软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(11):4893−4905 [doi: 10.13328/j.cnki.jos.007383] [CSTR: 32375.14.jos.007383]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于归一化的自适应方差缩减方法*

姜 伟[1], 杨斯凡[1,2], 王一博[1,2], 张利军[1,2]

[1](计算机软件新技术全国重点实验室 (南京大学), 江苏 南京 210023)

[2](南京大学 人工智能学院, 江苏 南京 210023)

通信作者: 张利军, E-mail: zhanglj@lamda.nju.edu.cn

摘　要: 随机优化算法是机器学习中处理大规模数据和复杂模型的重要方法. 其中, 方差缩减方法 (如 STORM 算法) 因其在随机非凸优化问题中能够实现最优的 $O(T^{-1/3})$ 收敛速率而受到广泛关注. 然而, 传统的方差缩减方法通常需要依赖特定的问题参数 (如光滑系数、噪声方差和梯度上界) 来设置学习率和动量, 使得它们在实际应用中难以直接使用. 为了解决这一问题, 提出了一种基于归一化的自适应方差缩减方法, 该方法无需预先知道问题参数, 仍然能够实现最优的收敛速率. 与现有的自适应方差缩减方法相比, 所提方法具有以下显著优势: (1) 无需依赖额外假设, 如梯度有界、函数值有界或极大的初始批量大小; (2) 实现了最优的 $O(T^{-1/3})$ 收敛速率, 不包含额外的 $O(\log T)$ 项; (3) 证明过程简洁明了, 便于推广到其他随机优化问题. 最后, 通过数值实验将该方法与其他方法进行了对比, 验证了其优越性.

关键词: 随机优化; 非凸优化; 自适应算法; 方差缩减; 收敛性分析

中图法分类号: TP311

中文引用格式: 姜伟, 杨斯凡, 王一博, 张利军. 基于归一化的自适应方差缩减方法. 软件学报, 2025, 36(11): 4893–4905. http://www.jos.org.cn/1000-9825/7383.htm

英文引用格式: Jiang W, Yang SF, Wang YB, Zhang LJ. Normalized Adaptive Variance Reduction Method. Ruan Jian Xue Bao/Journal of Software, 2025, 36(11): 4893–4905 (in Chinese). http://www.jos.org.cn/1000-9825/7383.htm

Normalized Adaptive Variance Reduction Method

JIANG Wei[1], YANG Si-Fan[1,2], WANG Yi-Bo[1,2], ZHANG Li-Jun[1,2]

[1](State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023, China)

[2](School of Artificial Intelligence, Nanjing University, Nanjing 210023, China)

Abstract: Stochastic optimization algorithms are recognized as essential for addressing large-scale data and complex models in machine learning. Among these, variance reduction methods, such as the STORM algorithm, have gained attention for their ability to achieve optimal convergence rates of $O(T^{-1/3})$. However, traditional variance reduction methods typically depend on specific problem parameters (e.g., the smoothness constant, noise variance, and gradient upper bound) for setting the learning rate and momentum, limiting their practical applicability. To overcome this limitation, this study proposes an adaptive variance reduction method based on a normalization technique, which eliminates the need for prior knowledge of problem parameters while maintaining optimal convergence rates. Compared to existing adaptive variance reduction methods, the proposed approach offers several advantages: (1) no reliance on additional assumptions, such as bounded gradients, bounded function values, or excessively large initial batch sizes; (2) the achievement of the optimal convergence rate of $O(T^{-1/3})$ without extra term of $O(\log T)$; (3) a concise and straightforward proof, facilitating extensions to other stochastic optimization problems. The superiority of the proposed method is further validated through numerical experiments, demonstrating enhanced performance when compared to other approaches.

Key words: stochastic optimization; non-convex optimization; adaptive algorithm; variance reduction; convergence analysis

---

* 基金项目: 国家自然科学基金 (62122037)
收稿时间: 2024-09-05; 修改时间: 2024-11-13; 采用时间: 2024-12-17; jos 在线出版时间: 2025-04-18
CNKI 网络首发时间: 2025-04-21

如今, 随机优化已成为现代机器学习中不可或缺的重要工具, 因为它能够有效应对大规模数据和复杂机器学习模型的挑战[1]. 传统优化方法在处理高维海量数据时往往面临计算效率低下的问题, 而随机优化方法通过在每次迭代中随机选择部分数据进行计算, 大幅降低了计算成本, 加快了模型训练速度, 并在常见的非凸问题中取得了显著的优化效果.

在经典的随机优化问题中[2], 我们给定一个光滑的非凸函数 $f:\mathbb{R}^d\to\mathbb{R}$, 目标是找到一个解 $x\in\mathbb{R}^d$ 使得目标函数 $f(x)$ 的值尽可能小. 其数学形式表现为:

$$\min_{x\in\mathbb{R}^d} f(x).$$

需要注意的是, 我们只能基于部分样本来计算该函数的梯度, 即获得 $\nabla f(\cdot,\xi)$, 其中 ξ 代表某个样本或一个小批量样本, 使得 $\mathbb{E}[\nabla f(\cdot,\xi)]=\nabla f(\cdot)$. 随机优化问题在机器学习中广泛存在[3,4]. 例如, 在监督学习中, x 通常表示模型的参数 (如神经网络的权重), ξ 表示一个数据样本, $f(x,\xi)$ 表示该样本的损失函数值, f 表示模型的整体训练损失.

由于通常不假设函数 f 具有凸性, 因此在一般情况下, 找到函数 f 的全局最小值可能是 NP 难的. 为此, 这类问题通常被放宽为寻找函数 f 的一个驻点, 即满足 $\nabla f(x)=0$ 的点. 此外, 我们仅假设可以访问任意点处的随机梯度 (即一阶信息), 而不使用黑塞矩阵等高阶信息. 在这种情况下, 最常用的方法是随机梯度下降法 (stochastic gradient descent, SGD). SGD 算法通过以下递归公式生成一系列迭代点 $x_1,\dots,x_T$:

$$x_{t+1}=x_t-\eta_t\nabla f(x_t,\xi_t),$$

其中, ξ_t 是从数据分布中独立采样的样本, 而 η_t 表示第 t 轮迭代的学习率. 当学习率 η_t 选择得当时, SGD 算法能够保证最终的迭代点 x_T 满足 $\mathbb{E}[\|\nabla f(x_T)\|]\leqslant \mathrm{O}(T^{-1/4})$, 其中 T 表示迭代轮数[5].

近年来, 为了进一步提升随机非凸优化问题中算法的收敛速率, 一类基于方差缩减的方法[6,7]被提出, 将收敛速度从传统 SGD 算法的 $\mathrm{O}(T^{-1/4})$ 进一步提升至 $\mathrm{O}(T^{-1/3})$. 与 SGD 算法不同, 方差缩减方法在每一步并非直接使用随机梯度 $\nabla f(x_t,\xi_t)$ 进行更新, 而是采用一个方差缩减估计器来追踪函数的梯度, 并基于该估计器更新变量. 以 STORM (stochastic recursive momentum) 算法[8]为例, 在初始迭代时 ($t=0$), 算法首先令 $v_0=\nabla f(x_0,\xi_0)$. 在随后的迭代中 ($t\geqslant 1$), 算法构建梯度估计器 v_t:

$$v_t=(1-\beta_t)v_{t-1}+\beta_t\nabla f(x_t;\xi_t)+(1-\beta_t)(\nabla f(x_t;\xi_t)-\nabla f(x_{t-1};\xi_t)).$$

这种构建方法确保在合理设置学习率 η_t 和动量参数 β_t 的前提下, v_t 的估计误差 $\mathbb{E}[\|v_t-\nabla f(x_t)\|]$ 能够随时间逐步减小. 在获得梯度估计器后, STORM 方法采用类似于 SGD 算法的方式进行变量更新, 即 $x_{t+1}=x_t-\eta_t v_t$. 然而, 为了实现最优的 $\mathrm{O}(T^{-1/3})$ 收敛速率, STORM 方法必须谨慎设置学习率 η_t 和动量参数 β_t, 具体如下所示:

$$\eta_t=\frac{k}{\left(w+\sum_{i=1}^{t}\|\nabla f(x_t;\xi_t)\|^2\right)^{1/3}},\ \beta_t=c\eta_t^2,$$

其中, $k=\mathrm{O}(G^{2/3}L^{-1})$, $w=\mathrm{O}(G^2)$, $c=\mathrm{O}(L^2)$, 且 G 为函数 f 的梯度上界, L 为函数 f 的光滑系数. 换言之, 学习率依赖于一系列问题参数和历史随机梯度的累计和. 在这种情况下, 算法需要事先知道问题参数 G 和 L, 才能正确设置学习率与动量参数, 从而获得最优的收敛速率. 然而, 在实际应用中, 我们很难准确知道参数 G 和 L 的值, 这使得算法在实际场景中难以应用.

为了解决上述问题, 研究者们对自适应方差缩减算法进行了深入研究[9−11], 旨在无需预先知晓问题参数的情况下, 自适应地自动调整学习率和动量参数, 同时仍然能够获得相应的收敛速率保障. 然而, 当前方法仍然存在一些局限: (1) 这些方法通常依赖额外的假设, 例如梯度有界、函数有界、较大的初始批量大小; (2) 它们只能获得 $\mathrm{O}(T^{-1/3}\log T)$ 的收敛速率, 无法匹配最优的 $\mathrm{O}(T^{-1/3})$ 收敛保障; (3) 方法的证明过程极为复杂, 需要分两个阶段分别对多项因子进行缩放, 不利于推广到其他随机优化问题. 为此, 本文基于归一化思想, 提出了一种全新的自适应方差缩减算法, 通过简单直接的证明技术, 在无需引入额外假设的情况下, 该方法能够实现最优的 $\mathrm{O}(T^{-1/3})$ 收敛速率,

并在数值实验中证明了其优越性.

本文的主要贡献如下.

(1) 提出了一种新颖的基于归一化的自适应方差缩减算法, 该算法能够自适应地调整超参数, 且学习率和动量参数仅依赖于迭代次数, 方法简单且易于实现.

(2) 在无需引入额外假设 (如梯度有界、函数值有界、极大初始批量大小) 的情况下, 本文证明了所提出算法能够实现最优的 $\mathrm{O}(T^{-1/3})$ 收敛速率, 并且其结果能够与理论下界相匹配. 在表 1 中我们对比了本文所提出的 NAVAR (normalized adaptive variance reduction) 方法与现有方差缩减算法间的差异.

表 1 不同方差缩减算法的对比

方法	收敛速率	是否自适应	额外假设
STORM[8]	$\mathrm{O}\left(T^{-1/3}\log T\right)$	否	梯度有界, 函数值有界
STORM+[9]	$\mathrm{O}\left(T^{-1/3}\right)$	是	梯度有界, 函数值有界
META-STORM[10]	$\mathrm{O}\left(T^{-1/3}\log T\right)$	是	梯度有界
Ada-STORM[11]	$\mathrm{O}\left(T^{-1/3}\right)$	是	极大初始批量大小
NAVAR (本文)	$\mathrm{O}\left(T^{-1/3}\right)$	是	无

(3) 通过数值实验验证了所提算法的有效性. 实验结果表明, 该算法能够在自适应调整超参数的同时取得较快的收敛速率, 优于其他相关算法.

本文第 1 节介绍方差缩减相关方法及其自适应技术的研究现状. 第 2 节阐述了方差缩减问题中所采用的基本假设. 第 3 节则详细介绍了本文提出的基于归一化的自适应方差缩减方法. 第 4 节给出了所提出算法的理论分析和证明过程. 第 5 节通过对比实验验证了所提出方法的有效性. 第 6 节对全文进行了总结.

1 相关工作

本节将简要介绍方差缩减方法和自适应算法的相关研究进展.

1.1 方差缩减方法

方差缩减算法在随机优化问题中得到了广泛应用, 该方法旨在通过减少梯度估计的误差, 显著提升算法的收敛速度. 方差缩减的概念最早可以追溯到 Roux 等人[12]提出的 SAG (stochastic average gradient) 算法, 该算法通过累积先前的梯度值来实现方差缩减效应, 并在强凸有限和优化问题中实现了线性收敛. 为了解决需要存储历史梯度的限制, Johnson 等人[13]和 Zhang 等人[14]提出了 SVRG (stochastic variance reduced gradient) 方法, 该方法通过周期性地计算全批量梯度, 达到了与 SAG 算法相同的收敛速度. 随后, Nguyen 等人[15]提出了 SARAH (stochastic recursive gradient algorithm) 方法, 进一步提高了光滑凸函数的收敛速率.

在非凸优化领域, Fang 等人[6]提出 SPIDER (stochastic path-integrated differential estimator) 估计器, 将随机非凸环境下的收敛率从 SGD 算法的 $\mathrm{O}(T^{-1/4})$ 提升到 $\mathrm{O}(T^{-1/3})$, 并在有限和场景下将收敛率进一步提升至 $\mathrm{O}(n^{1/4}T^{-1/2})$, 其中, n 代表有限和问题中的函数数量. 接着, Wang 等人[7]提出的 SpiderBoost 和 Prox-SpiderBoost 算法对 SPIDER 方法进行了优化, 采用了更大的常数步长, 并将其应用于复合优化问题. 然而, 这些方法的共同缺点是依赖于大批量样本, 导致在实际应用中计算需求极高. 为解决这一问题, Cutkosky 等人[8]随后提出了基于动量的 STORM 方法, 该方法能够在不依赖大批量样本的情况下, 实现 $\mathrm{O}(T^{-1/3}\log T)$ 的收敛速率.

1.2 自适应算法

在随机非凸优化问题中, Ghadimi 等人[5]证明了通过合理设计学习率, SGD 算法能够实现 $\mathrm{O}(T^{-1/4})$ 的收敛速率. 不同于该研究中基于迭代次数设定的学习率, 后续研究提出了许多基于历史随机梯度动态调整学习率的方法. 例如, Duchi 等人[16]提出的 AdaGrad 算法, 通过累计历史梯度来设定学习率, 在处理稀疏数据时表现优异. 随后,

Tieleman 等人[17]提出的 RMSprop 算法和 Kingma 等人[2]提出的 Adam 算法, 也采用了动态调整学习率的策略, 并在各种机器学习问题中取得了显著的效果. Huang 等人[18]则进一步结合 STORM 的方差缩减技术, 对 Adam 算法进行了改进, 提出了 Super-Adam 算法, 并获得了 $\mathrm{O}(T^{-1/3}\log T)$ 的收敛速率. 然而, 这些方法仍然需要预先知道某些问题参数, 才能准确设置学习率和动量参数, 因此它们并非完全自适应的算法. 为此, 许多研究[19−23]致力于开发完全自适应的 SGD 方法, 以在不知晓具体问题参数的情况下, 仍能保持类似的收敛速率.

近年来, 自适应方差缩减方法在随机优化领域得到了广泛关注. 其中的一个重要进展是 Levy 等人[9]引入的 STORM$^+$ (stochastic recursive momentum +) 方法, 该方法是 STORM 算法的一个自适应版本, 能够实现最优的收敛速率. 然而, STORM$^+$方法依赖于梯度有界和函数值有界的假设. 为了解决这一限制, Liu 等人[10]提出了 META-STORM-SG 和 META-STORM 方法, 去除了对函数值有界的依赖, 并证明了相似的收敛速率. 然而, 这两种方法仍然需要梯度有界的假设, 并且其收敛速率中包含了额外的 $\mathrm{O}(\log T)$ 项, 未能达到最优收敛速率. 最近, Jiang 等人[11]提出了 Ada-STORM (adaptive stochastic recursive momentum) 方法, 进一步去除了梯度有界的假设, 并实现了 $\mathrm{O}(T^{-1/3})$ 的收敛速率. 但是, 该方法在首轮迭代中需要使用极大的批量样本, 且证明过程非常复杂, 需分两阶段对多个因子分别进行约束, 这限制了其在其他随机优化问题中的推广应用.

2 基本假设

本文研究随机非凸优化问题, 下面介绍该问题所需的基本假设.

假设 1. 平均光滑性. 函数 f 的梯度随机采样满足平均光滑性, 即:

$$\mathbb{E}\left[\|\nabla f(x;\xi)-\nabla f(y;\xi)\|^2\right]\leqslant L^2\|x-y\|^2.$$

假设 2. 误差有上界. 函数 f 的梯度随机采样是对真实梯度的无偏估计, 且误差具有上界, 即:

$$\mathbb{E}[\nabla f(x;\xi)]=\nabla f(x)\ \text{且}\ \mathbb{E}\left[\|\nabla f(x;\xi)-\nabla f(x)\|^2\right]\leqslant\sigma^2.$$

假设 3. 初始误差. $f_*=\inf\limits_x f(x)\geqslant-\infty$ 并且 $f(x_1)-f_*\leqslant\Delta_f$, 其中 x_1 为初始点.

假设 1 是方差缩减算法的核心假设, 也是实现 $\mathrm{O}(T^{-1/3})$ 收敛率的关键. 假设 2 确保了梯度采样是无偏的且其误差是有限的, 而假设 3 则刻画了初始点处的误差大小. 基于上述 3 个假设, 现有的方差缩减方法[6,8,15,24]能够获得最优的 $\mathrm{O}(T^{-1/3})$ 收敛速率, 这也是本文采用的全部假设. 此外, 为了确保假设 2 中随机梯度是无偏估计的性质, 通常需要采用放回采样的方式. 当前也有文献 [25] 对不放回采样进行了研究, 但其仅能在有限和问题中获得理论保障, 无法分析本文所考虑的随机优化问题.

值得注意的是, 现有的自适应方差缩减方法通常还需要额外假设梯度有界和函数值有限, 具体如下.

假设 4. 梯度有界. 函数 f 的梯度随机采样具有上界, 即 $\|\nabla f(x;\xi)\|\leqslant G$.

假设 5. 函数值有上界. 函数 f 的函数值具有上界, 即 $\max_{x,y\in\mathbb{R}^d}|f(x)-f(y)|\leqslant B$.

Levy 等人[9]提出的方法依赖于假设 4 和假设 5, Liu 等人[10]提出的算法则需要假设 4, 而 Jiang 等人[11]提出的方法需要假设在初始轮使用极大的批量大小. 此外, 原始的 STORM 方法也依赖于假设 4. 与此不同的是, 本文的方法不依赖假设 4 或假设 5.

3 基于归一化的自适应方差缩减算法

本节介绍我们提出的自适应方差缩减算法. 与先前的自适应算法类似, 我们基于方差缩减算法 STORM 进行设计. 具体而言, 其核心在于构建方差缩减的梯度估计器 v_t, 在第 1 轮迭代时 (即 $t=0$), 令 $v_0=\nabla f(x_0,\xi_0)$. 在随后的迭代过程中 (即 $t\geqslant 1$), 梯度估计器 v_t 构建为:

$$v_t=(1-\beta_t)v_{t-1}+\beta_t\nabla f(x_t;\xi_t)+(1-\beta_t)(\nabla f(x_t;\xi_t)-\nabla f(x_{t-1};\xi_t)),$$

其中, β_t 为动量参数. 在该表达式中, 前两项 $(1-\beta_t)v_{t-1}+\beta_t\nabla f(x_t;\xi_t)$ 是传统动量法的更新方式, 而额外引入的第 3 项 $(1-\beta_t)(\nabla f(x_t;\xi_t)-\nabla f(x_{t-1};\xi_t))$ 起到了误差校正的作用, 是利用方差缩减提升收敛速率的关键. 与以往基于 STORM

的其他方法不同, 在更新梯度时, 我们的方法并非直接采用类似于 SGD 的更新方式 $x_{t+1}=x_t-\eta_t v_t$, 而是结合了归一化的思想, 进行如下的变量更新:

$$x_{t+1}=x_t-\eta_t\frac{v_t}{\|v_t\|},$$

其中, η_t 为学习率. 当 $\|v_t\|=0$ 时, 我们定义 $v_t/\|v_t\|=0$, 即此时参数 x_t 不会进行更新. 此外, 不同于先前方法需要将学习率及动量参数依赖于历史随机梯度的累积, 我们的方法无需对超参数 β_t 与 η_t 进行复杂设计, 只需设置 $\eta_t=\beta_t=T^{-2/3}$ 即可获得最优的理论保障. 我们将所提出的算法命名为 NAVAR (normalized adaptive variance reduction) 方法, 其伪代码如算法 1 所示.

算法 1. NAVAR 算法.

输入: 回合数 T;
输出: 最终结果 x_τ.

1. **for** $t=0$ **to** $T-1$ **do**
2. 设置 $\eta_t=\beta_t=\gamma=T^{-2/3}$;
3. **if** $t=0$ **then**
4. 计算 $v_0=\nabla f(x_0,\xi_0)$;
5. **else**
6. 计算 $v_t=(1-\beta_t)v_{t-1}+\beta_t\nabla f(x_t;\xi_t)+(1-\beta_t)(\nabla f(x_t;\xi_t)-\nabla f(x_{t-1};\xi_t))$;
7. 更新 $x_{t+1}=x_t-\eta_t\frac{v_t}{\|v_t\|}$;
8. **end if**
9. **end for**
10. 从集合 $\{1,2,\ldots,T\}$ 中采样 τ;
11. **return** x_τ;

在第 4 节中, 我们将证明所提出的算法 1 能够实现最优的 $\mathrm{O}(T^{-1/3})$ 收敛速率. 然而, 虽然算法 1 不再依赖问题参数来设置学习率与动量参数, 但这些超参数的设置仍然需要预先知道总的迭代次数 T. 为此, 我们可以通过构建一个分阶段算法来避免此限制. 具体而言, 我们假设算法具有 K 个阶段, 对于第 k 个阶段, 我们设置该阶段的迭代次数为 2^{k-1}, 并在完成迭代后进入下一个阶段, 同时将迭代次数翻倍并重置优化变量.

在超参数设置方面, 对于每个时间步 t, 我们首先确认其所在阶段为第 $1+\lfloor\log t\rfloor$ 个阶段, 从而确定该阶段的迭代次数为 $T_t=2^{\lfloor\log t\rfloor}$. 最后, 根据该迭代次数, 我们能够确定此时的学习率和动量参数为 $\eta_t=\beta_t=T_t^{-2/3}$. 通过这样的方式, 我们无需事先确定迭代次数 T 的大小. 同时, 学习率和动量参数不再在整个优化过程中维持不变, 而是会随着阶段数的增加而逐步下降. 在第 4 节中, 我们将给出该分阶段算法的理论保障.

4 理论分析

接下来, 我们对算法 1 进行理论分析. 方差缩减算法的核心在于确保梯度估计器 v_t 的估计误差随时间逐步降低, 这一性质可以通过引理 1 来描述.

引理 1. 算法 1 中的梯度估计器 v_t 满足如下性质: 当 $t\geqslant 1$ 时, 有:

$$\mathbb{E}[\|v_t-\nabla f(x_t)\|]\leqslant(1-\gamma)^t\sigma+(\sigma+L)\sqrt{\gamma}.$$

证明: 首先, 根据梯度估计器 v_t 的定义以及 $\beta_t=\gamma$, 当 $t\geqslant 1$ 时, 我们有:

$$\begin{aligned}v_t-\nabla f(x_t)&=(1-\gamma)v_{t-1}+\gamma\nabla f(x_t;\xi_t)+(1-\gamma)(\nabla f(x_t;\xi_t)-\nabla f(x_{t-1};\xi_t))-\nabla f(x_t)\\&=(1-\gamma)(v_{t-1}-\nabla f(x_{t-1}))+\gamma(\nabla f(x_t;\xi_t)-\nabla f(x_t))+(1-\gamma)(\nabla f(x_t;\xi_t)-\nabla f(x_{t-1};\xi_t)-\nabla f(x_t)+\nabla f(x_{t-1})).\end{aligned}$$

将上述公式进行进一步迭代, 我们可以得到:

$$v_t-\nabla f(x_t)=(1-\gamma)^t(v_0-\nabla f(x_0))+\gamma\sum_{i=1}^{t}(1-\gamma)^{t-i}(\nabla f(x_i;\xi_i)-\nabla f(x_i))$$
$$+\sum_{i=1}^{t}(1-\gamma)^{t+1-i}(\nabla f(x_i;\xi_i)-\nabla f(x_{i-1};\xi_i)+\nabla f(x_{i-1})-\nabla f(x_i)).$$

对等式两边同时取 l_2 范数并取期望, 上式变为:

$$\mathbb{E}[\|v_t-\nabla f(x_t)\|]\leqslant(1-\gamma)^t\mathbb{E}[\|v_0-\nabla f(x_0)\|]+\gamma\mathbb{E}\left[\left\|\sum_{i=1}^{t}(1-\gamma)^{t-i}(\nabla f(x_i;\xi_i)-\nabla f(x_i))\right\|\right]$$
$$+\mathbb{E}\left[\left\|\sum_{i=1}^{t}(1-\gamma)^{t+1-i}(\nabla f(x_i;\xi_i)-\nabla f(x_{i-1};\xi_i)+\nabla f(x_{i-1})-\nabla f(x_i))\right\|\right].$$

因为对于任意随机变量 X, 满足不等式 $(\mathbb{E}X)^2\leqslant\mathbb{E}[X^2]$, 即 $\mathbb{E}X\leqslant\sqrt{\mathbb{E}[X^2]}$, 所以上式可写作公式 (1):

$$\mathbb{E}[\|v_t-\nabla f(x_t)\|]\leqslant(1-\gamma)^t\mathbb{E}[\|v_0-\nabla f(x_0)\|]+\gamma\sqrt{\mathbb{E}\left[\left\|\sum_{i=1}^{t}(1-\gamma)^{t-i}(\nabla f(x_i;\xi_i)-\nabla f(x_i))\right\|^2\right]}$$
$$+\sqrt{\mathbb{E}\left[\left\|\sum_{i=1}^{t}(1-\gamma)^{t+1-i}(\nabla f(x_i;\xi_i)-\nabla f(x_{i-1};\xi_i)+\nabla f(x_{i-1})-\nabla f(x_i))\right\|^2\right]}\tag{1}$$

由于 $\mathbb{E}\left[\sum_{i=1}^{t}(1-\gamma)^{t-i}(\nabla f(x_i;\xi_i)-\nabla f(x_i))\right]=0$, 我们可得公式 (2):

$$\mathbb{E}\left[\left\|\sum_{i=1}^{t}(1-\gamma)^{t-i}(\nabla f(x_i;\xi_i)-\nabla f(x_i))\right\|^2\right]=\mathbb{E}\left[\sum_{i=1}^{t}(1-\gamma)^{2(t-i)}(\nabla f(x_i;\xi_i)-\nabla f(x_i))^2\right]$$
$$\leqslant\sigma^2\sum_{i=1}^{t}(1-\gamma)^{2(t-i)}\leqslant\frac{\sigma^2}{1-(1-\gamma)^2}\leqslant\frac{\sigma^2}{\gamma}\tag{2}$$

同理, 因为 $\mathbb{E}\left[\sum_{i=1}^{t}(1-\gamma)^{t+1-i}(\nabla f(x_i;\xi_i)-\nabla f(x_{i-1};\xi_i)+\nabla f(x_{i-1})-\nabla f(x_i))\right]=0$, 我们有公式 (3):

$$\mathbb{E}\left[\left\|\sum_{i=1}^{t}(1-\gamma)^{t+1-i}(\nabla f(x_i;\xi_i)-\nabla f(x_{i-1};\xi_i)+\nabla f(x_{i-1})-\nabla f(x_i))\right\|^2\right]$$
$$=\mathbb{E}\left[\sum_{i=1}^{t}(1-\gamma)^{2(t+1-i)}(\nabla f(x_i;\xi_i)-\nabla f(x_{i-1};\xi_i)+\nabla f(x_{i-1})-\nabla f(x_i))^2\right]$$
$$\leqslant\mathbb{E}\left[\sum_{i=1}^{t}(1-\gamma)^{2(t+1-i)}(\nabla f(x_i;\xi_i)-\nabla f(x_{i-1};\xi_i))^2\right]\leqslant\sum_{i=1}^{t}(1-\gamma)^{2(t+1-i)}L^2(x_i-x_{i-1})^2$$
$$=L^2\gamma^2\sum_{i=1}^{t}(1-\gamma)^{2(t+1-i)}\leqslant L^2\gamma^2\frac{1}{1-(1-\gamma)^2}\leqslant L^2\gamma^2\frac{1}{\gamma}=L^2\gamma\tag{3}$$

其中, 公式 (1) 成立是因为:

$$\mathbb{E}\left[\sum_{i=1}^{t}(1-\gamma)^{2(t+1-i)}(\nabla f(x_i;\xi_i)-\nabla f(x_{i-1};\xi_i)+\nabla f(x_{i-1})-\nabla f(x_i))^2\right]$$
$$\leqslant\mathbb{E}\left[\sum_{i=1}^{t}(1-\gamma)^{2(t+1-i)}((\nabla f(x_i;\xi_i)-\nabla f(x_{i-1};\xi_i))^2+2\langle\nabla f(x_i;\xi_i)-\nabla f(x_{i-1};\xi_i),\nabla f(x_{i-1})-\nabla f(x_i)\rangle+(\nabla f(x_{i-1})-\nabla f(x_i))^2)\right]$$
$$\leqslant\mathbb{E}\left[\sum_{i=1}^{t}(1-\gamma)^{2(t+1-i)}((\nabla f(x_i;\xi_i)-\nabla f(x_{i-1};\xi_i))^2-(\nabla f(x_{i-1})-\nabla f(x_i))^2)\right]$$
$$\leqslant\mathbb{E}\left[\sum_{i=1}^{t}(1-\gamma)^{2(t+1-i)}(\nabla f(x_i;\xi_i)-\nabla f(x_{i-1};\xi_i))^2\right].$$

将公式 (2) 和公式 (3) 代入公式 (1) 中, 最终得出结果:

$$\mathbb{E}[\|v_t-\nabla f(x_t)\|]\leqslant(1-\gamma)^t\mathbb{E}[\|v_0-\nabla f(x_0)\|]+\gamma\sqrt{\frac{\sigma^2}{\gamma}}+\sqrt{L^2\gamma}\leqslant(1-\gamma)^t\sigma+(\sigma+L)\sqrt{\gamma}.$$

证毕.

根据引理 1, 我们可以得知, 梯度估计器的估计误差会随时间逐渐减少, 表现出方差缩减的特性. 接着, 根据函数 f 的光滑性以及变量 x_t 的更新方式, 我们可以得出以下引理.

引理 2. 对于每一步迭代结果 x_t, 其满足:

$$\|\nabla f(x_t)\| \leqslant \frac{1}{\gamma}(f(x_t)-f(x_{t+1}))+2\|\nabla f(x_t)-v_t\|+\frac{\gamma L}{2}.$$

证明: 首先, 因为函数 f 是 L-光滑的 (根据假设 1), 我们有:

$$f(x_{t+1}) \leqslant f(x_t)+\langle \nabla f(x_t), x_{t+1}-x_t\rangle+\frac{L}{2}\|x_{t+1}-x_t\|^2.$$

由于算法的迭代方式为 $x_{t+1}=x_t-\eta_t\dfrac{v_t}{\|v_t\|}$, 且学习率 $\eta_t=\gamma$, 代入后上式可进一步简化为:

$$f(x_{t+1}) \leqslant f(x_t)+\left\langle \nabla f(x_t), -\gamma\frac{v_t}{\|v_t\|}\right\rangle+\frac{L}{2}\gamma^2.$$

通过在不等式右侧拆分出 $\gamma\left\langle v_t, -\dfrac{v_t}{\|v_t\|}\right\rangle$ 项, 上式可推出:

$$f(x_{t+1}) \leqslant f(x_t)+\left\langle \nabla f(x_t)-v_t, -\gamma\frac{v_t}{\|v_t\|}\right\rangle+\gamma\left\langle v_t, -\frac{v_t}{\|v_t\|}\right\rangle+\frac{L}{2}\gamma^2 \leqslant f(x_t)+\gamma\|\nabla f(x_t)-v_t\|-\gamma\|v_t\|+\frac{L}{2}\gamma^2.$$

通过移项并将不等式两边同时除以正的常数 γ, 我们得出:

$$\|v_t\| \leqslant \frac{1}{\gamma}(f(x_t)-f(x_{t+1}))+\|\nabla f(x_t)-v_t\|+\frac{\gamma L}{2}.$$

最后, 由于 $\|\nabla f(x_t)\| \leqslant \|\nabla f(x_t)-v_t\|+\|v_t\|$, 我们能够完成最终的证明:

$$\|\nabla f(x_t)\| \leqslant \|\nabla f(x_t)-v_t\|+\|v_t\| \leqslant \frac{1}{\gamma}(f(x_t)-f(x_{t+1}))+2\|\nabla f(x_t)-v_t\|+\frac{\gamma L}{2}.$$

证毕.

根据引理 1 和引理 2, 我们可以得出以下定理.

定理 1. 对于算法 1 的最终结果 x_τ, 其满足:

$$\mathbb{E}[\|\nabla f(x_\tau)\|] \leqslant \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla f(x_t)\|\right] \leqslant \frac{\Delta_f+4\sigma+3L}{T^{1/3}}.$$

证明: 首先, 根据引理 2, 求和得到:

$$\mathbb{E}\left[\sum_{t=1}^{T}\|\nabla f(x_t)\|\right] \leqslant \mathbb{E}\left[\frac{1}{\gamma}(f(x_1)-f(x_{T+1}))+2\sum_{t=1}^{T}\|\nabla f(x_t)-v_t\|+\frac{\gamma LT}{2}\right].$$

将上述不等式两边同时除以正数 T, 并根据引理 1 以及 $\gamma=T^{-2/3}$, 可以得到

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla f(x_t)\|\right] &\leqslant \mathbb{E}\left[\frac{1}{\gamma T}(f(x_1)-f(x_{T+1}))+\frac{2}{T}\sum_{t=1}^{T}\|\nabla f(x_t)-v_t\|+\frac{\gamma L}{2}\right] \\
&\leqslant \frac{1}{\gamma T}(f(x_1)-f_*)+\frac{2}{T}\sum_{t=1}^{T}(1-\gamma)^t\sigma+\frac{2}{T}\sum_{t=1}^{T}(\sigma+L)\sqrt{\gamma}+\frac{\gamma L}{2} \\
&\leqslant \frac{\Delta_f}{\gamma T}+\frac{2\sigma}{\gamma T}+2(\sigma+L)\sqrt{\gamma}+\frac{\gamma L}{2}=\frac{\Delta_f}{T^{1/3}}+\frac{2\sigma}{T^{1/3}}+\frac{2(\sigma+L)}{T^{1/3}}+\frac{L}{2T^{2/3}} \\
&= \frac{\Delta_f+4\sigma+2L}{T^{1/3}}+\frac{L}{2T^{2/3}} \leqslant \frac{\Delta_f+4\sigma+3L}{T^{1/3}}.
\end{aligned}$$

最后, 由于 τ 是从集合 $\{1,2,\ldots,T\}$ 中随机采样得到的, 我们可知:

$$\mathbb{E}[\|\nabla f(x_\tau)\|] \leqslant \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla f(x_t)\|\right] \leqslant \frac{\Delta_f+4\sigma+3L}{T^{1/3}}.$$

证毕.

根据定理 1, 我们可以得知, 本文提出的算法 1 具有 $O(T^{-1/3})$ 的最优收敛速率. 与先前的方法相比, 算法 1 不依赖于梯度有界和函数值有界的假设 (即假设 4 和假设 5), 也不需要在初始轮次中使用极大的批量大小. 所获得的收敛速率不包含额外的 $O(\log T)$ 项, 能够匹配随机非凸优化的最优理论下界.

虽然算法 1 能够获得最优的理论保障, 但其仍然需要事先知道迭代次数 T. 为了解决这一问题, 我们在第 3 节中引入了分阶段算法. 接下来我们给出该算法的收敛保障.

定理 2. 对于分阶段算法的输出结果 $\bar{x}$, 若最终的总迭代轮数为 T, 则其满足:

$$\mathbb{E}[\|\nabla f(\bar{x})\|] \leqslant \frac{4^{1/3}(\Delta_f + 4\sigma + 3L)}{T^{1/3}}.$$

证明: 设算法的最后一个完整阶段为第 S 个阶段. 因为前 K 个阶段的总迭代次数满足不等式:

$$2^0 + 2^1 + \ldots + 2^{K-1} < 2^K.$$

所以, 经过 T 次迭代后, 算法至少经历了 $\log T$ 个阶段. 因为第 k 个阶段的迭代次数为 2^{k-1}, 所以最后一个阶段的迭代次数为 $2^{\log T-1} = T/2$. 考虑到我们只采用完整阶段的输出结果, 即 $S = \lfloor \log T \rfloor$, 该阶段的迭代次数至少为 $T/4$. 根据定理 1 可知, 经过 T 次迭代后, 算法的输出结果满足:

$$\mathbb{E}[\|\nabla f(x_\tau)\|] \leqslant \frac{\Delta_f + 4\sigma + 3L}{T^{1/3}}.$$

于是, 当迭代次数为 $T/4$ 时, 我们可以确保

$$\mathbb{E}[\|\nabla f(\bar{x})\|] \leqslant \frac{\Delta_f + 4\sigma + 3L}{(T/4)^{1/3}} = \frac{4^{1/3}(\Delta_f + 4\sigma + 3L)}{T^{1/3}}.$$

证毕.

根据定理 2 可知, 在无需预先确定迭代次数 T 的情况下, 通过分阶段算法逐步加倍迭代次数的设计, 我们仍然能获得最优的 $O(T^{-1/3})$ 收敛速率. 与算法 1 中学习率与动量参数保持不变不同, 在这种情况下, 学习率和动量参数将随着阶段数的增加逐阶段下降.

5 实验分析

在本节中, 通过数值实验验证所提出方法的有效性. 我们在图像多分类任务和语言模型训练任务上进行实验, 并与相关的优化算法进行对比. 具体而言, 首先对比了常用的 SGD 算法[5]、Adam 算法[2]和 AdaBelief 算法[26]. 随后对比了方差缩减算法 STORM[8]及其自适应变体, 包括 STORM+[9]、META-STORM[10]和 Ada-STORM[11]. 在超参数设置方面, 首先参考了各方法原始论文中的推荐值, 然后进一步尝试对学习率进行搜索. 具体来说, 在集合 {1E–5, 1E–4, 1E–3, 1E–2, 1E–1}中进行了尝试, 并选择了表现最佳的结果进行汇报. 所有方法均在 PyTorch 框架[27]下实现, 实验在一台配备 8 卡 NVIDIA Tesla V100 GPU 的机器上完成. 为了确保结果的稳定性, 所有实验均运行多次, 并取平均值进行汇报.

5.1 图像多分类问题

首先, 我们在公开的图像多分类数据集 CIFAR-10 和 CIFAR-100[28]上进行了实验验证. 在 CIFAR-10 数据集上训练了 ResNet-18 网络[29]. 该数据集包含 10 个类别的 60 000 张 32×32 彩色图像, 其中 50 000 张用于训练, 10 000 张用于测试. 对于所有优化算法, 将批量大小设置为 256, 并训练了 200 个周期 (epoch). 在结果方面, 分别绘制了模型的训练损失 (training loss)、训练准确率 (training accuracy)、测试损失 (testing loss)、测试准确率 (testing accuracy), 如图 1 所示. 结果显示, 在训练损失和测试损失方面, 我们的 NAVAR 算法随着训练轮数的增加损失值迅速下降. 在准确率上, 我们的算法达到了更高的测试准确率, 证明了所提出方法的有效性.

接着, 在更为复杂的 CIFAR-100 数据集上训练 ResNet-34 网络[29]. 该数据集包含 100 个类别的 60 000 张 32×32 彩色图像, 其中 50 000 张用于训练, 10 000 张用于测试. 对于所有优化算法, 同样将批量大小设置为 256, 并

训练了 200 个周期 (epoch). 在图 2 中, 分别绘制了训练损失、训练准确率、测试损失和测试准确率. 结果显示, 在训练损失和测试损失方面, 我们的方法具有较快的下降速度, 在测试准确率上, 我们的 NAVAR 算法最终相较于其他算法取得了更高的测试准确率, 验证了所提出方法的有效性.

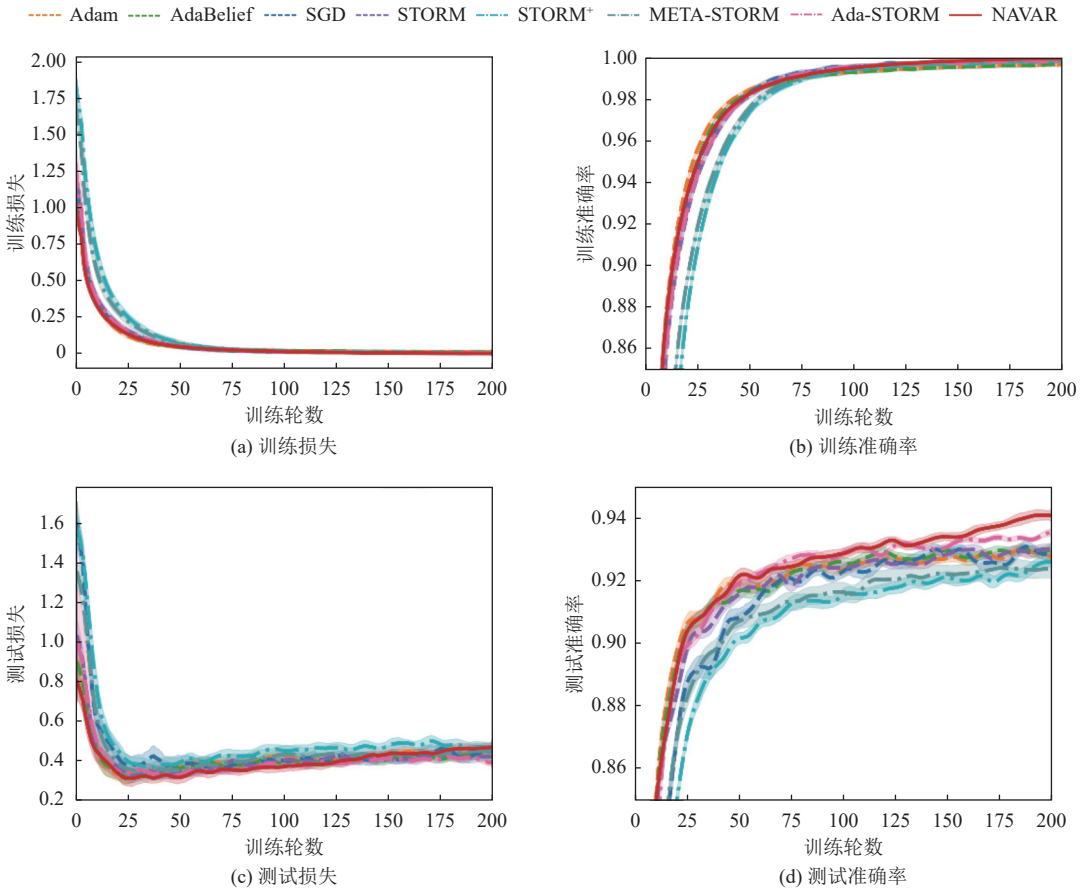


图 1　不同随机优化算法在 CIFAR-10 数据集上的表现

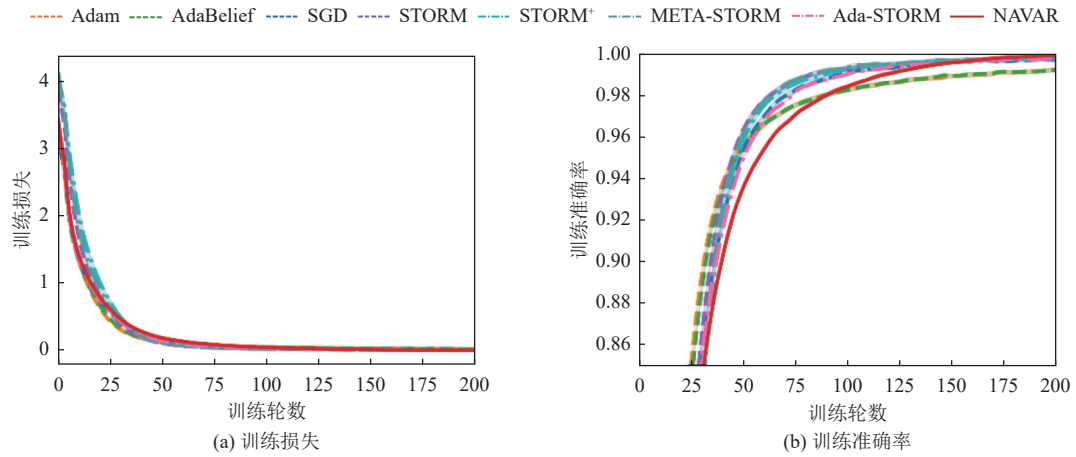
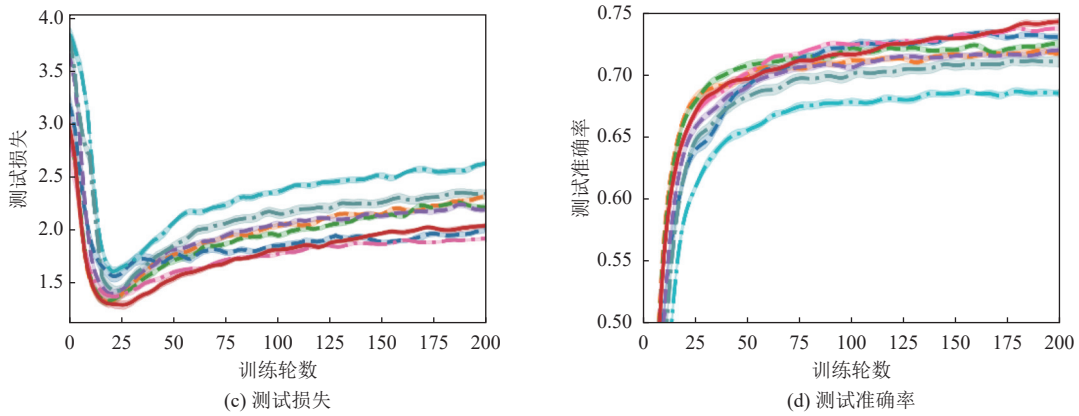


图 2　不同随机优化算法在 CIFAR-100 数据集上的表现



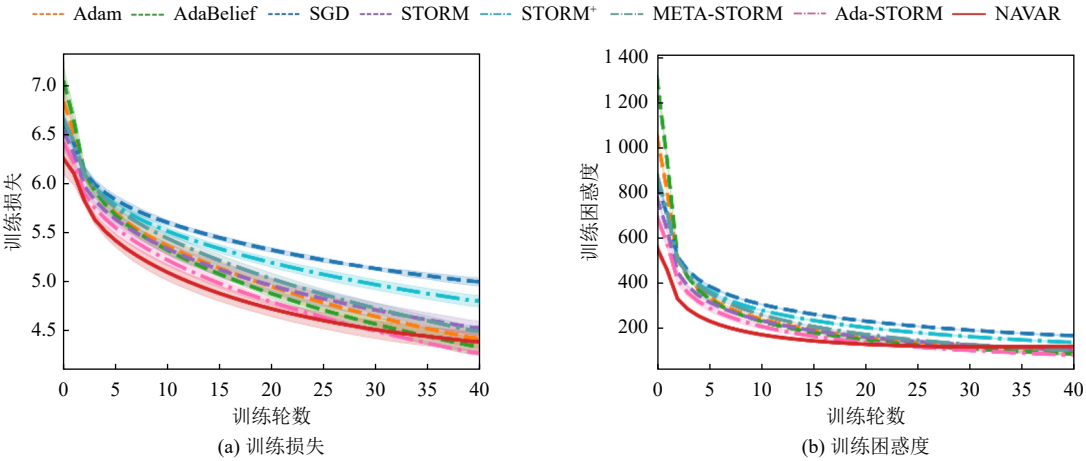
(c) 测试损失 (d) 测试准确率

图 2 不同随机优化算法在 CIFAR-100 数据集上的表现 (续)

5.2 语言模型训练问题

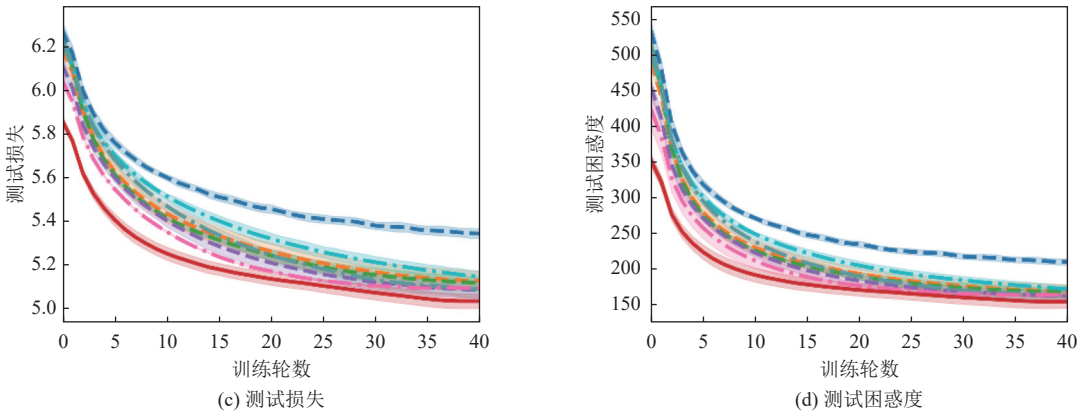
除了图像分类任务外, 我们还对语言模型的训练任务进行了实验. 为此, 在常用的公开文本数据集 WiKi-Text2[30] 上训练一个两层的 Transformer 网络[31]. WikiText2 是 WikiText-103 数据集的子集, 主要用于测试小型数据集的语言模型训练效果. 该数据集由约 10 万个句子组成, 包含约 200 万个词汇. 与其他数据集不同, WikiText2 保留了原始文本的丰富结构和标点符号, 适合用于研究语言模型的复杂性和生成质量. 而 Transformer 网络则是自然语言处理中的标准模型, 广泛应用于各种任务, 如机器翻译、文本生成、情感分析等. 在本次实验中, 我们使用 256 维的词嵌入, 设置了 512 个隐藏层和 2 个多头注意力机制. 在训练过程中, 所有优化算法的批量大小均设置为 20, 并进行了 40 个周期 (epoch) 的训练. 为了减缓过拟合, 训练中的丢失率 (dropout rate) 设置为 0.1. 图 3 展示了模型的训练损失 (training loss)、训练困惑度 (training perplexity)、测试损失 (testing loss) 和测试困惑度 (testing perplexity). 结果显示, 在训练损失和训练困惑度上, 我们的方法具有更快的下降速度. 而在测试损失和测试困惑度上, 我们的 NAVAR 算法在早期阶段相较于其他算法就展现出了较大优势, 且在最终阶段仍保持较低的测试损失和测试困惑度, 证明了所提出算法的优越性.

综上所述, 我们在图像多分类任务和自然语言模型训练任务中对所提出的算法进行了测试, 并与其他相关算法进行了对比. 在不同任务的 3 个数据集上, 所提出的方法在训练数据集上均很快完成了收敛, 并且在测试数据集上的表现优于其他相关算法, 证明了其具有较好的泛化性能. 总体而言, 所提出的自适应方差缩减算法在实际机器学习问题上展现了有效性和优越性.



(a) 训练损失 (b) 训练困惑度

图 3 不同随机优化算法在 WikiText2 数据集上的表现



(c) 测试损失 (d) 测试困惑度

图 3 不同随机优化算法在 WikiText2 数据集上的表现 (续)

6 总 结

方差缩减算法是随机优化方法中的一个重要分支, 通过设计具有方差缩减效应的梯度估计器, 可以实现最优的 $O(T^{-1/3})$ 收敛速率. 本文针对传统方差缩减方法需要事先知道诸多问题参数 (如光滑系数、噪声方差、梯度上界) 才能合理设置学习率和动量参数的局限性, 提出了一种能够自动调整这些超参数的自适应方差缩减算法. 首先, 我们通过归一化方法对现有的方差缩减算法进行了修改, 使其仅需将学习率和动量参数设置为与迭代次数相关的固定值, 并证明该算法具有最优的 $O(T^{-1/3})$ 收敛速率. 为了解决所提出算法需要事先知道迭代次数的限制, 我们进一步提出了分阶段算法, 通过逐步增加每个阶段内的迭代次数, 使学习率和动量参数逐阶段下降. 通过这种方式, 在无需预先确定迭代次数的情况下, 也能获得最优的收敛速率. 与现有的自适应方差缩减方法相比, 本文提出的方法无需额外假设, 如梯度有界、函数值有界, 且不需要在首轮使用较大的批量大小. 此外, 本文获得的收敛速率能够匹配理论下界, 不包含额外的对数项. 最后, 本文的证明方法简洁明了, 无需划分不同阶段分别约束不同项, 这有助于将该方法拓展至其他随机优化问题, 如双层优化[32]、多层优化[33]、内外耦合优化[34]、最小-最大优化[35]、联邦学习[36]等. 在数值实验方面, 我们在图像多分类任务和自然语言模型训练任务上进行了验证, 实验结果证明了所提出算法的优越性.

References:

[1] Zhou ZH, Wang W, Gao W, Zhang LJ. Introduction to the Theory of Machine Learning. Beijing: China Machine Press, 2020 (in Chinese).

[2] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980, 2017.

[3] Zhu XH, Tao Q, Shao YJ, Chu DJ. Stochastic optimization algorithm with variance reduction for solving non-smooth problems. Ruan Jian Xue Bao/Journal of Software, 2015, 26(11): 2752–2761 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4890.html [doi: 10.13328/j.cnki.jos.004890]

[4] Shao YJ, Tao Q, Jiang JY, Zhou B. Stochastic algorithm with optimal convergence rate for strongly convex optimization problems. Ruan Jian Xue Bao/Journal of Software, 2014, 25(9): 2160–2171 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4633.html [doi: 10.13328/j.cnki.jos.004633]

[5] Ghadimi S, Lan GH. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. SIAM Journal on Optimization, 2013, 23(4): 2341–2368. [doi: 10.1137/120880811]

[6] Fang C, Li CJ, Lin ZC, Zhang T. SPIDER: Near-optimal non-convex optimization via stochastic path integrated differential estimator. In: Proc. of the 32nd Int'l Conf. on Neural Information Processing Systems. Montréal: Curran Associates Inc., 2018. 687–697.

[7] Wang Z, Ji KY, Zhou Y, Liang YB, Tarokh V. SpiderBoost and momentum: Faster stochastic variance reduction algorithms. In: Proc. of the 33rd Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2019. 2406–2416.

[8] Cutkosky A, Orabona F. Momentum-based variance reduction in non-convex SGD. In: Proc. of the 33rd Int'l Conf. on Neural

Information Processing Systems. Vancouver: Curran Associates Inc., 2019. 15236–15245.

[9] Levy KY, Kavis A, Cevher V. STORM$^+$: Fully adaptive SGD with momentum for nonconvex optimization. In: Proc. of the 35th Int'l Conf. on Neural Information Processing Systems. Curran Associates Inc., 2021. 20571–20582.

[10] Liu ZJ, Nguyen TD, Nguyen TH, Ene A, Nguyen HL. META-STORM: Generalized fully-adaptive variance reduced SGD for unbounded functions. arXiv:2209.14853, 2022.

[11] Jiang W, Yang SF, Wang YB, Zhang LJ. Adaptive variance reduction for stochastic optimization under weaker assumptions. arXiv:2406.01959, 2024.

[12] Le Roux N, Schmidt M, Bach F. A stochastic gradient method with an exponential convergence rate for finite training sets. In: Proc. of the 26th Int'l Conf. on Neural Information Processing Systems. Lake Tahoe: Curran Associates Inc., 2012. 2663–2671.

[13] Johnson R, Zhang T. Accelerating stochastic gradient descent using predictive variance reduction. In: Proc. of the 27th Int'l Conf. on Neural Information Processing Systems. Lake Tahoe: Curran Associates Inc., 2013. 215–323.

[14] Zhang LJ, Mahdavi M, Jin R. Linear convergence with condition number independent access of full gradients. In: Proc. of the 27th Int'l Conf. on Neural Information Processing Systems. Lake Tahoe: Curran Associates Inc., 2013. 980–988.

[15] Nguyen LM, Liu J, Scheinberg K, Takáč M. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In: Proc. of the 34th Int'l Conf. on Machine Learning. Sydney: JMLR.org, 2017. 2613–2621.

[16] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 2011, 12: 2121–2159.

[17] Tieleman T, Hinton G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. 2012. https://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf

[18] Huang FH, Li JY, Huang H. SUPER-ADAM: Faster and universal framework of adaptive gradients. In: Proc. of the 35th Int'l Conf. on Neural Information Processing Systems. Curran Associates Inc., 2021. 9074–9085.

[19] Orabona F. Simultaneous model selection and optimization through parameter-free stochastic learning. In: Proc. of the 28th Int'l Conf. on Neural Information Processing Systems. Montreal: MIT Press, 2014. 1116–1124.

[20] Chen KY, Langford J, Orabona F. Better parameter-free stochastic optimization with ODE updates for coin-betting. In: Proc. of the 36th AAAI Conf. on Artificial Intelligence. Virtually: AAAI, 2022. 6239–6247. [doi: 10.1609/aaai.v36i6.20573]

[21] Carmon Y, Hinder O. Making SGD parameter-free. In: Proc. of the 35th Annual Conf. on Learning Theory. London, 2022. 2360–2389.

[22] Ivgi M, Hinder O, Carmon Y. Dog is SGD's best friend: A parameter-free dynamic step size schedule. In: Proc. of the 40th Int'l Conf. on Machine Learning. Honolulu: JMLR.org, 2023. 14465–14499.

[23] Yang JC, Li X, Fatkhullin I, He N. Two sides of one coin: The limits of untuned SGD and the power of adaptive methods. In: Proc. of the 37th Int'l Conf. on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2023. 74257–74288.

[24] Li ZZ, Bao HY, Zhang XL, Richtárik P. Page: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In: Proc. of the 38th Int'l Conf. on Machine Learning. PMLR, 2021. 6286–6295.

[25] Huang XM, Yuan K, Mao XH, Yin WT. Improved analysis and rates for variance reduction under without-replacement sampling orders. In: Proc. of the 35th Int'l Conf. on Neural Information Processing Systems. Curran Associates Inc., 2021. 3232–3243.

[26] Zhuang JT, Tang T, Ding YF, Tatikonda S, Dvornek N, Papademetris X, Duncan JS. AdaBelief optimizer: Adapting stepsizes by the belief in observed gradients. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 18795–18806.

[27] Paszke A, Gross S, Massa F, *et al*. PyTorch: An imperative style, high-performance deep learning library. In: Proc. of the 33rd Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2019. 8026–8037.

[28] Krizhevsky A. Learning multiple layers of features from tiny images [MS. Thesis]. Toronto: University of Toronto, 2009.

[29] He KM, Zhang XY, Ren SQ, Sun J. Deep residual learning for image recognition. In: Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778. [doi: 10.1109/CVPR.2016.90]

[30] Merity S. The WikiText long term dependency language modeling dataset. 2016. https://www.salesforce.com/ca/products/einstein/ai-research/the-wikitext-dependency-language-modeling-dataset

[31] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.

[32] Yang JJ, Ji KY, Liang YB. Provably faster algorithms for bilevel optimization. In: Proc. of the 35th Int'l Conf. on Neural Information Processing Systems. Curran Associates Inc., 2021. 13670–13682.

[33] Wang MD, Fang EX, Liu H. Stochastic compositional gradient descent: Algorithms for minimizing compositions of expected-value functions. Mathematical Programming, 2017, 161(1-2): 419–449. [doi: 10.1007/s10107-016-1017-3]

[34] Jiang W, Li G, Wang YB, Zhang LJ, Yang TB. Multi-block-single-probe variance reduced estimator for coupled compositional optimization. In: Proc. of the 36th Int'l Conf. on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2022. 32499–32511.

[35] Xian WH, Huang FH, Zhang YF, Huang H. A faster decentralized algorithm for nonconvex minimax problems. In: Proc. of the 35th Int'l Conf. on Neural Information Processing Systems. Curran Associates Inc., 2021. 25865–25877.

[36] Das R, Acharya A, Hashemi A, Sanghavi S, Dhillon IS, Topcu U. Faster non-convex federated learning via global and local momentum. In: Proc. of the 38th Conf. on Uncertainty in Artificial Intelligence. Eindhoven, 2022. 496–506.

附中文参考文献:

[1] 周志华, 王魏, 高尉, 张利军. 机器学习理论导引. 北京: 机械工业出版社, 2020.

[3] 朱小辉, 陶卿, 邵言剑, 储德军. 一种减小方差求解非光滑问题的随机优化算法. 软件学报, 2015, 26(11): 2752–2761. http://www.jos.org.cn/1000-9825/4890.html [doi: 10.13328/j.cnki.jos.004890]

[4] 邵言剑, 陶卿, 姜纪远, 周柏. 一种求解强凸优化问题的最优随机算法. 软件学报, 2014, 25(9): 2160–2171. http://www.jos.org.cn/1000-9825/4633.html [doi: 10.13328/j.cnki.jos.004633]

姜伟(1998－), 男, 博士生, 主要研究领域为机器学习与随机优化.

王一博(1999－), 男, 博士生, 主要研究领域为机器学习与优化.

杨斯凡(2001－), 男, 硕士生, 主要研究领域为机器学习与优化.

张利军(1986－), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为机器学习与优化.