{1}{2}\|x_*-x_0\|^2_{A_{k+1}}。$$

$\lambda_k=\gamma(k+1)^{1/2}$，令 $\gamma=\dfrac{1}{k^{3/4}D^{3/4}G^{1/2}}\|x_0-x_*\|^{3/2}$，$c_k=(3/2)/(k+3/2)$，其中 D 为梯度的维度：

$$(k+3/2)/(3/2)\gamma(k+1)^{1/2}[f(x_k)-f(x_*)]\leqslant$$
$$\sum_{t=0}^{k}\frac{\lambda_t^2}{2}\|\nabla f(x_t)\|^2_{A_k^{-1}}+\frac{1}{2}\|x_*-x_0\|^2_{A_{k+1}}。$$

已知维数为 D，且回顾对角矩阵 $\boldsymbol{a}_k=\sqrt[3]{\lambda_k\boldsymbol{G}^2+\boldsymbol{v}_k}$，$A_k=\mathrm{diag}(a_k)$，对上式不等式右侧应用引理 1：

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

由于

$$\sum_{i=0}^{k}(i+1)^{1/2}\leqslant\frac{2}{3}(k+2)^{3/2},g_{id}^2\leqslant G^2；$$

因此：

$$\frac{k+3/2}{3/2}\gamma(k+1)^{1/2}[f(x_k)-f(x_*)]\leqslant$$
$$\frac{1}{2}\sum_{d=0}^{D}\left(\left[\lambda_k\left(\sum_{i=0}^{k}\lambda_i G^2\right)^{2/3}3\Big/2\right]\right)+\frac{1}{2}\sum_{d=0}^{D}(x_{0d}-$$
$$x_{*d})^2(\sum_{i=0}^{k+1}\lambda_i G^2)^{1/3}\leqslant\frac{1}{2}\sum_{d=0}^{D}\left(\left[\frac{3}{2}\gamma(k+1)^{1/2}\left(\sum_{i=0}^{k}\gamma(i+$$
$$1)^{1/2}G^2\right)^{2/3}\right]\right)+\frac{1}{2}\sum_{d=0}^{D}(x_{0d}-x_{*d})^2(\sum_{i=0}^{k+1}\gamma(i+1)^{1/2}G^2)^{1/3}\leqslant$$
$$\frac{1}{2}\sum_{d=0}^{D}\left(\left[\frac{3}{2}\gamma(k+1)^{1/2}\left(\frac{2}{3}(k+2)^{3/2}\gamma G^2\right)^{2/3}\right]\right)+\frac{1}{2}\sum_{d=0}^{D}(x_{0d}-$$
$$x_{*d})^2(\frac{2}{3}(k+3)^{3/2}\gamma G^2)^{1/3}\leqslant\frac{1}{2}\sum_{d=0}^{D}\left(\left[\frac{3}{2}\gamma^{5/3}G^{4/3}(k+\right.\right.$$
$$\left.\left.1)^{1/2}\frac{2}{3}(k+2)\right]\right)+\frac{1}{2}\sum_{d=0}^{D}(x_{0d}-x_{*d})^2\frac{2}{3}(k+3)^{1/2}\gamma^{1/3}G^{2/3}\leqslant$$
$$\frac{1}{2}\sum_{d=0}^{D}\left(\left[\gamma^{5/3}G^{4/3}(k+1)^{1/2}(k+2)\right]\right)+\frac{1}{3}\sum_{d=0}^{D}(x_{0d}-$$
$$x_{*d})^2(k+3)^{1/2}\gamma^{1/3}G^{2/3}\frac{2}{3}[f(x_k)-f(x_*)]\leqslant$$
$$\frac{1}{2}\sum_{d=0}^{D}\gamma^{2/3}G^{4/3}\frac{k+2}{k+3/2}+\frac{1}{3}\sum_{d=0}^{D}(x_{0d}-x_{*d})^2$$
$$(k+3)^{1/2}\gamma^{-2/3}G^{2/3}/(k+3/2)(k+1)^{1/2}。$$

利用 $\dfrac{(k+3)^{1/2}}{(k+3/2)(k+1)^{1/2}}\leqslant\dfrac{2}{k+1}$，$\dfrac{k+2}{k+3/2}\leqslant 2$，上式可以进一步简化为：

$$\frac{2}{3}[f(x_k)-f(x_*)]\leqslant\gamma^{2/3}G^{4/3}D+$$
$$\frac{2}{3(k+1)}\gamma^{-2/3}G^{2/3}\|x_0-x_*\|^2。$$

为后续化简方便，右侧放大如下：

$$f(x_k)-f(x_*)\leqslant 3\gamma^{2/3}G^{4/3}D+\frac{3}{k+1}\gamma^{-2/3}G^{2/3}\|x_0-x_*\|^2。$$

γ 求导得到最优值

$$\gamma^{2/3}=\frac{1}{k^{1/2}D^{1/2}G^{1/3}}\|x_0-x_*\|;$$

$$f(x_k)-f(x_*)\leqslant \frac{6}{k^{1/2}}\|x_0-x_*\|GD^{1/2}。$$

定理 4 得证。

3 实验

通过深度学习实验，并与其他算法对比，进一步检验 AdaDA 算法的可行性与预期效果。

3.1 实验模型和数据集

笔者深度学习实验中使用典型的 ResNet-18 神经网络模型，所采用的 3 个常用标准数据集分别为 CIFAR10(50 000 训练样本、10 000 测试样本)、CIFAR100(50 000 训练样本、10 000 测试样本)、MNIST(60 000 训练样本、10 000 测试样本)。

3.2 比较算法和超参数设置

实验中选取 3 种目前使用较多且效果较好的算法：SGD 算法、AdaGrad 算法、Adam 算法[8]，以及基本型的 DA 方法与本文中的 AdaDA 算法进行对比。DA 算法采用文献[7]中的基本型，其他算法步长及参数设置分别为：SGD 算法使用 $\gamma_k=\gamma/\sqrt{k+1}$；AdaGrad 算法使用 $\gamma_k=\gamma/\sqrt{k+1}$，$\varepsilon=1e-8$；Adam 算法使用 $\gamma_k=\gamma/\sqrt{k+1}$，$\varepsilon=1e-8$，$\beta_1=0.9$，$\beta_2=0.99$；AdaDA 算法使用 $\gamma_k=\gamma/\sqrt{k+1}$，$\varepsilon=1e-8$。对于共同超参数，采取了从{1, 0.1, 0.01, 0.001, 0.000 1}中线性搜索的方式，并取其中最好的一次实验结果，作为该算法的最终输出。为降低随机因素产生的影响，各算法在每个数据集上均运行 5 次，并取平均值作为最后的输出。

3.3 实验效果及结论

图 1—3 分别为 5 种算法在 ResNet-18 神经网络模型上 3 种数据集的测试精度对比，图 4—6 为损失对比。DA 方法较 SGD、AdaGrad 方法在每次迭代中都使用了历史梯度信息，而 SGD、AdaGrad 仅仅使用当前的梯度信息，且由于样本抽取的随机性，梯度方向变化大，收敛振荡明显。从实验结果可以看出，DA 方法框架下的 AdaDA 依旧继承了 DA 方法收敛稳定的特性，较 SGD、AdaGrad 振荡有明显改善，且在测试精度及损失降低方面也达到优于或相当其他自适应方法的效果。

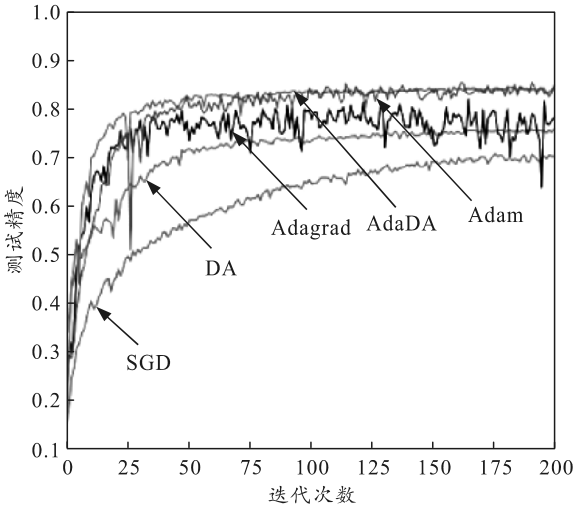

图 1 测试精度对比(CIFAR10)

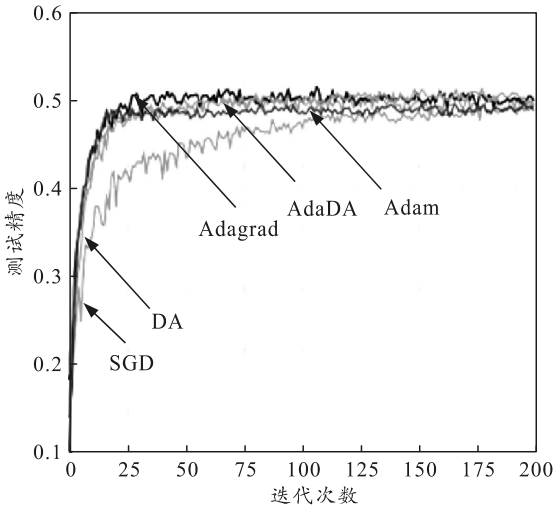

图 2 测试精度对比(CIFAR100)

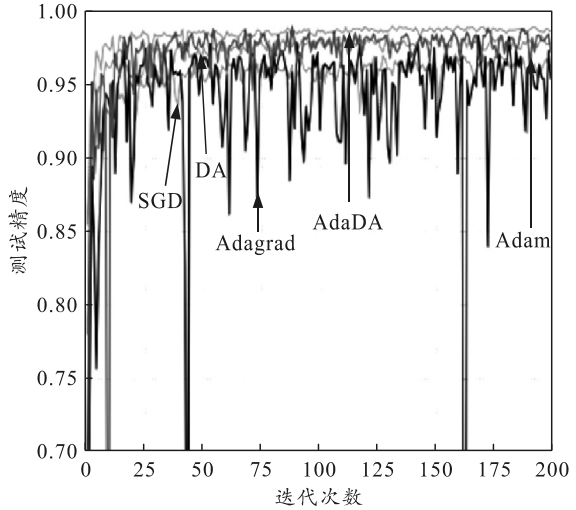

图 3 测试精度对比(MNIST)

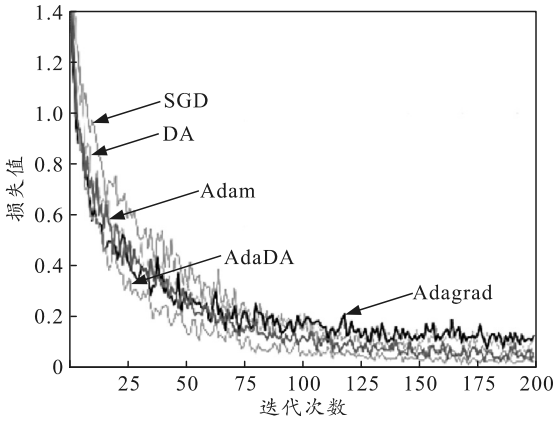

图 4　损失对比(CIFAR10)

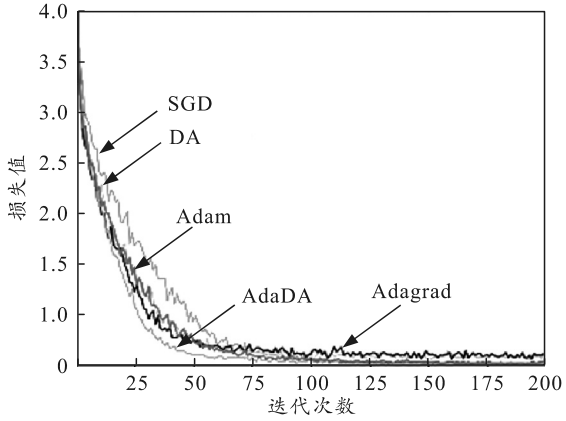

图 5　损失对比(CIFAR100)

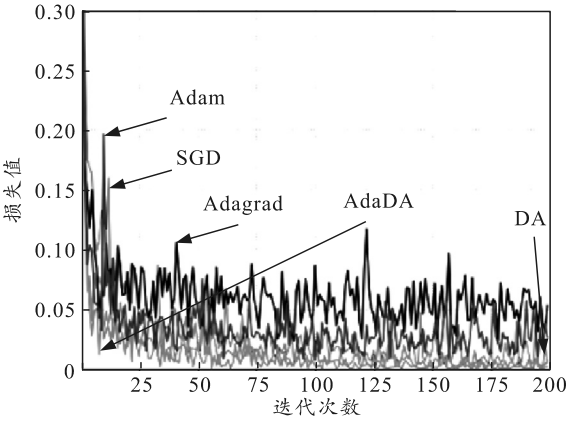

图 6　损失对比(MNIST)

4　结束语

笔者提出一种名为 AdaDA 的自适应对偶平均方法，通过数学推导，证明了该方法对于非光滑条件下的一般凸函数可以达到与维数相关的 $O(1/\sqrt{t})$ 最优个体收敛速率，并通过算法在深度学习上的实验应用，检验其收敛稳定性和精度，达到了预期的实验效果。后续，笔者将继续对 AdaDA 算法在强凸情况下的收敛性进行分析。

参考文献：

[1] JOHN C D, ELAD H. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization[J]. The Journal of Machine Learning Research, 2011, 12: 2121-2159.

[2] MATTHEW D Z. ADADELTA: An Adaptive Learning Rate Method[J]. CoRR abs, 2012(12): 1212. 5701.

[3] TIELEMAN T, HINTON G. RMSProp: Divide the Gradient by a Running Average of its Recent Magnitude[R]. Toronto: University of Toronto, 2012.

[4] KINGMA D P, BA J. Adam: A Method for Stochastic Optimization[C]//Proc of the 3rd IntConf for Learning Representations. San Diego: ICLR, 2015: 1-13.

[5] ZINKEVICH M. Online convex programming and generalized infinitesimal gradient ascent[C]//In: Proc. of the 20th International Conference on Machine Learning (ICML 2003). IEEE, 2003: 928-936.

[6] HAZAN E, KALAI A, KALE S, et al. Logarithmic Regret Algorithms for Online Convex Optimization[C]//Proceedings of the 19th Annual Conference on Learning Theory (COLT). Pittsburgh: COLT, 2006: 499-513.

[7] SHAMIR O, ZHANG T. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes[C]//Proceedings of the 29th International Conference on Machine Learning. ACM, 2013: 71-79.

[8] NICHOLAS J A H, CHRISTOPHER L, YANIV P, et al. Tight analyses for non-smooth stochastic gradient descent[J]. COLT, 2019(1): 1579-1613.

[9] YURII E, NESTER O V, VLADIMIR S. Quasi-monotone Subgradient Methods for Nonsmooth Convex Minimization[J]. J. Optim. Theory Appl, 2015, 165(3): 917-940.

[10] YURII E, NESTER O V. Primal-dual subgradient methods for convex problems[J]. Math Program, 2009, 120(1): 221-259.