

最优化问题中步长越大、收敛速度越快，梯度下降算法数十年的传统思路被打破

机器之心 2023-08-13 00:38 Posted on 北京



Scan to Follow

机器之心编译

编辑：杜伟、梓文

本文中，约翰霍普金斯大学应用数学与统计学助理教授 Benjamin Grimmer 提出了理解梯度下降算法的全新思路。

在机器学习的世界中，最优化问题非常重要，它们能使世界变得更好。最优化问题旨在寻求完成某件事情的最佳方式，比如手机 GPS 计算达到目的地的最短路线，旅游网站搜索与行程相匹配的最便宜的航班。同时，机器学习应用通过分析数据模式进行学习，并试图为任何给定的最优化问题提供最准确和最人性化的答案。

对于简单的最优化问题，找到最佳解决方案只是一个算术问题。1847 年，法国数学家奥古斯丁·路易·柯西 (Augustin-Louis Cauchy) 研究了一个相当复杂的例子——天文计算。在那时他开创了一种常见的优化方法，也就是现在的梯度下降，它是优化方法中最经典和最简单的一阶方法之一。

如今，得益于其较低复杂度和简单操作，大多数机器学习程序都极其依赖梯度下降方法，其他领域也用它分析数据和解决工程问题。一百多年来，数学家们一直在完善梯度下降方法。然而上个月的一篇论文表明，关于梯度下降方法的基本假设可能是错误的。

这篇论文为《Provably Faster Gradient Descent via Long Steps》，唯一作者为约翰霍普金斯大学应用数学与统计学助理教授 Benjamin Grimmer。他对于自己的发现感到非常惊讶，就像直觉被打破一样。

他的反直觉结果表明，如果长期以来被认可的、找到给定问题最佳答案的规则被打破，则梯度下降的速度可以实现近 3 倍提升。再具体一点：他认为梯度下降算法可以通过包含意想不到的大步长 (large step size) 来更快地工作，这与研究人员长期以来所认为的相反。

Provably Faster Gradient Descent via Long Steps

Benjamin Grimmer*

Abstract

This work establishes provably faster convergence rates for gradient descent in smooth convex optimization via a computer-assisted analysis technique. Our theory allows nonconstant stepsize policies with frequent long steps potentially violating descent by analyzing the overall effect of many iterations at once rather than the typical one-iteration inductions used in most first-order method analyses. We show that long steps, which may increase the objective value in the short term, lead to provably faster convergence in the long term. A conjecture towards proving a faster $O(1/T \log T)$ rate for gradient descent is also motivated along with simple numerical validation.

论文地址：<https://arxiv.org/pdf/2307.06324.pdf>

虽然这一理论上的进展可能不适用于机器学习解决更棘手的问题，但可以促使研究人员重新考虑对梯度下降的理解。

MIT 的一名优化研究员 Shuvomoy Das Gupta 对此表示，「事实证明，我们并没有完全理解梯度下降背后的理论。现在，这项研究让我们更接近理解梯度下降的作用了。」



我们接下来看一看这项工作的具体内容。

研究概览

本文通过一种计算机辅助分析技术，在平滑凸优化中建立了可以证明更快的梯度下降收敛速度。其中，作者分析了一次多次迭代的整体效果而非大多数一阶方法分析中使用的典型单次迭代归纳，从而允许非恒定步长策略。

结果表明，更大的步长在短期内增加了目标值，但长期内实现了可证明的、更快的收敛。此外通过简单的数值验证，作者还提出了证明更快 $O(1/T \log T)$ 梯度下降率的一个猜想。

具体地讲，作者的证明基于性能估计问题（PEP）思路，它将计算或限制给定算法的最坏情况问题实例作为半定规划（Semidefinite Program, SDP）来处理。通过相关 SDP 可行解的存在，作者证明了应用非恒定步长模式后的下降保证，从而获得更快收敛保证。

在具体操作中，设计可证明的更快非恒定步长梯度下降方法相当于寻找具有很大平均步长值的直接（straightforward）步长模式。证明给定的模式很简单，可以利用半定规划来完成，参见定理3.1。

Theorem 3.1. A stepsize pattern $h \in \mathbb{R}^t$ is $\epsilon \geq 0$ -straightforward if for some $\Delta \in (0, 1/2]$, $\mathcal{S}_{h,\epsilon,\Delta}$ is nonempty where

$$\mathcal{S}_{h,\epsilon,\Delta} = \left\{ (\lambda, \gamma) \in \mathbb{R}^{(t+2) \times (t+2)} \times \mathbb{R}^{(t+2) \times (t+2)} \mid \begin{array}{l} \sum_{i,j \in I_t^*; i \neq j} \lambda_{i,j} a_{i,j} = a_{*,t} - a_{*,0} \\ \sum_{i,j \in I_t^*; i \neq j} \gamma_{i,j} a_{i,j} = 2 \sum_{i=0}^{t-1} h_i a_{i,*0} \\ m_h(\lambda) = 0 \\ \lambda \geq 0, \lambda + \Delta \gamma \geq 0 \\ \begin{bmatrix} \sum_{i=0}^{t-1} (h_i + \epsilon) & m_h(\gamma)^T \\ m_h(\gamma) & M_h(\lambda) \end{bmatrix} \succeq 0 \\ \begin{bmatrix} \sum_{i=0}^{t-1} (h_i + \epsilon) & m_h(\gamma)^T \\ m_h(\gamma) & M_h(\lambda + \Delta \gamma) \end{bmatrix} \succeq 0 \end{array} \right\}.$$

Proof. Let $(\lambda, \gamma) \in S_{h,e,\Delta}$. We prove this by showing $\lambda^{(\delta)} := \lambda + \delta\gamma \in \mathcal{R}_{h,e,\delta}$ for every $\delta \in [0, \Delta]$ by Lemma 3.1. This amounts to verifying the three conditions defining $\mathcal{R}_{h,e,\delta}$ for each $\lambda^{(\delta)}$.

下表 1 展示了越来越快的收敛保证的直接步长模式，其中每个模式都使用计算机生成的、精确算术半定规划解决方案进行了验证。未来的工作将确定更大步长的直接模式和其他可处理的非恒定、周期性大步长策略。

Pattern Length	"Straightforward" Stepsize Pattern h (longest stepsize marked in bold)	Convergence Rate $(+O(1/T^2))$ omitted
$t = 2$	(3 - η , 1.5) for any $\eta \in (0, 3)$	LD^2 $\frac{(2.25 - \eta/2) \times T}{LD^2}$
$t = 3$	(1.5, 4.9 , 1.5)	$2.63333... \times T$ LD^2
$t = 7$	(1.5, 2.2, 1.5, 12.0 , 1.5, 2.2, 1.5)	$3.1999999 \times T$ LD^2
$t = 15$	(1.4, 2.0, 1.4, 4.5, 1.4, 2.0, 1.4, 29.7 , 1.4, 2.0, 1.4, 4.5, 1.4, 2.0, 1.4)	$3.8599999 \times T$ LD^2
$t = 31$	(1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 8.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 72.3 , 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 8.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4)	$4.6032258 \times T$ LD^2
$t = 63$	(1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 7.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 12.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 7.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 164.0 , 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 7.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 14.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 7.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 12.6)	$5.2253968 \times T$ LD^2
$t = 127$	(1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 7.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 12.6, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 7.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 23.5, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 7.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 12.6, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 7.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 370.0 , 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 7.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 12.6, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 7.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 23.5, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 7.5, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 12.6, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 7.2, 1.4, 2.0, 1.4, 3.9, 1.4, 2.0, 1.4, 12.6)	$5.8346303 \times T$ LD^2

Table 1: Improved convergence rates for Gradient Descent with stepsizes cycling through a “straight-forward” pattern. Each convergence rate is proven by producing a certificate of feasibility for a related SDP, which is sufficient by our Theorems 2.1 and 3.1. Coefficients for $t \geq 7$ rates are slightly smaller than the ideal $\text{avg}(h)$ due to rounding to produce an exact arithmetic certificate.

但是，寻找长的、直接步长模式 h 很困难，所有直接模式的集合都是非凸的，导致局部搜索常常没有结果。如表 1 所示，长度 $t = 2^m - 1$ 的模式是通过重复 $t = 2^{m-1} - 1$ 两次而创建的，中间添加了一个新的步长，并手动缩短长度 $2^m - 1 - 1$ 子模式中的步长。作者表示，这种递归模式与以往研究中的二次极小化的循环和分形切比雪夫模式具有强相似性，还没有证明它们之间的联系。

作者表示，其方法与宾夕法尼亚大学优化研究员 Jason Altschuler 首次提出的方法非常相似，后者建立了长度为 2 或 3 的重复步长模式，并向最小化器更快收缩，实现平滑、强凸的最小化。

更细节的内容请参阅原论文。

从小步长到大步长，突破长度限制

我们知道，尽管没人能证明步长越小越好，但几十年来该领域的传统观点一直是采用小步长。这意味着在梯度下降方程中，步长不大于 2。

随着计算机辅助技术的进步，优化理论家已经开始测试更极限的技术。比如最近发表在《数学编程》期刊上的一项工作，Das Gupta 和其他研究者要求计算机为仅限 50 步的算法找到最佳步长，这是一种元优化问题。他们发现，最佳 50 步的长度变化很大，序列中一个步骤的长度几乎达到了 37，远高于长度 2 的典型上限。

[Home](#) > [Mathematical Programming](#) > Article

Full Length Paper | Published: 07 June 2023

Branch-and-bound performance estimation programming: a unified methodology for constructing optimal optimization methods

[Shuvomoy Das Gupta, Bart P. G. Van Parys & Ernest K. Ryu](#)

[Mathematical Programming](#) (2023) | [Cite this article](#)

324 Accesses | 9 Altmetric | [Metrics](#)

论文地址：<https://link.springer.com/article/10.1007/s10107-023-01973-1>

这一结果表明，优化研究人员遗漏了一些东西。因此，出于好奇，Grimmer 将 Das Gupta 的数值结果转化为了更普遍的定理。为了突破 50 步的任意上限，他探索了可重复序列的最佳步长，每次重复都更接近最佳答案。Grimmer 让计算机进行了数百万次步长序列的排列，从而找到那些最快收敛到答案的序列。

Grimmer 发现，最快的序列总是有一个共同点，即中间的一步总是很大，其大小取决于重复序列中的步骤数。对于 3 步序列，大步的长度为 4.9；对于 15 步序列，算法建议步长为 29.7；对于测试中最长的 127 步序列，中间的最大步长为 370。最终的结果表明，序列达到最佳点的速度是连续小步长速度的近三倍。

理论虽新颖，但无法改变当前使用方式

法国帕莱索理工学院优化研究员 Aymeric Dieuleveut 表示，这种循环方法代表了一种不同的梯度下降思维方式。他说道，「直觉告诉我，我不应该一步一步地思考问题，而是应该连续思考多个步骤。我认为很多人都忽略了这一点。」

不过，虽然这些见解可能会改变研究人员对梯度下降的看法，但可能不会改变这项技术目前的使用方式。毕竟，Grimmer 的论文只关注光滑函数和凸函数，光滑函数没有尖锐弯曲，凸函数的形状像一个碗，底部只有一个最优值。这些函数在理论上是最基础的，但在实践中却并不那么重要。机器学习研究人员使用的优化程序通常要复杂得多。

蒙特利尔大学优化与机器学习研究员 Gauthier Gidel 表示，一些经过改进的技术可以使 Grimmer 的大步长方法更快，但这些技术需要付出额外的运行成本。因此人们一直希望常规梯度下降法能在步长的正确组合下胜出。遗憾的是，这项新研究的三倍提速还远远不够。

Gidel 提出自己的疑问，「虽然表明情况略有改善，但我想真正的问题是：我们真的能缩小这个差距吗？」



© THE END

- 来自工业制造业、金融、医疗和农业等传统行业的，正在关注大模型落地的技术研发人员及工程师
- 在自身研究和学习中需要使用大模型的科研人员、老师和学生
-

那么，不要再犹豫，加入本次活动，高效升级你的大模型技术！论坛即将满额，赶快扫描下图二维码锁定你的入场资格！



© THE END

转载请联系本公众号获得授权

投稿或寻求报道：content@jizixin.com

Modified on 2023-08-13

People who liked this content also liked

OpenAI官方的Prompt工程指南：你可以这么玩ChatGPT
机器之心



"用 ChatGPT 轻松完成一篇学术论文！"
le读博日记



Transformer不读《红楼梦》
硅星人Pro

