LION 优化器的收敛速度分析

董一鸣[1)] 李 欢[2)] 林宙辰[1),3),4),5)]

[1)](北京大学智能学院 北京 100871)

[2)](南开大学人工智能学院 天津 300071)

[3)](北京大学跨媒体通用人工智能全国重点实验室 北京 100871)

[4)](北京大学人工智能研究院 北京 100871)

[5)](琶洲实验室(黄埔) 广州 510335)

摘 要 LION(evoLved sIgn mOmeNtum)是 Google 公司通过启发式程序搜索的方式发现的优化器,是一种独特的基于学习的优化算法。LION 算法通过在上步动量和本步梯度之间维持两个不同的插值,并有效结合了解耦的权重衰减技术,实现了超越传统符号梯度下降类算法的性能。LION 算法在许多大规模深度学习问题中展现了较强的优势,得到了广泛的应用。然而,尽管已有工作已经证明了 LION 的收敛性,但尚未有研究给出一个全面的收敛速度分析。已有研究证明,LION 能够解决一类特定的盒约束优化问题,本文着重证明了,在 ℓ_1 范数度量下,LION 收敛到这类问题的 Karush-Kuhn-Tucker (KKT)点的速度为 $\mathcal{O}(\sqrt{d}K^{-1/4})$,其中 d 为问题维度,K 为算法的迭代步数。更进一步,我们移除了约束条件,证明 LION 在一般无约束问题上以相同的速度收敛至目标函数的驻点。与已有研究工作相比,本文证明的收敛速度达到了关于问题维度 d 的最优依赖关系;关于迭代步数 K,这一速度还达到了非凸优化问题中随机梯度类算法能实现的最优理论下界。此外,这一理论下界以梯度的 ℓ_2 范数度量,而 LION 所属的符号梯度下降类算法通常度量的是更大的 ℓ_1 范数。由于在不同的梯度范数度量下关于问题维度 d 得到的收敛速度结果会有所差异,为了验证本文证明的收敛速度关于维度 d 同样是最优的,我们在多种深度学习任务上设计了全面的实验,不仅证明了 LION 与同样匹配理论下界的随机梯度下降法相比具有更低的训练损失和更强的性能,而且还验证了 LION 算法在迭代过程中梯度的 ℓ_1/ℓ_2 范数比始终处于 $\Theta(\sqrt{d})$ 的量级,从而在经验上说明了本文证明的收敛速度同样匹配关于 d 的最优下界。

关键词 机器学习;深度学习;非凸优化;收敛速度分析;LION 优化器

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2025.02008

Convergence Rate Analysis of LION

DONG Yi-Ming[1)] LI Huan[2)] LIN Zhou-Chen[1),3),4),5)]

[1)](School of Intelligence Science and Technology, Peking University, Beijing 100871)

[2)](College of Artificial Intelligence, Nankai University, Tianjin 300071)

[3)](State Key Lab of General AI, Peking University, Beijing 100871)

[4)](Institute for Artificial Intelligence, Peking University, Beijing 100871)

[5)](Pazhou Laboratory (Huangpu), Guangzhou 510335)

Abstract The LION (evoLved sIgn mOmeNtum) optimizer was found by Google via program search, making it a unique learning-based optimization algorithm. With the incorporation of two distinct interpolations between the previous step momentum and the current step gradient, as well as the integration of decoupled weight decay, LION successfully outperforms many traditional sign-based gradient descent algorithms, showing impressive performance in solving large

收稿日期:2025-01-23;在线发布日期:2025-05-15。本课题得到国家重点研发计划(2022ZD0160300)、国家自然科学基金面上项目(62276004,62476142)资助。**董一鸣**,博士研究生,中国计算机学会(CCF)学生会员,主要研究领域为深度学习、优化算法。E-mail:yimingdong_ml@outlook.com。**李 欢**,博士,副教授,中国计算机学会(CCF)会员,主要研究领域为机器学习、数值优化。**林宙辰(通信作者)**,博士,教授,中国计算机学会(CCF)杰出会员,主要研究领域为机器学习、数值优化。E-mail:zlin@pku.edu.cn。

scale deep learning problems. Although previous studies have investigated its convergence properties, no research has yet provided a comprehensive analysis of its convergence rate, which is more practically relevant. Recognizing that LION can be regarded as solving a specific box-constrained problem, this paper focuses on demonstrating its convergence to the Karush-Kuhn-Tucker (KKT) point at the rate of $\mathcal{O}(\sqrt{d}K^{-1/4})$, where d is the problem dimension and K is the number of iteration steps. Step further, we remove the constraint and establish that LION converges to the critical point of the general unconstrained problem at the same rate. This rate not only delivers the currently optimal dependence on the problem dimension d among existing studies but also tightly matches the theoretical lower bound for nonconvex stochastic optimization algorithms with respect to the number of iterations K. Additionally, the lower bound is typically measured by the gradient ℓ_2 norm, while the LION optimizer, as a member of the SignSGD family, usually measures the larger ℓ_1 norm. Since different gradient norm measures may lead to different convergence rate dependencies on d, to verify that our convergence rate is also optimal with respect to d, we conduct extensive experiments across various deep learning tasks. Through these experiments, we not only demonstrate that LION achieves lower loss and higher performance compared to standard SGD, but also empirically confirm that the gradient ℓ_1/ℓ_2 norm ratio aligns with $\Theta(\sqrt{d})$, thus proving that our convergence rate matches the theoretical lower bound with respect to d in the empirical sense.

Keywords　machine learning; deep learning; nonconvex optimization; convergence rate analysis; LION optimizer

1　引　言

优化问题在科学研究和工程实践中广泛存在。例如,在机器学习中,我们希望找到最优的模型参数,最小化训练过程中的误差损失[1-2];在物流管理中,需要规划最优的运输路线,以降低运行成本并提高工作效率[3-4];在金融投资中,需要设置最优的投资组合策略,在实现最大化收益的同时合理控制风险[5-7]。这些问题尽管来源不同,但本质上都可以抽象为数学上的优化问题,即寻找一组最优的决策变量,使目标函数达到最优值。

根据目标函数的性质,优化问题可以分为凸优化和非凸优化两大类。凸优化问题具有良好的数学性质,使得其求解过程相对稳定且高效[8-9];然而,现实世界中的绝大多数问题都是非凸优化问题[10-11]。以机器学习领域为例,深度神经网络训练、强化学习策略优化、计算机视觉中的图像分类和推荐系统中的矩阵分解等问题都属于非凸优化问题。

在非凸优化领域,寻找全局最优解通常是 NP 难问题[10,12]。另一方面,一个优化问题的求解效果极大地依赖于所选择的优化算法[13]。深度学习问题作为一类特定的非凸优化问题,其优化算法的选择不仅显著影响着问题解决的收敛速度,也决定了最终所得解的质量[14]。在这种背景下,研究高效快速的优化算法显得尤为重要。

学界通常使用迭代过程中梯度范数的上界来刻画算法的收敛速度。迄今为止,绝大多数工作致力于研究收敛速度与迭代步数 K 之间的关系,以寻求在更少步数内达到更高精度的优化算法。然而,随着深度学习任务规模的快速增长,在当今的大模型时代,需要处理的优化问题的维度 d 可能非常大。例如,在 GPT-3 模型的训练任务中,我们有 $d = 1.75 \times 10^{11}$ [15]。在这种背景下,收敛速度为 $\mathcal{O}(d)$ 和 $\mathcal{O}(\sqrt{d})$ 的优化算法可能具有若干个数量级的差距。因此,研究深度模型训练过程的收敛速度与模型大小 d 之间的关系具有很大的意义。

在以深度学习为代表的大规模复杂优化问题的处理中,随机梯度下降法(Stochastic Gradient Descent, SGD)及其各种变体一直占据着主导地位[16-20]。随机梯度类算法①的一个分支是自适应学习率算法,在这类算法的收敛速度分析中,大多数工

① 本文中提及一些相似的概念,在这里进行澄清:"随机梯度类算法"指满足第 3 节中假设 1-3 的所有算法;"随机梯度下降法(SGD)"为其中的一个算法;"符号梯度下降类算法"指随机梯度类算法中使用了符号函数的那类算法。

作采用梯度的 ℓ_2 范数作为度量;符号梯度下降类算法作为另一个重要分支,其中的工作几乎都采用 ℓ_1 范数作为度量。在深度学习研究的早期阶段,由于有限维空间中各范数的等价性,在梯度的不同范数度量下所得的收敛速度关于迭代步数 K 并无区别;然而,对梯度 $\nabla f(\boldsymbol{\theta})\in\mathbb{R}^d$,$\ell_1$ 与 ℓ_2 范数的等价不等式为

$$\|\nabla f(\boldsymbol{\theta})\|_2 \leqslant \|\nabla f(\boldsymbol{\theta})\|_1 \leqslant \sqrt{d}\,\|\nabla f(\boldsymbol{\theta})\|_2 \quad (1)$$

从式(1)中可以看出,当研究收敛速度关于问题维度 d 之间的关系时,采用梯度的不同范数度量可能会产生不同的结果。

尽管许多随机梯度类算法在多种深度学习任务上取得了良好的性能,然而,目前关于其理论性质的研究仍较为有限。Arjevani 等人指出,随机梯度类算法在解决非凸问题的过程中关于迭代步数 K 的收敛速度不会超过 $\mathcal{O}(K^{-1/4})$,此即为收敛速度的下界[21]。原始的 SGD 算法匹配了这一下界[22],但许多工作指出,该算法已无法适应当今大规模深度网络的训练过程[23-25]。在自适应学习率算法的研究中,除了 Li 等人关于 RMSProp 算法的分析外[15],绝大多数工作关于迭代步数 K 的收敛速度均无法匹配下界,且并未指明关于问题维度 d 的关系;但 RMSProp 算法在实际深度学习任务中已较少使用。在符号梯度下降类算法的研究中,尽管许多工作关于 K 匹配了理论下界,但这些工作要么没有针对 d 展开研究,要么关于 d 展示了较差的依赖。读者可参阅表 1 进行相关工作的详细对比。在此背景下,研究一个在大规模深度学习任务中经常使用且具有优秀理论性质的优化器成为了一个意义重大的关键问题。

作为符号梯度下降类算法的一员,LION (evoLved sIgn mOmeNtum)优化器是一个非常有意义的创新工作[26]。Chen 等人将优化器的发现问题等价为计算机程序的搜索问题,通过设计基于遗传算法的搜索策略得到了能普适于各种深度学习任务的 LION 算法。

LION 在许多实际应用场景中展现了出色的性能。然而,与传统的专家手工设计范式不同,LION 是通过启发式的搜索得到的,并不自然具备理论上的收敛性,因此关于其理论性质的深入分析变得非常关键。尤其是在要求较高可信度或计算资源有限的优化任务中,对其收敛速度的研究结果显得更为重要[27]。

本文填补了这一研究空白,通过全面的收敛速度分析,展示了 LION 优化器在解决一种特定的带约束优化问题和广义的无约束优化问题的理论表现。本文证明,在梯度的 ℓ_1 范数度量下,LION 优化器在解决一种特定的带约束优化问题和广义的无约束优化问题中均表现出 $\mathcal{O}(\sqrt{d}K^{-1/4})$ 的收敛速度,其中 K 为算法迭代步数。该速度不仅具有当前研究工作中对问题维度 d 的最佳依赖关系,而且在迭代步数 K 方面紧密贴合了非凸优化问题中随机梯度类算法所能达到的理论下界。我们在(1)式中已经指出,关于 d 的收敛速度结果与所选取的范数有关。这个理论下界是使用梯度的 ℓ_2 范数刻画的,为了说明与本文结果之间的关系,我们在多种大规模任务上完成了全面的实验,指明在深度网络的训练过程中梯度的 ℓ_1/ℓ_2 范数比始终处于 $\Theta(\sqrt{d})$ 的量级(而不是常数量级 $\Theta(1)$),从而可以约去收敛速度中的 $\sqrt{d}$ 项,进而从经验意义上说明在问题维度 d 方面本文推出的收敛速度也贴合了随机梯度类算法所能实现的理论下界。

综上所述,本文的研究深入而全面地分析了 LION 的收敛速度,不仅增强了非凸优化领域对该算法的理论理解,也说明了 LION 关于问题维度和迭代步数均能够高效地完成大规模优化任务,为在实际优化场景中的更可靠应用开辟了道路。

本文的剩余部分结构如下:第 2 节介绍相关背景,讨论了前人在深度学习优化领域的工作,着重介绍 LION 优化器的相关研究,并总结本文的贡献;第 3 节详细描述了本文的主要贡献,即 LION 优化器在一个特定的有约束和广义的无约束问题下的收敛速度分析;第 4 节详细给出了 LION 优化器收敛速度的证明过程;第 5 节介绍了具体的实验设定,分析了实验结果,并指出这些结果佐证了我们的理论发现;第 6 节是本文的简要总结。

2 相关背景

假设存在一个目标函数 f,输入参数为 $\boldsymbol{\theta}$。设数据集 S 包含来自随机变量 $\boldsymbol{\zeta}$ 的分布 $\{\boldsymbol{\zeta}^1,\boldsymbol{\zeta}^2,\cdots,\boldsymbol{\zeta}^s\}$。绝大多数深度学习问题都可以表示为最小化目标函数 f 的优化问题:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} f(\boldsymbol{\theta}) = \mathbb{E}_{\zeta\sim\mathcal{D}}[\hat{f}(\boldsymbol{\theta},\boldsymbol{\zeta})] \quad (2)$$

其中,d 表示问题的维度,$\hat{f}$ 是关于模型参数 $\boldsymbol{\theta}$ 和随机变量 $\boldsymbol{\zeta}$ 的观测值(Realization)的损失函数。

表 1　随机梯度类算法收敛速度的对比一览表

优化器	研究工作	范数	收敛速度	关于 K 是否匹配下界	关于 d 是否最优依赖	备注
经典算法						
SGD	Bottou 等人[22]	ℓ_2	$\mathcal{O}(K^{-1/4})$	√	√	
自适应学习率算法						
AdaGrad	Wang 等人[28]、Liu 等人[29]	ℓ_2	$\mathcal{O}(\mathrm{polylog}(K)\cdot K^{-1/4})$	×	?	近似版本
	Hong 等人[30]	ℓ_2	$\mathcal{O}(\sqrt{d}\,\mathrm{polylog}(K)\cdot K^{-1/4})$	×	?	
RMSProp	Shi 等人[31]	$\min\{\ell_1,\gamma\ell_2\}$	$\mathcal{O}(\mathrm{polylog}(K)\cdot K^{-1/4})$	×	?	
	Zou 等人[32]	ℓ_2	$\mathcal{O}(\mathrm{polylog}(K)\cdot K^{-1/4})$	×	?	
	Li 等人[15]	ℓ_1	$\mathcal{O}(\sqrt{d}\cdot K^{-1/4})$	√	√	较少使用
Adam	Zhang 等人[33]	$\min\{\ell_2,\gamma\ell_2^2\}$	$\mathcal{O}(\mathrm{polylog}(K)\cdot K^{-1/4})$	×	?	
	Zou 等人[32]、Wang 等人[34]	ℓ_2	$\mathcal{O}(\mathrm{polylog}(K)\cdot K^{-1/4})$	×	?	
符号梯度下降类算法						
SignSGD	Bernstein 等人[35]	ℓ_1	$\mathcal{O}(K^{-1/4})$	√	?	假设不合实际
	Sun 等人[36]	ℓ_1	$\mathcal{O}(d\cdot K^{-1/4})$	√	×	
SignSGD-EF	Karimireddy 等人[37]	ℓ_2	$\mathcal{O}(K^{-1/4})$	√	?	较少使用
SignSGD-CF	Safaryan 等人[38]	ℓ_1	$\mathcal{O}(K^{-1/4})$	√	?	较少使用
LION	**Dong 等人(本文)**	$\boldsymbol{\ell_1}$	$\boldsymbol{\mathcal{O}(\sqrt{d}\cdot K^{-1/4})}$	√	√	

注:表中以经典算法、自适应学习率算法和符号梯度下降类算法作为分类,详细探讨了各类算法在不同梯度范数度量下的收敛速度,并给出了全面的对比分析。表中“$\min\{\ell_1,\gamma\ell_2\}$”表示梯度 ℓ_1 范数与 ℓ_2 范数的 γ 倍(γ 为常数)之中较小值的上界。可以清晰地看出,本文所得的收敛速度关于 K 紧密匹配随机梯度类算法的下界,且关于 d 达到了当前研究工作的最优依赖。

2.1　符号约定

在本文中,我们使用粗体符号表示向量,普通非粗体符号表示标量。对于向量 $\boldsymbol{x}$,用 $\boldsymbol{x}^k$ 表示在第 k 次迭代时 $\boldsymbol{x}$ 的值,而 $\boldsymbol{x}_i$ 表示该向量的第 i 个分量。我们用 $\boldsymbol{\theta}$ 表示模型参数,并将随机梯度估计表示为 $\boldsymbol{g}$。对于标量 s,s^p 仍然表示它的 p 次幂。我们使用 $\|\cdot\|_p$ 来表示 ℓ_p 范数,为简便起见,使用 $\|\cdot\|$ 表示欧几里得范数。对于最优性,定义 $f^*=\inf_{\boldsymbol{\theta}\in\mathcal{F}} f(\boldsymbol{\theta})$,其中$\mathcal{F}$是可行域。虽然在非凸情形中优化算法可能不会收敛到 f^*,但我们还是给出了这一定义①。对于渐进性,我们使用 $f(n)=\mathcal{O}(g(n))$ 表示存在常数 $c>0$ 和正整数 N,使得对所有 $n\geqslant N$,总有 $|f(n)|\leqslant c|g(n)|$ 成立;使用 $f(n)=\Theta(g(n))$ 表示存在 $c_1,c_2(c_2\geqslant c_1>0)$ 和正整数 N,使得对所有 $n\geqslant N$,总有 $c_1|g(n)|\leqslant|f(n)|\leqslant c_2|g(n)|$ 成立。最后,我们使用 $\mathrm{polylog}(n)$ 表示多对数函数(Polylogarithmic Function)。

2.2　相关工作

随机梯度类算法的收敛性已在众多研究中得到广泛探讨。在假设 1~3 下(见第 3 节),Bottou 等人已证明,随机梯度下降法(SGD)的收敛速度为[22]

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}^k)\|_2\right]\leqslant\mathcal{O}\left(\frac{\sqrt[4]{\sigma^2L(f(\boldsymbol{\theta}^1)-f^*)}}{K^{1/4}}\right)\quad(3)$$

该结果与非凸优化问题中随机梯度类算法相对于 K 的理论下界相匹配[21]②。尽管其具有良好的理论性质,但许多研究已指出,SGD 已无法应用于当今基于 Transformer 架构的大规模深度学习任务的训练过程[23-25]。

在 SGD 的基础成果上,许多工作研究了自适应学习率算法的收敛速度。这类方法对每一个参数基于其历史梯度信息动态调整相应的学习率,广泛应用于深度学习等非凸优化任务中。对于 AdaGrad 算法[16],Wang 等人[28]、Liu 等人[29]和 Hong 等人[30]分别给出了相应的收敛速度分析。然而,前两个工作的分析结果是基于其近似版本 AdaGrad-Norm[39]完成的,并不严格等价于 AdaGrad 算法;且他们并未给出关于问题维度 d 的严格依赖关系。Hong 等人的工作[30]分析了 AdaGrad 算法本身,经过最优化简后,在 ℓ_2 范数下关于问题维度 d 的收敛速度仍比本文慢 $\mathcal{O}(\sqrt{d})$ 倍;此外,这些工作的收敛速度关于迭代步数 K 的分析结果为 $\mathcal{O}(\mathrm{polylog}(K)\cdot K^{-1/4})$,并不能匹配随机梯度类算法收敛速度的下界。对于 RMSProp 算法[19],Zou 等人[32]和 Shi 等人[31]给出的收敛速度结果同样具有 AdaGrad 分析中的弊端;Li 等人虽然指明了 RMSProp 的收

① 本文不会证明 f 将收敛至 f^*;证明的主要结论是,带约束问题收敛至 KKT 点和无约束问题收敛至驻点的速度均为 $\mathcal{O}(\sqrt{d}K^{-1/4})$。

② 由于 SGD 的收敛速度式(3)匹配了随机梯度类算法的理论下界,因此本文后续内容中与式(3)的对比即为与理论下界的对比。

敛速度关于 K 和 d 的关系[15]，但由于该算法在实际深度学习任务中已较少使用，对实际的指导意义有限。对于 Adam[17] 算法，Zhang 等人[33]、Zou 等人[32]和 Wang 等人[34]分别给出了该算法的收敛速度，但仍然与下界相差 $\mathcal{O}(\text{polylog}(K))$ 因子，且由于 Adam 算法的复杂性，这些工作都未严格指明关于 d 的依赖关系。

另外一系列的研究是围绕符号梯度下降类算法展开的。这类方法最主要的特点是在参数更新式中引入了符号函数，LION 优化器正是其中之一。与 SGD 和自适应学习率算法通常使用的 ℓ_2 范数度量不同，这类算法的收敛性证明大都是在梯度的 ℓ_1 范数下完成的[35-36,40-41]。Bernstein 等人首次正式提出 SignSGD 算法，并设计了分布式训练场合下的部署方式[35]。尽管他们给出了 SignSGD 的收敛性证明，但该证明需要假设批大小(Batchsize)是逐步增加的，大大限制了其在实际场景中的说服力。随后，学界又提出了 SignSGD-EF[37] 和 SignSGD-CF[38] 算法。这些算法的收敛性证明中解耦了与批大小之间的关系，但并未应用于实际的深度学习训练中。之后，Sun 等人在更弱假设下给出了 SignSGD 算法的动量版本的收敛速度证明[36]。具体来说，在与本文相同的假设条件下，以梯度的 ℓ_1 范数为度量，带动量的 SignSGD 算法的收敛速度为

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}^k)\|_1\right]\leqslant\mathcal{O}\left(\frac{f(\boldsymbol{\theta}^1)-f^*}{LK^{1/4}}+\frac{d}{K^{1/4}}\right) \quad (4)$$

然而，该结果关于问题维度 d 并未达到当前研究工作的最优依赖。

作为符号梯度下降类算法的一员，LION 的收敛性质也已得到了初步研究。Xiao 等人给出了一个包含 LION 的随机梯度类算法的框架[42]，虽然他们证明了该框架之下所有算法的收敛性，但并未给出收敛速度的具体结果。Chen 等人创新性地引入了李雅普诺夫(Lyapunov)函数，展示了 LION 的优化动态(Dynamics)，并证明 LION 可作为一般性的方法解决一个盒约束(Box Constrained)优化问题[43]：

$$\min f(\boldsymbol{\theta}),\ \text{s.t.}\ \|\boldsymbol{\theta}\|_\infty\leqslant\frac{1}{\lambda} \quad (5)$$

其中，问题(5)的约束可以看作一种正则化形式，而正则化恰好是机器学习问题中防止过拟合的一项重要技巧[44-45]。

Liu 等人提出了在多种不同的分布式设定下部署 LION 的方法，并给出了相应的收敛性证明[40]。这项工作仅仅证明了 LION 对于一种特定的带约束问题(5)的收敛性，而我们更精确地刻画了 LION 的收敛速度，且该速度在迭代步数 K 上与随机梯度类算法的理论下界相匹配，在问题维度 d 上展现了当前研究工作的最优依赖。此外，我们还提供了对一般无约束问题(2)的收敛速度分析。

2.3 研究贡献

本文的主要贡献是对 LION 优化算法提供了完整的在一种特定的带约束优化问题和广义无约束优化问题下的收敛速度分析。本文证明，在这两种情况下，LION 的收敛速度均为 $\mathcal{O}(\sqrt{d}K^{-1/4})$。在带约束优化问题中，本文的结果表明 LION 的收敛速度匹配了非凸问题下随机梯度类算法关于 K 的理论下界，并且在问题维度 d 上表现出当前研究工作中最优的依赖关系。在无约束设定下，我们首次给出了 LION 的收敛性证明，并刻画了收敛速度，该速度同样具有上述优势。此外，我们的收敛速度是基于梯度 ℓ_1 范数刻画的，该范数与下界式(3)中使用的 ℓ_2 范数有所不同。如式(1)所示，由于优化算法关于问题维度 d 的收敛速度结果与所选取的范数有关，为了与下界进行公平对比，我们还完成了全面而广泛的实验，不仅验证了 LION 的收敛速度关于 K 匹配了理论下界，还在实际深度学习任务中验证梯度的 ℓ_1/ℓ_2 范数比值始终处于 $\Theta(\sqrt{d})$ 的量级。因此，从经验意义上看，我们的速度也与式(3)中所得的关于 d 的结果相匹配。这些结果在验证了 LION 优化器的高效性的同时，更为实际优化任务提供了关键的理论指导。

3 收敛速度分析

LION 优化器最初是由 Chen 等人通过基于遗传算法的搜索策略发现的，是一种基于学习的优化算法。LION 的更新规则如算法 1 所示[26]。

算法 1. LION

1. 初始化 $\boldsymbol{\theta}^1,\boldsymbol{m}^0$
2. FOR $k=1,2,\cdots,K$ DO
3. $\quad\boldsymbol{g}^k\leftarrow\nabla\hat{f}(\boldsymbol{\theta}^k,\boldsymbol{\zeta}^k)$
4. $\quad\boldsymbol{c}^k\leftarrow\beta_1\boldsymbol{m}^{k-1}+(1-\beta_1)\boldsymbol{g}^k$
5. $\quad\boldsymbol{\theta}^{k+1}\leftarrow\boldsymbol{\theta}^k-\eta(\text{sign}(\boldsymbol{c}^k)+\lambda\boldsymbol{\theta}^k)$
6. $\quad\boldsymbol{m}^k\leftarrow\beta_2\boldsymbol{m}^{k-1}+(1-\beta_2)\boldsymbol{g}^k$
7. END FOR

其中 η 为学习率，λ 为权重衰减系数，β_1 和 β_2 为算

法的超参数。尽管 LION 是由启发式的搜索得到的,但其算法原理融合了多种优化器设计的技巧:

(1)符号函数。LION 优化器的更新规则是基于符号函数的,而符号梯度下降类算法恰好是随机梯度类算法的一个重要子分支[46]。先前的研究已表明,基于符号函数的优化算法是解决深度网络训练问题中的一类有效方案[35-38]。

(2)动量增强。动量法早在 1964 年便由 Polyak 等人提出[18],是随机梯度类算法中增强有效梯度信号的一项经典技巧。动量法的有效性已在多种非凸优化问题中得到充分证明。

(3)权重衰减。相比于传统基于目标函数 ℓ_2 正则化的权重衰减方案,LION 采用了解耦的权重衰减(Decoupled Weight Decay)技术。Loshchilov 在 2017 年便认识到了解耦权重衰减在深度学习任务中的优势,并给出了具体原理的详细分析[47]。此外,解耦的权重衰减也是 AdamW[47] 超越传统 Adam 算法[17]的根本原因。

LION 算法最大的特点是在上一步动量 $\boldsymbol{m}^{k-1}$ 和本步随机梯度 $\boldsymbol{g}^k$ 之间使用了两个不同的插值,即第 4 行的 $\boldsymbol{c}^k$ 和第 6 行的 $\boldsymbol{m}^k$。当参数 $\beta_1=\beta_2$ 时,该算法退化为 SignSGD 的动量版本;作者在文章[26]附录的第 L 节指出,采用不同的 β_1 和 β_2 值是 LION 算法能高效运用于多种深度学习任务的关键所在。作者推荐的默认设置为 $(\beta_1,\beta_2)=(0.9,0.99)$;在大模型训练中,常采用 $(\beta_1,\beta_2)=(0.9,0.95)$。学习率 η 和权重衰减系数 λ 通常由使用者自行设定。

为严格分析 LION 优化器的收敛速度,我们首先列出随机梯度类算法的相关文献中最常使用的若干基本假设。这些假设经常用于非凸优化和深度学习领域主流的随机梯度类算法的收敛性分析中,如 SGD[21]、AdaGrad[28-30]、RMSProp[15,32] 和 Adam[48-49] 等等。

假设 1(L −光滑性). 目标函数的梯度是 Lipschitz 连续的:对任意点 $\boldsymbol{x}$ 和 $\boldsymbol{y}$,目标函数 f 满足

$$\|\nabla f(\boldsymbol{x})-\nabla f(\boldsymbol{y})\|\leqslant L\|\boldsymbol{x}-\boldsymbol{y}\| \tag{6}$$

其中 L 是目标函数 f 的 Lipschitz 常数。

假设 2(随机梯度的无偏性). 随机梯度 $\boldsymbol{g}^k$ 是真实梯度 $\nabla f(\boldsymbol{\theta}^k)$ 的无偏估计:

$$\mathbb{E}\left[\boldsymbol{g}^k\right]=\nabla f(\boldsymbol{\theta}^k) \tag{7}$$

假设 3(梯度噪声的有界性). 随机梯度 $\boldsymbol{g}^k$ 的方差是有界的:

$$\mathbb{E}\left[\|\boldsymbol{g}^k-\nabla f(\boldsymbol{\theta}^k)\|^2\right]\leqslant\sigma^2 \tag{8}$$

其中,σ^2 为方差的上界。

3.1　带约束情形

考虑带约束问题 (5)。我们首先引入 Xie 等人工作中的一个引理[50]:

引理 1(论文[50]中引理 3.8 在 ℓ_∞ 范数下的推论). 对带约束问题 (5),$\boldsymbol{\theta}^*$ 为 KKT 点的充分必要条件是

$$\|\boldsymbol{\theta}^*\|_\infty\leqslant\frac{1}{\lambda},\lambda\langle\boldsymbol{\theta}^*,\nabla f(\boldsymbol{\theta}^*)\rangle+\|\nabla f(\boldsymbol{\theta}^*)\|_1=0。$$

我们指出,通过适当初始化 $\boldsymbol{\theta}^1$,LION 算法的迭代序列 $\{\boldsymbol{\theta}^k\}$ 将始终保持在可行域内,并且使得 $\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle+\|\nabla f(\boldsymbol{\theta}^k)\|_1$ 项一直保持非负(见定理 2)。结合引理 1 的知识和定理 2 提供的上界,我们可推出 LION 算法最多需要 $\mathcal{O}(\sqrt{d}K^{-1/4})$ 步即可收敛到问题 (5)的 KKT 点,如推论 3 所示。

定理 2. 假设条件 1～3 成立。设 $\beta_1=1-\frac{c_1}{\sqrt{K}}$,$\beta_2=1-\frac{c_2}{\sqrt{K}}$,$\eta=\frac{c_3}{\sqrt{d}K^{3/4}}$,且 $\|\boldsymbol{\theta}^1\|_\infty\leqslant\frac{1}{\lambda}$,其中 c_1、c_2 和 c_3 与 K 和 d 无关。则对于算法 1,我们有

(i) $\|\boldsymbol{\theta}^k\|_\infty\leqslant\frac{1}{\lambda},\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle+\|\nabla f(\boldsymbol{\theta}^k)\|_1\geqslant 0$,

以及

$$\begin{aligned}\text{(ii)}\quad &\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle+\|\nabla f(\boldsymbol{\theta}^k)\|_1\right]\\ &\leqslant\frac{(f(\boldsymbol{\theta}^1)-f^*)\sqrt{d}}{c_3K^{1/4}}+\frac{2\sigma\sqrt{d}}{c_2K^{1/2}}+\frac{4Lc_3\sqrt{d}}{c_2K^{1/4}}\\ &\quad+\frac{2\sigma(2c_1+c_2)\sqrt{d}}{\sqrt{c_2}K^{1/4}}+\frac{2Lc_3\sqrt{d}}{K^{3/4}}.\end{aligned}$$

由于常数 c_1、c_2 和 c_3 与 K 和 d 无关,定理 2 在一些特定取值下转化为推论 3。

推论 3. 在定理 2 的设置下,设 $c_1=c_2=\frac{\sqrt{L(f(\boldsymbol{\theta}^1)-f^*)}}{\sigma}$,$c_3=\frac{(f(\boldsymbol{\theta}^1)-f^*)^{3/4}}{L^{1/4}\sigma^{1/2}}$,且 $K\geqslant\max\left\{\frac{\sigma^6}{L^3(f(\boldsymbol{\theta}^1)-f^*)^3},\frac{L(f(\boldsymbol{\theta}^1)-f^*)}{\sigma^2}\right\}$,则有

$$\begin{aligned}&\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle+\|\nabla f(\boldsymbol{\theta}^k)\|_1\right]\\ &\leqslant\frac{15\sqrt{d}}{K^{1/4}}\sqrt[4]{\sigma^2L(f(\boldsymbol{\theta}^1)-f^*)}\end{aligned} \tag{9}$$

推论 3 表明,LION 算法在解决带约束优化问题 (5)的过程中,就迭代次数 K 而言实现了相同于式(3)中所刻画的 SGD 算法的收敛速度。由于我们已知式(3)与非凸优化中随机梯度类算法的理论下

界相匹配[21]，故推论 3 中的界同样与该下界在 K 上是匹配的。关于问题维度 d 方面，尽管学界尚未建立随机梯度类算法关于 d 的收敛速度下界，本文的结果提供了当下研究中关于 d 的最优依赖关系（详细对比见表 1）。此外，我们的不等式约束的是梯度的 ℓ_1 范数 $\|\nabla f(\boldsymbol{\theta}^k)\|_1$，与下界所使用的梯度 ℓ_2 范数 $\|\nabla f(\boldsymbol{\theta}^k)\|_2$ 有所不同。根据 (1)式，当二者相近，即 $\|\nabla f(\boldsymbol{\theta}^k)\|_1 \approx \|\nabla f(\boldsymbol{\theta}^k)\|_2$ 时，采用梯度的 ℓ_1 或 ℓ_2 范数并无区别；而当二者相差较远，即 $\|\nabla f(\boldsymbol{\theta}^k)\|_1 \approx \sqrt{d}\,\|\nabla f(\boldsymbol{\theta}^k)\|_2$ 时，采用梯度的 ℓ_1 范数度量相比于 ℓ_2 范数会多出 $\sqrt{d}$ 的因子。定义 ℓ_1/ℓ_2 梯度范数比为

$$r=\frac{\|\nabla f(\boldsymbol{\theta})\|_1}{\|\nabla f(\boldsymbol{\theta})\|_2} \tag{10}$$

在本文的第 5 节中，我们在多种视觉和语言模态的深度学习任务上进行了一系列实验，通过具体的实验验证在深度模型的训练过程中梯度范数比 r 处于 $\Theta(\sqrt{d})$ 的水平（而不是常数量级 $\Theta(1)$），从而可以约去 $\mathcal{O}(\sqrt{d}K^{-1/4})$ 中的 $\sqrt{d}$ 项。因此，我们的界在实证意义上也与随机梯度类算法的收敛速度下界式 (3)关于 d 的结论相匹配。

3.2 无约束情形

本节介绍本文工作的另一个重要贡献：首次为 LION 优化器在无约束问题下的收敛性提供了严格保证，并详细分析了收敛速度。为了得出这一结果，我们对目标函数 f 做出了额外的强制性假设，该假设在许多理论分析与实际应用场景下均成立。

假设 4（强制性）. f 为强制(Coercive)函数：

$$\lim_{\|\boldsymbol{\theta}\|\to\infty} f(\boldsymbol{\theta})\to\infty \tag{11}$$

在理论分析方面，非凸优化算法理论研究的相关文献中广泛使用强制性假设，用于许多理论性质的分析[42]。在实际应用方面，目标函数 f 的强制性在许多领域的优化问题中广泛存在，如物理学中的能量最小化问题[51-52]和交通与网络流系统中的最优路径问题[53-55]等。对深度学习优化问题而言，使用了正则化技巧的目标函数都具备强制性[44-45]；此外，许多深度学习任务的目标函数本身就满足假设 4。例如，在回归任务[56-57]和图像重建任务[58-60]中，常用的均方误差(Mean Squared Error, MSE)损失就满足这一假设。

值得指出的是，在符号梯度下降类算法中，SignSGD 算法[35]及其变体 SignSGD-EF[37]和 SignSGD-CF[38]均没有设计解耦的权重衰减机制（即不存在权重衰减系数 λ），使得这些算法在无约束情形下的收敛速度结果在假设 1～3 的条件下即可成立；LION 引入的权重衰减机制使得我们在分析广义的无约束优化问题时需要对目标函数 f 做出强制性假设。当移除该机制，即 $\lambda=0$ 时，由于此时式(5)中的约束已不复存在，从而使得 3.1 节中仅依赖于假设 1～3 的结论成为不含权重衰减的 LION 在无约束问题下的收敛速度结果。

在无约束问题中，如果我们初始化 $\|\boldsymbol{\theta}^1\|_\infty\leqslant\frac{1}{\lambda}$，则定理 2 中的所有假设和条件在此均得到满足，因此其推导过程中所有的结论在此也适用。基于定理 2 和式(14)，可得

$$\mathbb{E}\left[f(\boldsymbol{\theta}^k)\right]-f^*\leqslant f(\boldsymbol{\theta}^1)-f^*+\frac{2c_3\sigma}{c_2}+\frac{4Lc_3^2}{c_2}+\frac{2c_3\sigma(2c_1+c_2)}{\sqrt{c_2}}+2Lc_3^2 \tag{12}$$

不等式 (12)表明，$\mathbb{E}\left[f(\boldsymbol{\theta}^k)\right]$ 对任意 k 是有界的。结合假设 4，可推出 $\|\boldsymbol{\theta}^k\|_\infty\leqslant C$ 对任意 k 成立。进一步设 $\lambda\leqslant\frac{1}{2C}$，得到的不等式 $\lambda\|\boldsymbol{\theta}^k\|_\infty\leqslant\frac{1}{2}$ 表明 LION 的迭代序列不会触及问题(5)的约束边界，从而在本质上转化为无约束优化问题。

本文证明，在无约束情形下，LION 同样至多需要 $\mathcal{O}(\sqrt{d}K^{-1/4})$ 步即可收敛至问题(2)的驻点，如推论 5 所示。

定理 4. 假设条件 1～4 成立。设 $\beta_1=1-\frac{c_1}{\sqrt{K}}$，$\beta_2=1-\frac{c_2}{\sqrt{K}}$，$\eta=\frac{c_3}{\sqrt{d}K^{3/4}}$，且 $\lambda\leqslant\frac{1}{2C}$，其中 c_1、c_2 和 c_3 与 K 和 d 无关。初始化 $\|\boldsymbol{\theta}^1\|_\infty\leqslant\frac{1}{\lambda}$，那么对于算法 1，我们有

$$\frac{1}{2K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}^k)\|_1\right]\leqslant\sqrt{d}\,\frac{f(\boldsymbol{\theta}^1)-f^*}{c_3K^{1/4}}+\frac{2\sigma\sqrt{d}}{c_2K^{1/2}}+\frac{4Lc_3\sqrt{d}}{c_2K^{1/4}}+\frac{2\sigma(2c_1+c_2)\sqrt{d}}{\sqrt{c_2}K^{1/4}}+\frac{2Lc_3\sqrt{d}}{K^{3/4}}。$$

类似地，推论 5 描述了 LION 收敛到问题 (2)的驻点的速度：

推论 5. 在定理 4 的设置下，如果我们设 $c_1=c_2=\frac{\sqrt{L(f(\boldsymbol{\theta}^1)-f^*)}}{\sigma}$，$c_3=\frac{(f(\boldsymbol{\theta}^1)-f^*)^{3/4}}{L^{1/4}\sigma^{1/2}}$，且 $K\geqslant\max\left\{\frac{\sigma^6}{L^3(f(\boldsymbol{\theta}^1)-f^*)^3},\frac{L(f(\boldsymbol{\theta}^1)-f^*)}{\sigma^2}\right\}$，则

$$\frac{1}{2K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}^k)\|_1\right]\leqslant\frac{15\sqrt{d}}{K^{1/4}}\sqrt[4]{\sigma^2 L(f(\boldsymbol{\theta}^1)-f^*)}\tag{13}$$

推论 5 表明，在无约束问题 (2) 上，LION 与 SGD 关于迭代步数 K 以相同的速度收敛至驻点。

4　定理证明

本节给出第 3 章中核心定理的证明过程。思路如下：首先证明定理 2 的第(i)部分，再给出 $\boldsymbol{\delta}^k=\boldsymbol{c}^k-\nabla f(\boldsymbol{\theta}^k)$ 范数的上界，最后完成其余定理的证明。

4.1　定理 2 (i)的证明

证明.　借鉴 Xie 等人的工作[50]，根据 $\boldsymbol{\theta}^k$ 的更新公式，可得

$$\begin{aligned}\|\boldsymbol{\theta}^{k+1}\|_\infty-\frac{1}{\lambda}&=\|(1-\eta\lambda)\boldsymbol{\theta}^k-\eta\,\mathrm{sign}(\boldsymbol{c}^k)\|_\infty-\frac{1}{\lambda}\\&\leqslant(1-\eta\lambda)\|\boldsymbol{\theta}^k\|_\infty+\eta\|\mathrm{sign}(\boldsymbol{c}^k)\|_\infty-\frac{1}{\lambda}\\&=(1-\eta\lambda)\|\boldsymbol{\theta}^k\|_\infty+\eta-\frac{1}{\lambda}\\&=(1-\eta\lambda)\left(\|\boldsymbol{\theta}^k\|_\infty-\frac{1}{\lambda}\right),\end{aligned}$$

从 $\|\boldsymbol{\theta}^1\|_\infty\leqslant\frac{1}{\lambda}$ 可直接推出第一条结论；对第二条结论，我们有

$$\begin{aligned}&\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle+\|\nabla f(\boldsymbol{\theta}^k)\|_1\\&\geqslant-\lambda\|\nabla f(\boldsymbol{\theta}^k)\|_1\|\boldsymbol{\theta}^k\|_\infty+\|\nabla f(\boldsymbol{\theta}^k)\|_1\geqslant0,\end{aligned}$$

证毕。

4.2　$\|\boldsymbol{\delta}^k\|$ 的上界引理

引理 6. 假设条件 1-3 成立。记 $\boldsymbol{\delta}^k=\boldsymbol{c}^k-\nabla f(\boldsymbol{\theta}^k)$。则对于算法 1，我们有

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\boldsymbol{\delta}^k\|\right]\leqslant\frac{\sigma}{K(1-\beta_2)}+\frac{2L\eta\sqrt{d}}{1-\beta_2}+(|\beta_1-\beta_2|+1-\beta_1)\cdot\frac{\sigma}{\sqrt{1-\beta_2}}。$$

证明.　再记 $\boldsymbol{\xi}^k=\boldsymbol{g}^k-\nabla f(\boldsymbol{\theta}^k)$。我们希望将 $\boldsymbol{\delta}^k$ 整理为 $\nabla f(\boldsymbol{\theta}^t)-\nabla f(\boldsymbol{\theta}^{t-1})$ 和 $\boldsymbol{\xi}^t$ 等变量的线性组合的形式，并分别利用假设 1 和假设 3 进行放缩。根据算法 1 的更新规则，可得

$$\begin{aligned}\boldsymbol{\delta}^k&=\beta_1\boldsymbol{m}^{k-1}+(1-\beta_1)\boldsymbol{g}^k-\nabla f(\boldsymbol{\theta}^k)\\&=\beta_1\beta_2\boldsymbol{m}^{k-2}+\beta_1(1-\beta_2)\boldsymbol{g}^{k-1}+(1-\beta_1)\boldsymbol{g}^k-\nabla f(\boldsymbol{\theta}^k)\\&=\beta_2(\boldsymbol{c}^{k-1}-(1-\beta_1)\boldsymbol{g}^{k-1})+\beta_1(1-\beta_2)\boldsymbol{g}^{k-1}+(1-\beta_1)\boldsymbol{g}^k-\nabla f(\boldsymbol{\theta}^k)\\&=\beta_2(\boldsymbol{\delta}^{k-1}+\nabla f(\boldsymbol{\theta}^{k-1}))-\beta_2(1-\beta_1)\boldsymbol{g}^{k-1}+\beta_1(1-\beta_2)\boldsymbol{g}^{k-1}+(1-\beta_1)\boldsymbol{g}^k-\nabla f(\boldsymbol{\theta}^k)\\&=\beta_2(\boldsymbol{\delta}^{k-1}+\nabla f(\boldsymbol{\theta}^{k-1}))+(\beta_1-\beta_2)(\boldsymbol{\xi}^{k-1}+\nabla f(\boldsymbol{\theta}^{k-1}))+(1-\beta_1)(\boldsymbol{\xi}^k+\nabla f(\boldsymbol{\theta}^k))-\nabla f(\boldsymbol{\theta}^k)\\&=\beta_2\boldsymbol{\delta}^{k-1}-\beta_1(\nabla f(\boldsymbol{\theta}^k)-\nabla f(\boldsymbol{\theta}^{k-1}))+(\beta_1-\beta_2)\boldsymbol{\xi}^{k-1}+(1-\beta_1)\boldsymbol{\xi}^k\\&=\beta_2^{k-1}\boldsymbol{\delta}^1+\sum_{t=2}^{k}\beta_2^{k-t}(-\beta_1(\nabla f(\boldsymbol{\theta}^t)-\nabla f(\boldsymbol{\theta}^{t-1}))+(\beta_1-\beta_2)\boldsymbol{\xi}^{t-1}+(1-\beta_1)\boldsymbol{\xi}^t)\\&=\beta_2^{k-1}\boldsymbol{\delta}^1-\beta_1\sum_{t=2}^{k}\beta_2^{k-t}(\nabla f(\boldsymbol{\theta}^t)-\nabla f(\boldsymbol{\theta}^{t-1}))+(\beta_1-\beta_2)\sum_{t=2}^{k}\beta_2^{k-t}\boldsymbol{\xi}^{t-1}+(1-\beta_1)\sum_{t=2}^{k}\beta_2^{k-t}\boldsymbol{\xi}^t。\end{aligned}$$

不等式两侧取数学期望，可得

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{\delta}^k\|\right]&\leqslant\beta_2^{k-1}\mathbb{E}\left[\|\boldsymbol{\delta}^1\|\right]+\underbrace{\beta_1\sum_{t=2}^{k}\beta_2^{k-t}\,\mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}^t)-\nabla f(\boldsymbol{\theta}^{t-1})\|\right]}_{\text{(a)项}}\\&+\underbrace{\mathbb{E}\left[\left\|(\beta_1-\beta_2)\sum_{t=2}^{k}\beta_2^{k-t}\boldsymbol{\xi}^{t-1}+(1-\beta_1)\sum_{t=2}^{k}\beta_2^{k-t}\boldsymbol{\xi}^t\right\|\right]}_{\text{(b)项}}。\end{aligned}$$

对(a)项，应用定理 2 (i)的第一个不等式，可得

$$\begin{aligned}\text{(a) 项}&\leqslant L\sum_{t=2}^{k}\beta_2^{k-t}\mathbb{E}\left[\|\boldsymbol{\theta}^t-\boldsymbol{\theta}^{t-1}\|\right]\\&=L\eta\sum_{t=2}^{k}\beta_2^{k-t}\mathbb{E}\left[\|\mathrm{sign}(\boldsymbol{c}^{t-1})+\lambda\boldsymbol{\theta}^{t-1}\|\right]\\&\leqslant2L\eta\sqrt{d}\sum_{t=2}^{k}\beta_2^{k-t}\\&\leqslant\frac{2L\eta\sqrt{d}}{1-\beta_2}。\end{aligned}$$

对(b)项，可作如下放缩：

$$\begin{aligned}\text{(b) 项}&\leqslant|\beta_1-\beta_2|\,\mathbb{E}\left[\left\|\sum_{t=2}^{k}\beta_2^{k-t}\boldsymbol{\xi}^{t-1}\right\|\right]+(1-\beta_1)\,\mathbb{E}\left[\left\|\sum_{t=2}^{k}\beta_2^{k-t}\boldsymbol{\xi}^t\right\|\right]\\&\leqslant|\beta_1-\beta_2|\sqrt{\mathbb{E}\left[\left\|\sum_{t=2}^{k}\beta_2^{k-t}\boldsymbol{\xi}^{t-1}\right\|^2\right]}+(1-\beta_1)\sqrt{\mathbb{E}\left[\left\|\sum_{t=2}^{k}\beta_2^{k-t}\boldsymbol{\xi}^t\right\|^2\right]}\\&=|\beta_1-\beta_2|\sqrt{\sum_{t=2}^{k}\beta_2^{2(k-t)}\,\mathbb{E}\left[\|\boldsymbol{\xi}^{t-1}\|^2\right]}+(1-\beta_1)\sqrt{\sum_{t=2}^{k}\beta_2^{2(k-t)}\,\mathbb{E}\left[\|\boldsymbol{\xi}^t\|^2\right]}\end{aligned}$$

$$=|\beta_1-\beta_2|\sqrt{\sigma^2\sum_{t=2}^{k}\beta_2^{2(k-t)}}+(1-\beta_1)\sqrt{\sigma^2\sum_{t=2}^{k}\beta_2^{2(k-t)}}$$
$$\leqslant|\beta_1-\beta_2|+(1-\beta_1)\cdot\frac{\sigma}{\sqrt{1-\beta_2^2}}$$
$$\leqslant|\beta_1-\beta_2|+(1-\beta_1)\cdot\frac{\sigma}{\sqrt{1-\beta_2}}。$$

其中,第二个不等号利用了琴生不等式(Jensen's Inequality)的直接推论:由于 $f(x)=x^2$ 是凸函数,根据琴生不等式,对随机变量 X 有 $[\mathbb{E}(X)]^2\leqslant\mathbb{E}(X^2)$,即 $\mathbb{E}(X)\leqslant\sqrt{\mathbb{E}(X^2)}$ 成立。将(a)项和(b)项代回,可得

$$\mathbb{E}[\|\boldsymbol{\delta}^k\|]\leqslant\beta_2^{k-1}\mathbb{E}[\|\boldsymbol{\delta}^1\|]+\frac{2L\eta\sqrt{d}}{1-\beta_2}+(|\beta_1-\beta_2|+(1-\beta_1))\cdot\frac{\sigma}{\sqrt{1-\beta_2}},$$

以及

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}[\|\boldsymbol{\delta}^k\|]\leqslant\frac{1}{K(1-\beta_2)}\mathbb{E}[\|\boldsymbol{\delta}^1\|]+\frac{2L\eta\sqrt{d}}{1-\beta_2}+(|\beta_1-\beta_2|+(1-\beta_1))\cdot\frac{\sigma}{\sqrt{1-\beta_2}}。$$

初始化 $\boldsymbol{m}^0=\boldsymbol{g}^1$,则有 $\mathbb{E}[\|\boldsymbol{\delta}^1\|]=\mathbb{E}[\|\boldsymbol{g}^1-\nabla f(\boldsymbol{\theta}^1)\|]\leqslant\sigma$。

证毕。

4.3 定理 2 (ii)的证明

证明. 根据梯度的 Lipschitz 性质,我们有

$$\begin{aligned}
&f(\boldsymbol{\theta}^{k+1})-f(\boldsymbol{\theta}^k)\\
&\leqslant\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^k\rangle+\frac{L}{2}\|\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^k\|^2\\
&=-\eta\langle\nabla f(\boldsymbol{\theta}^k),\mathrm{sign}(\boldsymbol{c}^k)+\lambda\boldsymbol{\theta}^k\rangle+\frac{L\eta^2}{2}\|\mathrm{sign}(\boldsymbol{c}^k)+\lambda\boldsymbol{\theta}^k\|^2\\
&=-\eta\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle-\eta\langle\nabla f(\boldsymbol{\theta}^k),\mathrm{sign}(\nabla f(\boldsymbol{\theta}^k))\rangle\\
&\quad-\eta\langle\nabla f(\boldsymbol{\theta}^k),\mathrm{sign}(\boldsymbol{c}^k)-\mathrm{sign}(\nabla f(\boldsymbol{\theta}^k))\rangle+\frac{L\eta^2}{2}\|\mathrm{sign}(\boldsymbol{c}^k)+\lambda\boldsymbol{\theta}^k\|^2\\
&=-\eta\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle-\eta\|\nabla f(\boldsymbol{\theta}^k)\|_1-\eta\langle\nabla f(\boldsymbol{\theta}^k),\mathrm{sign}(\boldsymbol{c}^k)-\mathrm{sign}(\nabla f(\boldsymbol{\theta}^k))\rangle+\frac{L\eta^2}{2}\|\mathrm{sign}(\boldsymbol{c}^k)+\lambda\boldsymbol{\theta}^k\|^2\\
&\leqslant-\eta\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle-\eta\|\nabla f(\boldsymbol{\theta}^k)\|_1+\eta\sum_{i=1}^{d}|\nabla_i f(\boldsymbol{\theta}^k)|\,|\mathrm{sign}(\boldsymbol{c}_i^k)-\mathrm{sign}(\nabla_i f(\boldsymbol{\theta}^k))|\\
&\quad+2dL\eta^2。
\end{aligned}$$

当 $\mathrm{sign}(\boldsymbol{c}_i^k)=\mathrm{sign}(\nabla_i f(\boldsymbol{\theta}^k))$ 时,我们有

$$|\nabla_i f(\boldsymbol{\theta}^k)|\,|\mathrm{sign}(\boldsymbol{c}_i^k)-\mathrm{sign}(\nabla_i f(\boldsymbol{\theta}^k))|=0。$$

当 $\mathrm{sign}(\boldsymbol{c}_i^k)\neq\mathrm{sign}(\nabla_i f(\boldsymbol{\theta}^k))$ 时,可得 $\boldsymbol{c}_i^k\,\nabla_i f(\boldsymbol{\theta}^k)\leqslant0$,以及

$$|\nabla_i f(\boldsymbol{\theta}^k)|\,|\mathrm{sign}(\boldsymbol{c}_i^k)-\mathrm{sign}(\nabla_i f(\boldsymbol{\theta}^k))|=2|\nabla_i f(\boldsymbol{\theta}^k)|\leqslant2|\nabla_i f(\boldsymbol{\theta}^k)-\boldsymbol{c}_i^k|。$$

结合上述两种情况,记 $\boldsymbol{\delta}^k=\boldsymbol{c}^k-\nabla f(\boldsymbol{\theta}^k)$,可得

$$\begin{aligned}
&f(\boldsymbol{\theta}^{k+1})-f(\boldsymbol{\theta}^k)\\
&\leqslant-\eta\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle-\eta\|\nabla f(\boldsymbol{\theta}^k)\|_1+2\eta\|\boldsymbol{\delta}^k\|_1+2dL\eta^2\\
&\leqslant-\eta\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle-\eta\|\nabla f(\boldsymbol{\theta}^k)\|_1+2\eta\sqrt{d}\|\boldsymbol{\delta}^k\|+2dL\eta^2。
\end{aligned}$$

不等式两侧取期望,对 $k=1,\cdots,K$ 求和,并应用引理 6,可得

$$\begin{aligned}
&\mathbb{E}[f(\boldsymbol{\theta}^{K+1})]-f(\boldsymbol{\theta}^1)\\
&\leqslant-\eta\sum_{k=1}^{K}\mathbb{E}[\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle+\|\nabla f(\boldsymbol{\theta}^k)\|_1]+2\eta\sqrt{d}\sum_{k=1}^{K}\mathbb{E}[\|\boldsymbol{\delta}^k\|]+2KdL\eta^2\\
&\leqslant-\eta\sum_{k=1}^{K}\mathbb{E}[\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle+\|\nabla f(\boldsymbol{\theta}^k)\|_1]+2\eta\sqrt{d}\left(\frac{\sigma}{1-\beta_2}+\frac{2KL\eta\sqrt{d}}{1-\beta_2}+(|\beta_1-\beta_2|+1-\beta_1)\cdot\frac{K\sigma}{\sqrt{1-\beta_2}}\right)+2KdL\eta^2。
\end{aligned}$$

设 $\beta_1=1-\frac{c_1}{\sqrt{K}}$,$\beta_2=1-\frac{c_2}{\sqrt{K}}$,$\eta=\frac{c_3}{\sqrt{d}K^{3/4}}$,代入得

$$\begin{aligned}
&\mathbb{E}[f(\boldsymbol{\theta}^{K+1})]-f^*+\eta\sum_{k=1}^{K}\mathbb{E}[\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle+\|\nabla f(\boldsymbol{\theta}^k)\|_1]\\
&\leqslant f(\boldsymbol{\theta}^1)-f^*+2\eta\sqrt{d}\left(\frac{\sigma}{1-\beta_2}+\frac{2KL\eta\sqrt{d}}{1-\beta_2}+(|\beta_1-\beta_2|+1-\beta_1)\cdot\frac{K\sigma}{\sqrt{1-\beta_2}}\right)+2KdL\eta^2\\
&=f(\boldsymbol{\theta}^1)-f^*+\frac{2c_3\sigma}{c_2K^{1/4}}+\frac{4Lc_3^2}{c_2}+(|c_1-c_2|+c_1)\cdot\frac{2c_3\sigma}{\sqrt{c_2}}+\frac{2Lc_3^2}{K^{1/2}}\\
&\leqslant f(\boldsymbol{\theta}^1)-f^*+\frac{2c_3\sigma}{c_2K^{1/4}}+\frac{4Lc_3^2}{c_2}+\frac{2c_3\sigma(2c_1+c_2)}{\sqrt{c_2}}+\frac{2Lc_3^2}{K^{1/2}}
\end{aligned}\tag{14}$$

以及

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle+\|\nabla f(\boldsymbol{\theta}^k)\|_1\right]$$
$$\leqslant\sqrt{d}\,\frac{f(\boldsymbol{\theta}^1)-f^*}{c_3K^{1/4}}+\frac{2\sigma\sqrt{d}}{c_2K^{1/2}}+\frac{4Lc_3\sqrt{d}}{c_2K^{1/4}}$$
$$+\frac{2\sigma(2c_1+c_2)\sqrt{d}}{\sqrt{c_2}K^{1/4}}+\frac{2Lc_3\sqrt{d}}{K^{3/4}}。$$

证毕。

4.4　定理 4 的证明

证明.　如 3.2 节所述,尽管本节研究无约束问题的情形,由于定理 2 中的所有假设和条件在此均满足,故其证明过程中的所有中间结论在此也同样适用。根据不等式 (12)可知,$\mathbb{E}\left[f(\boldsymbol{\theta}^k)\right]$ 对任意 k 一致有界。结合假设 4,可以得出存在一个常数 C,使得 $\|\boldsymbol{\theta}^k\|_\infty\leqslant C$ 对任意 k 恒成立。设 $\lambda\leqslant\frac{1}{2C}$,则有

$$\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle\geqslant-\lambda\|\nabla f(\boldsymbol{\theta}^k)\|_1\|\boldsymbol{\theta}^k\|_\infty$$
$$\geqslant-\frac{1}{2}\|\nabla f(\boldsymbol{\theta}^k)\|_1。$$

应用定理 2,可得

$$\frac{1}{2K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}^k)\|_1\right]$$
$$=\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[-\frac{1}{2}\|\nabla f(\boldsymbol{\theta}^k)\|_1+\|\nabla f(\boldsymbol{\theta}^k)\|_1\right]$$
$$\leqslant\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle+\|\nabla f(\boldsymbol{\theta}^k)\|_1\right]$$
$$\leqslant\sqrt{d}\,\frac{f(\boldsymbol{\theta}^1)-f^*}{c_3K^{1/4}}+\frac{2\sigma\sqrt{d}}{c_2K^{1/2}}+\frac{4Lc_3\sqrt{d}}{c_2K^{1/4}}$$
$$+\frac{2\sigma(2c_1+c_2)\sqrt{d}}{\sqrt{c_2}K^{1/4}}+\frac{2Lc_3\sqrt{d}}{K^{3/4}}。$$

证毕。

5　实验验证

在本节中,我们在多种视觉和语言模态的深度学习任务上进行了充分的实验,从实践中证明了 LION 相较于已知匹配理论下界的 SGD 的优越性。我们还统计了各项任务优化过程中的梯度范数比 r,确认这一比值与模型大小 d 的平方根大致成正比关系:$r=\Theta(\sqrt{d})$,即在实证意义上可以约去 $\mathcal{O}(\sqrt{d}K^{-1/4})$ 中的 $\sqrt{d}$ 项,从而说明本文的收敛速度与下界式(3)关于 d 同样匹配①。此外,为进一步说明实验与理论结果的一致性,在附录 A.1 节中,我们针对简单的非凸优化问题和复杂的深度学习问题验证了 LION 的收敛速度与理论预测较为吻合的事实,并更加细致地分析了 LION 相比于 SGD 优越性背后的原因。

对于视觉任务,我们在 CIFAR-10[61]、CIFAR-100[61] 和 ImageNet-1K[62] 数据集上训练了 ResNet18[63]、ResNet50[63] 和 ViT-S[64] 模型,共计 9 组实验。记数据集的大小为 $|S|$,不同于传统的训练范式,我们计算了整个训练集上的平均样本损失:

$$f(\boldsymbol{\theta})=\frac{1}{|S|}\sum_{k=1}^{|S|}\hat{f}(\boldsymbol{\theta},\boldsymbol{\zeta}_k)\tag{15}$$

并计算全梯度 $\nabla f(\boldsymbol{\theta})$,以统计整个训练集 S 的完整信息并得到梯度范数比 r 的无噪声精确测量。我们将训练过程中一个完整的轮次(Epoch)划分为训练轮、统计轮和测试轮。其中训练轮与典型的深度学习范式一致,将训练集分为不同的批(Batch),分批计算随机梯度,并使用该梯度更新模型的参数;在统计轮中,由于式(15)中关于 f 的梯度计算是线性运算,因此我们可以在保持参数 $\boldsymbol{\theta}$ 不变的情况下,累计各批所对应的损失函数值与相应的梯度信息,从而得到 $f(\boldsymbol{\theta})$ 和 $\nabla f(\boldsymbol{\theta})$ 的精确值;在测试轮中,我们使用测试集上的 Top-1 图像分类准确率来评估模型的性能。

图 1 展示了 ResNet18[63]、ResNet50[63] 以及 ViT-S[64] 模型在 CIFAR-100[61] 数据集上的实验结果。结果显示,LION 相比于 SGD 在所有任务上都实现了更低的训练误差和更高的图像识别准确率,对应于深度学习模型更强的性能。此外,通过观察图 1 中的(b)、(d)和(f)子图中的梯度范数比可知,LION 优化过程中的梯度范数比与问题维度的平方呈现正比的关系,即 $r=\Theta(\sqrt{d})$。这些结果有效证明了 LION 在处理大规模的复杂优化问题的普适性和优越性。

此外,我们还完成了 CIFAR-10[61] 和 ImageNet-1K[62] 数据集上的实验,并将对应的结果整理在附录 A.2 中。这些补充实验进一步展示了 LION 在多种任务中的优越性能,充分证实了我们的结论。为便于实验复现,我们在附录 B 中介绍了具体的实验背景并明确了详细的超参数设定。

① 本文实验的目的是比较 LION 与已知匹配理论下界的 SGD 的实验结果,以验证本文的核心结论,而不是说明在这些任务上 LION 比其他常用的深度网络训练算法(如 AdamW 等)效果更佳. 有关这类比较,可参考 LION 优化器的设计原文[26]。

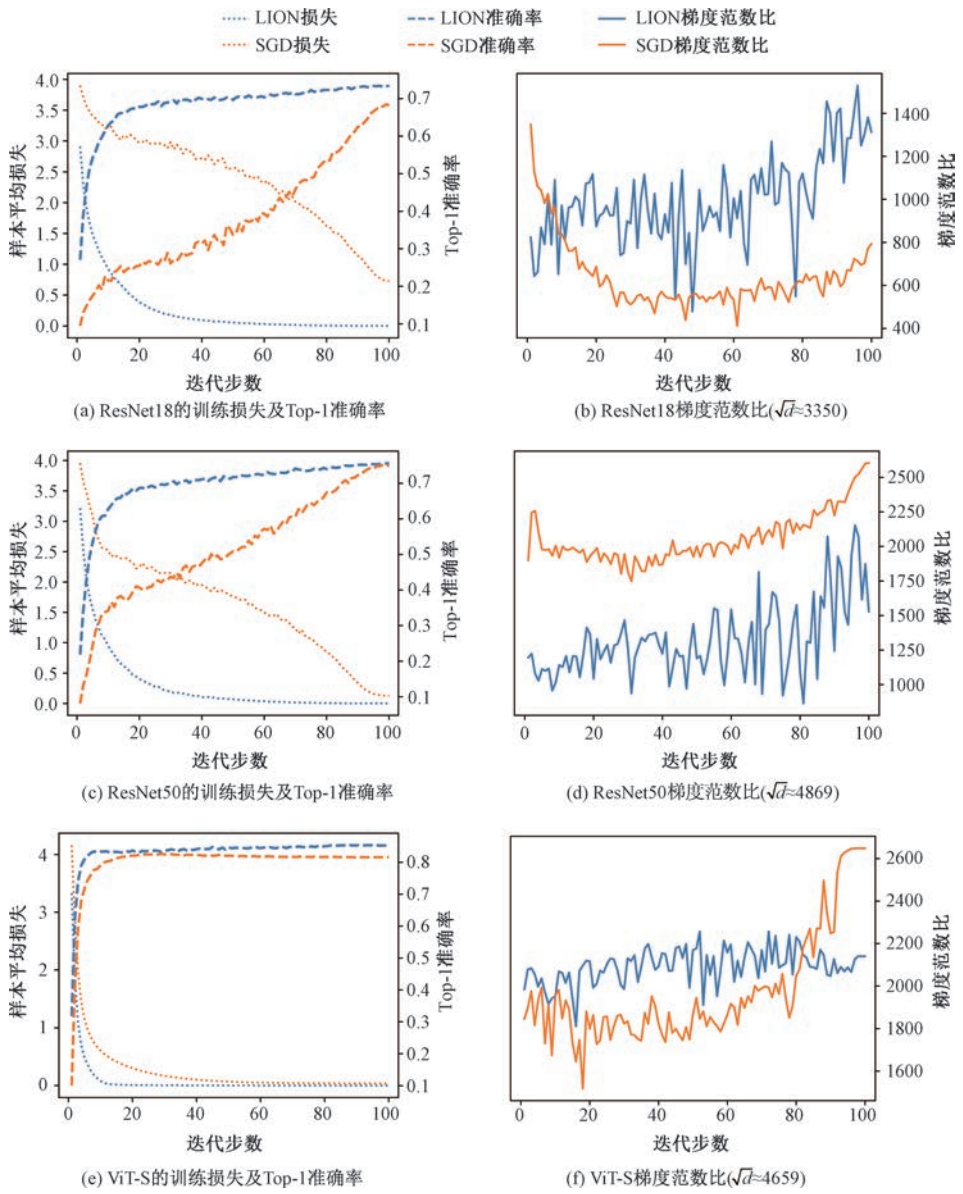


图 1 ResNet18[63]、ResNet50[63]和 ViT-S[64]模型在 CIFAR-100 数据集[61]上的实验结果(子图(a)、(c)和(e)展示了训练过程中的样本平均损失和测试时 Top-1 分类准确率,子图(b)、(d)和(f)展示了梯度范数比的变化过程。)

对于语言任务,我们在 OpenWebText[65]数据集上预训练了经典的 BERT-Small[66]、BERT-Base[66]以及 GPT-2 Small[67]和 GPT-2 Medium[67]模型。对于 BERT 模型,我们将损失函数定义为掩码语言建模(Masked Language Modeling, MLM)损失和句子顺序预测(Sentence Order Prediction, SOP)损失之和,并以测试集上的总损失作为性能评价指标。对于 GPT-2 模型,定义损失函数为语言模型(Language Model, LM)损失,即预测的下一个词概率与实际词之间的交叉熵损失。同样地,我们将性能评价指标定义为测试集上的 LM 损失。

与视觉任务类似,为了计算 $f(\boldsymbol{\theta})$ 和 $\nabla f(\boldsymbol{\theta})$,我们也修改了默认的训练范式。然而,当今大语言模型训练的数据集极其庞大,例如 OpenWebText[65]的训练集大约包含 90 亿个分词(Token)。由于计算资源的限制,我们无法给出整个数据集 S 上的损失函数及其梯度的精确值,而是通过汇总 100 个批次的结果来给出这些值的一个近似估计。进一步的实验表明,将累积的批次数量修改为 1000 或 5000 均不影响实验结论。

图 2 和图 3 分别展示了 BERT 和 GPT-2 模型在 OpenWebText[65]的训练集和测试集上的样本平均损失,以及训练过程中的梯度范数比变化情况。这些结果再次证明了 LION 相比于 SGD 的优越性。

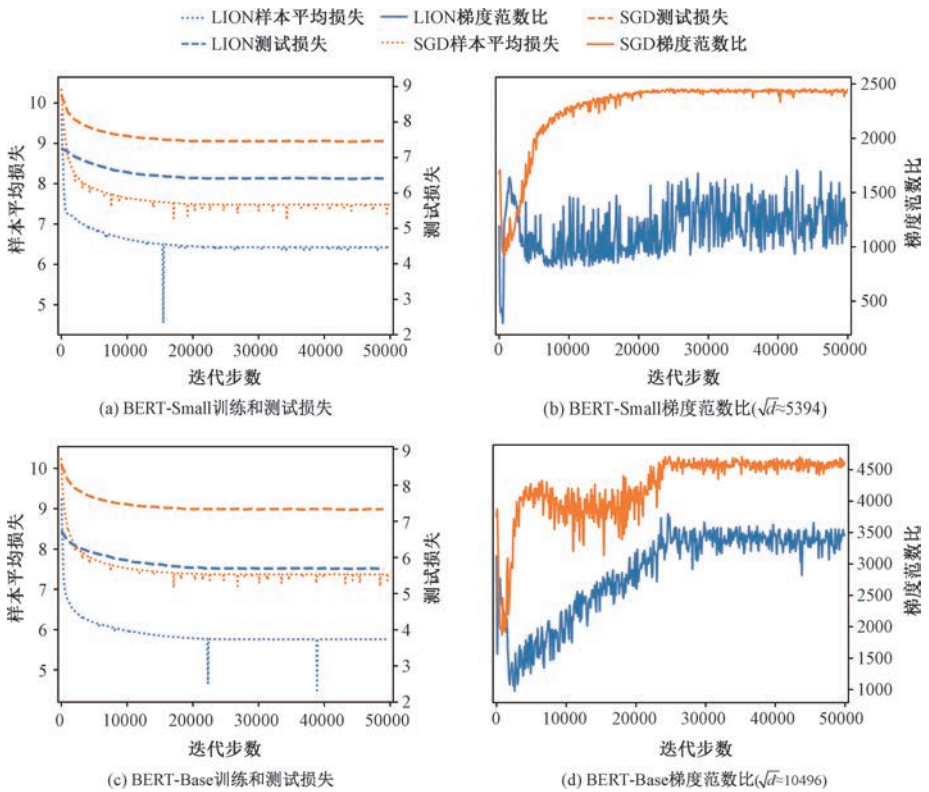


图 2　BERT-Small 和 BERT-Base 模型[66]在 OpenWebText 数据集[65]上的实验结果(子图(a)和(c)展示了训练过程中的样本平均损失和测试损失,子图(b)和(d)展示了梯度范数比的变化过程。)

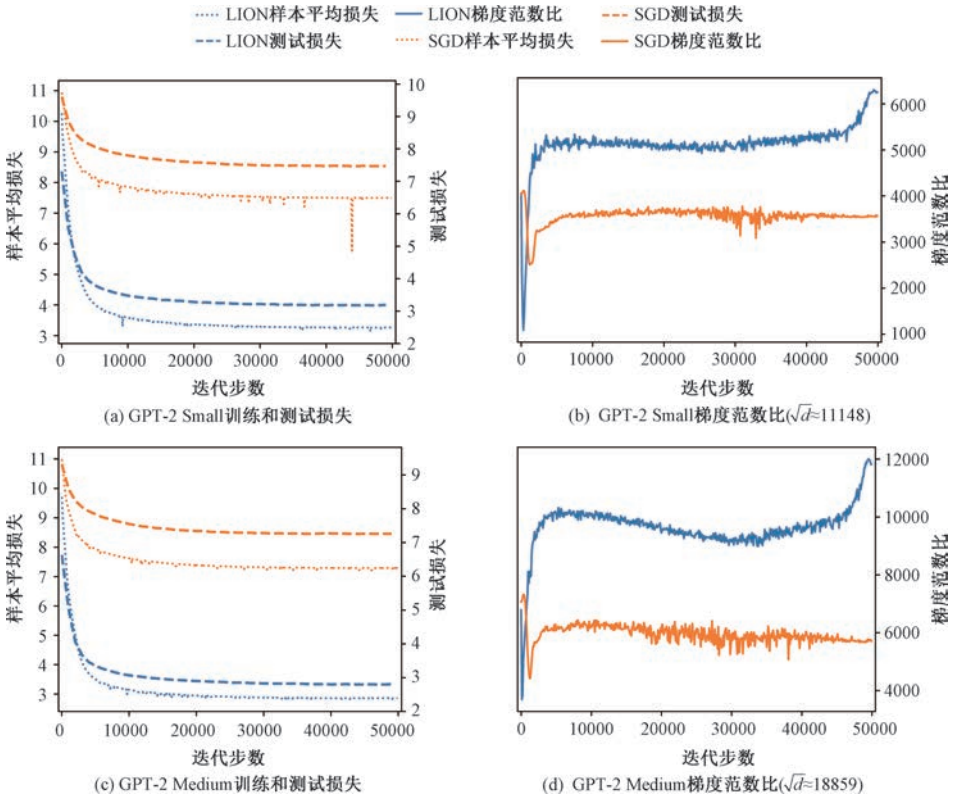


图 3　GPT-2 Small 和 GPT-2 Medium 模型[67]在 OpenWebText 数据集[65]上的实验结果(子图(a)和(c)展示了训练过程中的样本平均损失和测试损失,子图(b)和(d)展示了梯度范数比的变化过程。)

容易观察到，在这些任务上 LION 优化器的训练和测试损失均显著低于 SGD。相关领域的许多研究已经解释了这种现象，即 SGD 在训练 Transformer 类模型时的性能显著劣于 AdamW[23-25]，而 LION 在这些任务上通常具有与 AdamW 相似或更强的性能[26]。另一方面，LION 优化器在这些模型上的训练过程中梯度范数比仍满足 $r=\Theta(\sqrt{d})$，从而验证了该关系在语言模态的深度学习任务上同样成立。

另外，在 GPT-2 模型训练的各个阶段，我们还评估了这些中间模型在 WikiText-103 数据集[68]上的零样本(Zero-shot)性能。结果显示，使用 LION 优化器仅需进行少量迭代即可超越使用 SGD 训练结束时的性能，达到了更低的困惑度(Perplexity)。具体细节请参阅附录 A.3。

本节在视觉和语言模态的多种深度学习任务已充分证明，LION 优化器在实际应用中不仅体现出更优的性能，而且具有良好的稳定性。该优化器目前已广泛应用于 Transformer 等深度网络模型的训练[26]。

总结来说，尽管 LION 和 SGD 在理论上关于迭代步数 K 均已匹配随机梯度类算法的下界，我们在多种任务上的实验结果充分证明了 LION 相对于 SGD 的明显优势，在所有任务上均实现了更低的损失与更强的性能。另外，这些任务的训练过程中 LION 的梯度范数比 r 总是保持在 $\Theta(\sqrt{d})$ 的量级，在本文基于梯度 ℓ_1 范数与下界基于梯度 ℓ_2 范数的收敛速度的对比过程中，可以在实证意义上约去 $\mathcal{O}(\sqrt{d}K^{-1/4})$ 中的 $\sqrt{d}$ 项，证明了本文的收敛速度关于问题维度 d 同样可以匹配随机梯度类算法的理论下界。本文完成的多组实验提供了 LION 优化器理论分析结果的有效支持。

6 结论与展望

本文对 LION 优化器进行了全面的收敛速度分析，详细阐明了该算法的理论及实际优势。本文证明，LION 在梯度 ℓ_1 范数的度量下，优化一种特定的带约束问题和广义的无约束问题的收敛速度均为 $\mathcal{O}(\sqrt{d}K^{-1/4})$。该速度精确匹配了非凸优化问题中随机梯度类算法的关于迭代步数 K 的理论下界，且实现了当前研究工作中关于问题维度 d 的最优依赖。另外，通过在视觉和语言任务上的广泛实验，我们不仅证明了 LION 比 SGD 在实际应用中具有更强的性能，而且说明了 LION 优化过程中梯度的 ℓ_1/ℓ_2 范数比始终保持在 $\Theta(\sqrt{d})$ 的量级，从而在经验意义上说明我们的收敛速度关于问题维度 d 与理论下界同样匹配。

未来的研究可以进一步探讨 LION 优化器在更广泛应用场景中的适应性。例如，现有工作尚未严格建立随机梯度类算法在梯度 ℓ_1 范数度量下关于问题维度 d 的理论下界，深入研究这一问题并给出明确的理论界限，有助于进一步完善非凸优化算法的收敛理论，并为 LION 的最优性提供更坚实的理论支撑。其次，在分布式优化和联邦学习环境下，结合误差补偿机制或梯度压缩技术，探讨 LION 在分布式场景下的应用潜力是一个值得研究的问题。最后，在当今超大规模的 Transformer 类大模型的应用背景下，业界广泛采用浮点数量化和混合精度技术来提高计算效率。然而，这些技术在加速训练的同时也引入了一定的数值误差。因此，如何在考虑量化误差和精度损失的情况下严格分析 LION 的收敛速度，仍是一个值得进一步研究的问题。相信这些研究会在非凸优化领域和深度学习领域带来更多的创新和突破。

附 录

A 补充实验

A.1 LION 收敛速度的实证分析

为充分说明本文所证明收敛速度的实际意义，本节首先以简单的非凸优化问题为载体，说明 LION 在符合理论假设的实验设定下解决盒约束问题(5)和无约束问题(2)的收敛速度均能较好吻合理论结果，然后在实际的深度学习任务中，在说明 LION 同样能够匹配理论收敛速度的同时，通过详细的数值分析展示了 LION 相对于 SGD 的优越性。

A.1.1 简单非凸优化问题的结果

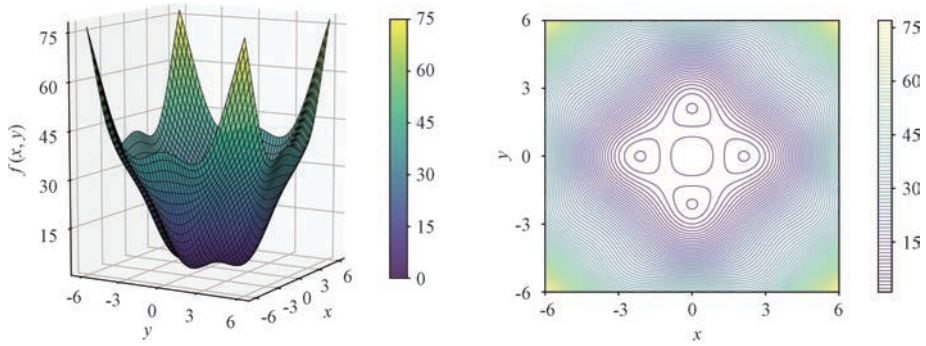
本文第 3 节已证明，LION 在解决带约束优化问题(5)的过程中收敛至 KKT 点的速度为 $\mathcal{O}(\sqrt{d}K^{-1/4})$，如式(9)所示；在解决无约束优化问题时同样具有上述收敛速度，如式(13)所示。本节构造了一个简单的二元目标函数 f，在确保 f 符合本文理论假设的前提下，有效验证了带约束情形下不等式(9)的左侧 $\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}[\lambda\langle\nabla f(\boldsymbol{\theta}^k),\boldsymbol{\theta}^k\rangle+\|\nabla f(\boldsymbol{\theta}^k)\|_1]$ 和无约束情形下不等式(13)的左侧

$\frac{1}{2K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}^k)\|_1\right]$ 均能较好地匹配不等式右侧的理论收敛速度值。

具体而言,定义目标函数为①

$$f(\boldsymbol{\theta})=(\boldsymbol{\theta}_1)^2+(\boldsymbol{\theta}_2)^2+5\cos\boldsymbol{\theta}_1\cos\boldsymbol{\theta}_2,$$

其中 $\boldsymbol{\theta}=[\boldsymbol{\theta}_1,\boldsymbol{\theta}_2]^{\mathrm{T}}\in\mathbb{R}^2$。容易验证,$f$ 为非凸函数,且满足假设 1 和 4。f 的 Lipschitz 常数 $L=7$,全局最小值点为 $(0,\pm 2.125)$ 和 $(\pm 2.125,0)$,对应最小值为 $f^*=1.884$(均精确到小数点后三位)。f 的三维曲面图(3D Surface Plot)和等高线图(Contour Plot)分别如附图 1(a)和 1(b)所示。



(a)f 的三维曲面图　　(b)f 的等高线图

附图 1　二元目标函数的图像特征

对任意点 $\boldsymbol{\theta}$,目标函数 f 的真实梯度 $\nabla f(\boldsymbol{\theta})$ 可直接求出。为模仿随机梯度,我们定义 $\boldsymbol{g}^k=\nabla f(\boldsymbol{\theta}^k)+\boldsymbol{\varepsilon}$,其中 $\boldsymbol{\varepsilon}$ 的各分量为独立同分布的标准高斯噪声,即 $\boldsymbol{\varepsilon}=[\varepsilon_1,\varepsilon_2]^{\mathrm{T}},\varepsilon_1,\varepsilon_2\sim N(0,1)$。易知 $\boldsymbol{g}^k$ 满足假设 2 和 3,其中假设 3 的 $\sigma^2=2$。

取权重衰减系数 $\lambda=0.1$,则盒约束问题的约束条件为 $\|\boldsymbol{\theta}\|_\infty\leqslant 10$。针对该问题,我们将 $\boldsymbol{\theta}^1$ 初始化在约束边界上,令 $\boldsymbol{\theta}_1^1=10$,$\boldsymbol{\theta}_2^1$ 为 $[-10,10]$ 之间的随机数。由于 f 关于 $\boldsymbol{\theta}_1^1$ 和 $\boldsymbol{\theta}_2^1$ 具有对称性,故以此原则选取不同的初始化点 $\boldsymbol{\theta}^1$ 时可全面衡量 LION 在盒约束问题下的收敛速度。

为降低实验过程中初始化点 $\boldsymbol{\theta}^1$ 和随机梯度 $\boldsymbol{g}^k$ 带来的随机因素,我们选取 5 个不同的初始化点展开实验,并针对每个点分别运行 5 次重复的实验过程,共计 25 组实验。由于目标函数 f 的结构较为简单,我们可以算出式(9)右侧 L、σ 和 f^* 等各常数的精确值。设置学习率 $\eta=0.01$,迭代步数 $K=2000$,记录每组实验迭代过程中不等式(9)左侧和右侧的值并取其平均,我们将结果展示在附图 2(a)中。

附图 2(a)中的实线为 LION 优化过程中式(9)左侧的实际收敛速度,虚线反映了式(9)右侧的理论值。可以看出,本文证明的在带约束问题中的理论结果与实际优化过程较为吻合,进一步突出了推论 3 的实际指导意义。

值得指出的是,LION 在实际应用中的收敛速度并不会完全吻合其理论值。因为本文结论衡量的是 LION 对任意一个满足假设的非凸目标函数在最坏情况下的收敛速度,仅当第 4 节的证明过程中所有不等式的相等条件同时成立时才能完全吻合。实际应用中 LION 优化器的收敛速度会略好于该值。

最后,我们尝试了不同的学习率 η 和权重衰减系数 λ,发现 LION 优化器在这些参数的不同取值下均能得到与附图 2(a)相同的结论。

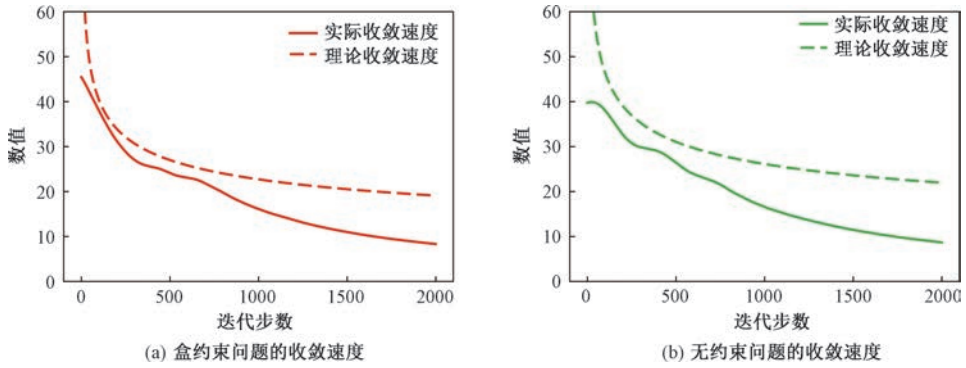
对于无约束问题,由于 f 同样满足定理 4 的所有假设,我们仍使用该函数进行研究。不同于盒约束问题的情形,在无约束问题中我们将初始化点的两个分量 $\boldsymbol{\theta}_1^1$ 和 $\boldsymbol{\theta}_2^1$ 均设置为 $[-10,10]$ 之间的随机数,并在迭代过程中衡量式(13)左右两侧的值。同样地,我们随机选取了 5 个不同的初始化点,对每个点分别进行 5 次重复实验,并汇报所有实验的平均结果。

附图 2(b)展示了 LION 在解决无约束问题的迭代过程中理论与实际收敛速度的比较。可以看出,LION 的实际收敛速度仍能较好地匹配推论 5 中给出的理论值。进一步实验发现,使用不同的学习率和权重衰减系数均不会影响附图 2(b)中的主要结论。

A.1.2　实际深度学习任务的结果

为详细验证 LION 优化器在实际深度学习任务

① 根据 2.1 节中的符号约定,我们将参数 $\boldsymbol{\theta}$ 的第 1 个分量的平方记为 $(\boldsymbol{\theta}_1)^2$,以便与 $\boldsymbol{\theta}$ 在第 2 步时的第 1 个分量 $\boldsymbol{\theta}_1^2$ 相区分。



附图 2 LION 优化器在简单盒约束和无约束非凸优化问题中的收敛速度图(我们构造了一个简单的二元目标函数 f,使用 LION 算法多次进行实验,并汇报所有实验的平均值。结果显示,LION 解决两种问题的收敛速度均能较好地匹配本文所证明的理论值。)

中的性能,充分体现本文理论贡献的价值,本节对 LION 和 SGD 的收敛速度进行了细致的对比,不仅说明了 LION 收敛速度与理论结果的适应性,而且详细地阐述了 LION 在实际应用中相比于 SGD 更强的原因。

本文 2.2 节的相关工作已提到,SGD 的收敛速度为

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}^k)\|_2\right]\leqslant \underbrace{c'_{\mathrm{SGD}}\cdot\sqrt[4]{\sigma^2 L(f(\boldsymbol{\theta}^1)-f^*)}}_{c_{\mathrm{SGD}}}\cdot K^{-1/4} \quad (\text{附 } 1)$$

其中,c'_{SGD} 为常数。在推论 5 中已证明,LION 优化器解决无约束问题的收敛速度为

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}^k)\|_1\right]\leqslant \underbrace{c'_{\mathrm{LION}}\cdot\sqrt[4]{\sigma^2 L(f(\boldsymbol{\theta}^1)-f^*)}}_{c_{\mathrm{LION}}}\cdot\sqrt{d}K^{-1/4} \quad (\text{附 } 2)$$

其中,c'_{LION} 为常数。虽然二者关于迭代步数 K 均能匹配随机梯度类算法的理论收敛速度下界 $\mathcal{O}(K^{-1/4})$,但在实际应用中,其中的常数差异会对收敛速度产生显著影响。

本节以在 CIFAR-100[61] 数据集上训练 ResNet18[63] 模型为例,说明在实际深度学习任务中梯度范数期望的平均值(即不等式(附 1)和(附 2)的左侧)关于迭代步数 K 均能较好地匹配 $\mathcal{O}(K^{-1/4})$ 的下界,以强调本文贡献的实际指导意义。另外,通过说明在深度学习实验中一般有 $c'_{\mathrm{LION}}<c'_{\mathrm{SGD}}$ 成立,从而解释了 LION 相比于同样匹配下界的 SGD 在实际应用中收敛速度更快的根本原因。

在 A.1.1 节中,我们计算出了目标函数的全局最小值 f^*、Lipschitz 常数 L 以及随机梯度的噪声上界 σ^2。然而,由于深度网络内含的复杂嵌套结构,在实际任务中我们难以得到这些常数的值。观察到式(附 1)和(附 2)的右侧关于这些常数的乘法因子同为 $\sqrt[4]{\sigma^2 L(f(\boldsymbol{\theta}^1)-f^*)}$,因此我们可以通过比较式(附 1)中 $K^{-1/4}$ 的系数 c_{SGD} 和式(附 2)中 $\sqrt{d}K^{-1/4}$ 的系数 c_{LION},以达到比较 c'_{SGD} 和 c'_{LION} 的目的。

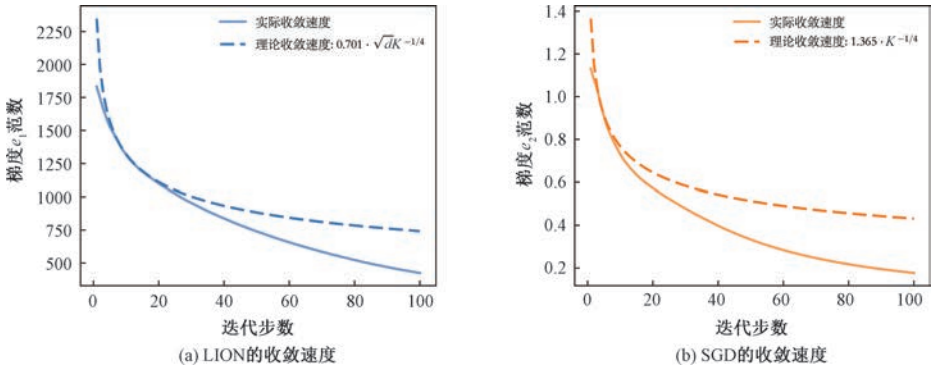
我们使用与正文第 5 节中完全相同的实验设定,分别使用 LION 和 SGD 优化器完成深度学习任务的训练。在 LION 的训练过程中,我们记录式(附 2)左侧梯度 ℓ_1 范数的平均值;在 SGD 的训练过程中,我们记录式(附 1)左侧梯度 ℓ_2 范数的平均值。为降低实验的随机性,针对每个优化器我们都重复进行 5 次训练,并取所有实验结果的平均值。LION 和 SGD 的实际收敛速度分别如附图 3(a)和附图 3(b)中实线所示。

另外,我们根据实际收敛速度完成了对常数 c_{LION} 和 c_{SGD} 的估计。在保证不等式对任意 K 成立的前提下,我们计算出满足条件的各常数的最小值,以得到对 c_{LION} 和 c_{SGD} 最紧密的估计。经过计算得知,$c_{\mathrm{LION}}=0.701$,$c_{\mathrm{SGD}}=1.365$,从而说明了 LION 在实际深度学习任务中收敛得比同样匹配下界的 SGD 更快。

在得到 c_{LION} 和 c_{SGD} 的值之后,我们将对应的理论收敛速度曲线分别体现在了附图 3(a)和附图 3(b)中的虚线上。可以看到,LION 在实际深度学习任务中仍然能够较好地体现 $\mathcal{O}(\sqrt{d}K^{-1/4})$ 的收敛速度,充分佐证了本文贡献的实际价值。

最后,我们再次指出,对一个优化器的理论分析结果仅代表其在最坏情况下的收敛速度,实际应用

中的收敛速度会略好于该值。不过，附图 3 的结果也能说明最坏情况下收敛速度的实际指导意义。



附图 3　ResNet18 模型使用 CIFAR-100 数据集训练时 LION 和 SGD 优化器迭代过程中收敛速度的比较（结果显示，LION 优化器在梯度 ℓ_1 范数下收敛速度为 $0.701 \cdot \sqrt{d}K^{-1/4}$，而 SGD 优化器在梯度 ℓ_2 范数下收敛速度为 $1.365 \cdot K^{-1/4}$。因此，LION 在实际应用中收敛得比 SGD 更快。）

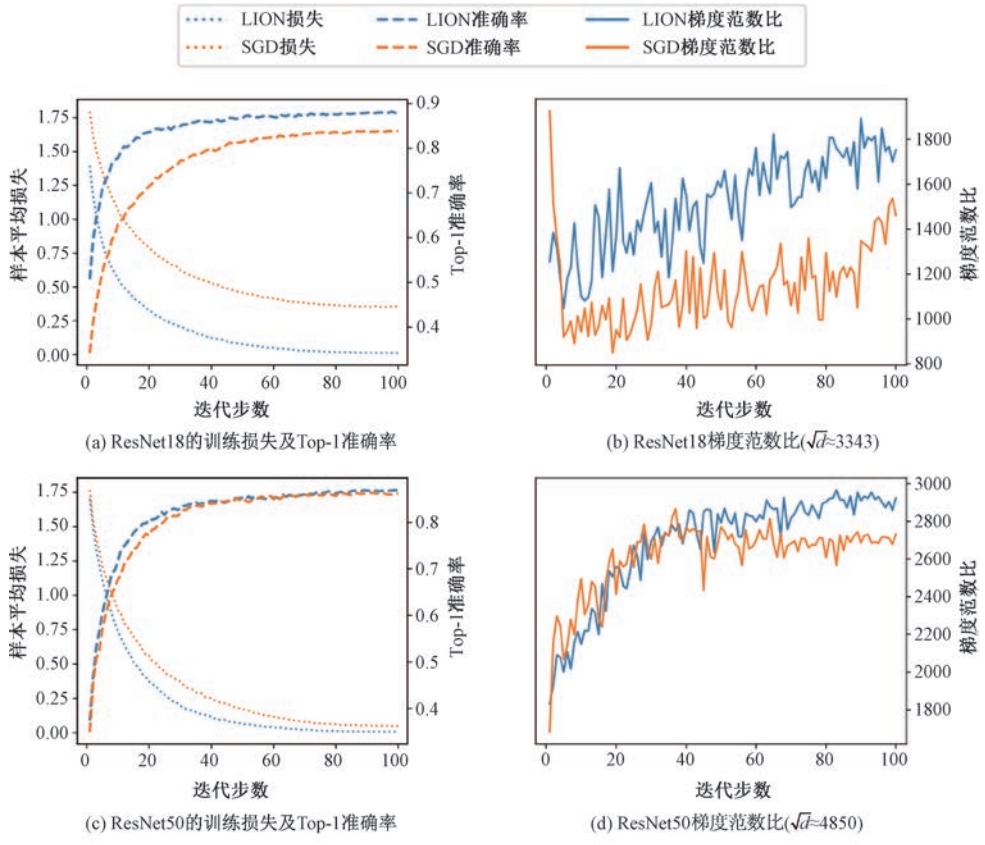
A.2 视觉任务：CIFAR-10 和 ImageNet-1K 数据集的评测结果

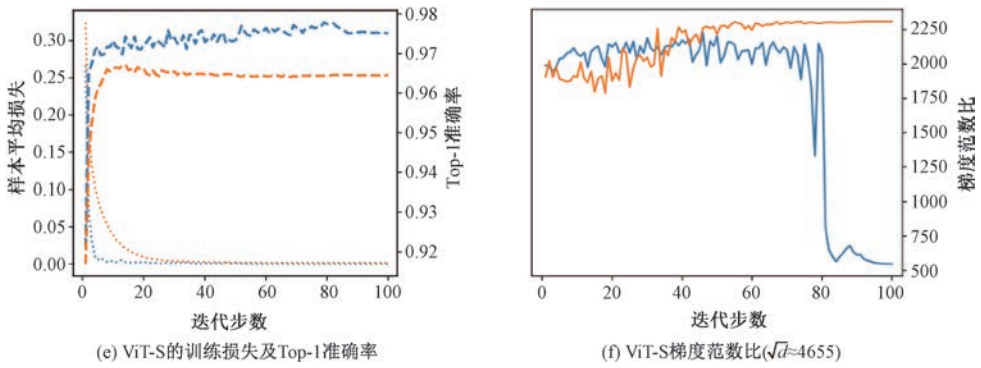
如第 5 节所述，我们在 CIFAR-10[61] 和 ImageNet-1K[62] 数据集上进行了补充实验，结果分别如附图 4 和附图 5 所示。这些结果进一步印证了我们的结论。

A.3 语言任务：WikiText-103 上的零样本评测结果

在本节中，我们提取了 GPT-2 模型预训练过程中在第 10000、20000、30000、40000 和 50000 步处的检查点（Checkpoint），并评测这些中间产物在 WikiText-103 数据集[68] 上的零样本性能。我们以困惑度作为性能的衡量指标。

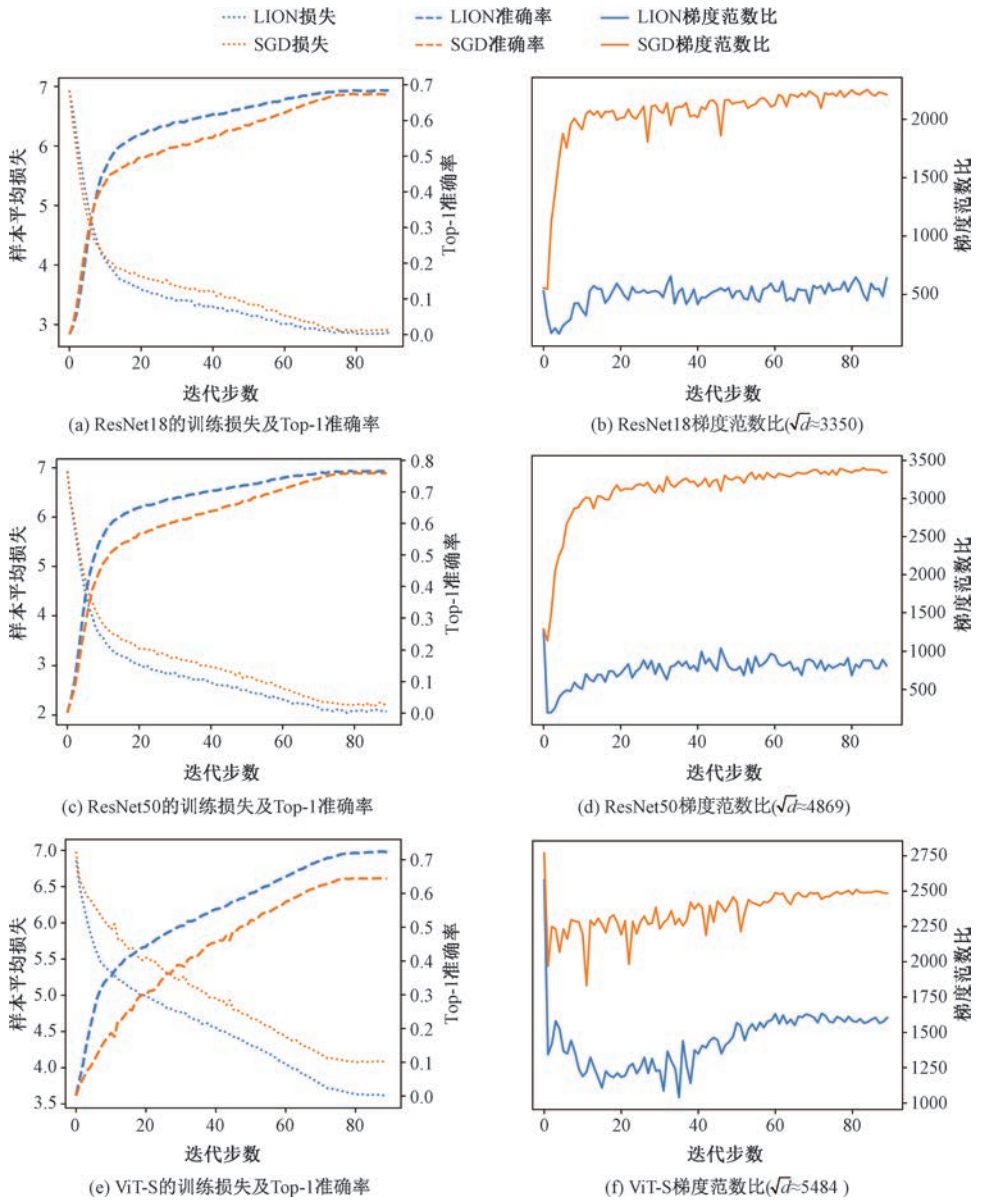
我们将实验结果整理在附表 1 中。为保持数量级的一致性，我们采取了常用的对困惑度取对数的缩放操作。结果显示，使用 LION 优化算法训练的 GPT-2 模型在训练的一开始就达到了比使用 SGD 训练到最终阶段更低的困惑度，进一步证明了 LION 在实践中能够实现更快的收敛速度和更优的模型性能。





(e) ViT-S的训练损失及Top-1准确率　(f) ViT-S梯度范数比($\sqrt{d}\approx4655$)

附图 4　ResNet18[63]、ResNet50[63]和 ViT-S[64]模型在 CIFAR-10 数据集[61]上的实验结果



(a) ResNet18的训练损失及Top-1准确率　(b) ResNet18梯度范数比($\sqrt{d}\approx3350$)

(c) ResNet50的训练损失及Top-1准确率　(d) ResNet50梯度范数比($\sqrt{d}\approx4869$)

(e) ViT-S的训练损失及Top-1准确率　(f) ViT-S梯度范数比($\sqrt{d}\approx5484$)

附图 5　ResNet18[63]、ResNet50[63]和 ViT-S[64]模型在 ImageNet-1K 数据集[62]上的实验结果

附表 1　LION 和 SGD 优化器在 GPT-2 Small 和 Medium 架构上的表现

模型架构	优化器	迭代步数				
		10k	20k	30k	40k	50k
GPT-2 Small	SGD	8.202	8.005	7.922	7.892	7.887
	LION	**3.659**	**3.409**	**3.326**	**3.293**	**3.284**
GPT-2 Medium	SGD	8.029	7.807	7.737	7.715	7.712
	LION	**3.221**	**3.004**	**2.918**	**2.882**	**2.876**

注：我们在模型预训练过程中的第 10000、20000、30000、40000 和 50000 步处的检查点进行评测，并在本表中汇报困惑度的对数值，该指标越低，证明效果越好。LION 优化器在训练的一开始就超越了 SGD 的最终结果，展现了更快的收敛速度和更佳的性能。

B　实验细节

为方便复现，本节介绍正文中第 5 节所完成实验的细节。关于数据集的划分，本文使用的所有数据集均包含官方统一的训练集和测试集划分，可以保证在重复实验时所使用的训练和测试数据完全相同。

(1) CIFAR-10 和 CIFAR-100 数据集的训练。鉴于最佳参数可能因数据集和模型而异，我们针对不同的学习率开展了网格式搜索，具体的搜索范围为[3e-3，1e-3，3e-4，1e-4，3e-5，1e-5，3e-6，1e-6]。该范围涵盖了绝大多数深度学习任务中使用的学习率设定，通过搜索的方式可以为每组实验找出最优的学习率。具体学习率的搜索结果如附表 2 所示。对于动量系数 β_1 和 β_2，我们采用 LION 优化器默认的参数设置。整个训练包含 100 个轮次，在训练过程中，我们采用了经典的余弦退火学习率调度策略。我们将批大小设置为 64，为避免过拟合现象，我们使用了权重衰减技术，并将系数设置为 0.1。考虑到 ViT 模型最初是为处理 ImageNet 数据集的 224×224 大小的图像设计的，而 CIFAR 数据集的图片大小只有 32×32，我们在此应用了一种常见的数据预处理方法，即先把 CIFAR 数据集中的图片缩放为 224×224，然后基于 ImageNet 的预训练模型进行微调。结果显示，LION 无论是在 ResNet 的预训练任务还是 ViT 的微调任务上都实现了超过 SGD 的性能。本组实验均在单个 NVIDIA A6000 GPU 上完成。

附表 2　CIFAR-10 和 CIFAR-100 数据集不同模型架构下的学习率搜索结果

模型架构	CIFAR-10	CIFAR-100
ResNet18	3e-6/1e-6	3e-5/3e-4
ResNet50	3e-6/3e-6	3e-5/1e-4
ViT-S	1e-6/1e-6	1e-6/1e-6

注：在每组设定中，斜杠“/”前表示 LION 使用的学习率，斜杠后表示 SGD 使用的学习率。

(2) ImageNet 数据集的训练。ImageNet 是广泛应用于评测深度模型图像分类能力的知名数据集，其中最常用的是 ImageNet-1K 版本。ImageNet-1K 的数据包含 1000 个类别，涵盖了日常生活中常见的各种物体。学界对于该数据集的训练模式已有公认的设定[63]，本文同样沿用此设定作为标准训练配置。在该设定下，整个训练过程包含 90 个轮次，其中前 10 个轮次是学习率热身(Warm-up)阶段，中间 70 个轮次为余弦退火阶段，最后 10 个轮次为冷却(Cooldown)阶段。读者可参考文章[63]来获取各参数的详细数值。对于动量系数 β_1 和 β_2，我们采用 LION 优化器默认的参数设置。本组实验在 8 个 NVIDIA A6000 GPU 上以分布式训练的设定完成。

(3) 语言任务的训练。我们使用英伟达公司开发的知名大语言模型训练框架 Megatron-LM[69]，在 OpenWebText[65] 数据集上训练和评估模型。由于计算资源有限，我们无法精确复刻工业级模型训练过程中数千个 GPU 的超大规模运算配置。读者可参考文章[69]来获取各参数的详细数值。不过，本文实验的绝大多数配置都沿用了 Megatron-LM 的官方设定，只对推荐配置做了少量修改。具体来说，我们将 SGD 的学习率设置为 1e-4，LION 的学习率设置为 1e-5，全局批大小设置为 640，在 100 个批次上累积训练集的平均样本损失和梯度信息。对于动量系数 β_1 和 β_2，如本文第 3 节中所述，在大语言模型训练中通常采用 $(\beta_1,\beta_2)=(0.9,0.95)$，我们使用了这一参数设置。整个训练过程包含 50000 步，在 16 个 NVIDIA A100 GPU 上完成。

参　考　文　献

[1] Shai Shalev-Shwartz, Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms. New York, USA: Cambridge University Press, 2014

[2] Trevor Hastie, Robert Tibshirani, Jerome Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York, USA: Springer, 2009

[3] Lumír Pečený, Pavol Meško, Rudolf Kampf, Jozef Gašparík. Optimisation in transport and logistic processes. Transportation Research Procedia, 2020, 44:15-22

[4] Sai-Ho Chung. Applications of smart technologies in logistics and transport: A review. Transportation Research Part E: Logistics and Transportation Review, 2021,153:102455

[5] Duan Li, Wan-Lung Ng. Optimal dynamic portfolio selection: Multiperiod mean-variance formulation. Mathematical Fi-

nance, 2000,10(3):387-406

[6] Hans Follmer, Alexander Schied. Stochastic finance: an introduction in discrete time. Berlin,Germany: Walter de Gruyter, 2011

[7] Frank J Fabozzi, Petter N Kolm, Dessislava A Pachamanova, Sergio M Focardi. Robust portfolio optimization and management. Hoboken, USA: John Wiley & Sons, 2007

[8] Sebastien Bubeck et al. Convex optimization: Algorithms and complexity. Foundations and Trends ® in Machine Learning, 2015,8(3—4):231-357

[9] Stephen Boyd, Lieven Vandenberghe. Convex optimization. New York, USA: Cambridge University Press, 2004

[10] Prateek Jain, Purushottam Kar, et al. Non-convex optimization for machine learning. Foundations and Trends ® in Machine Learning, 2017. 10(3—4):142-363

[11] Jorge Nocedal, Stephen J Wright. Numerical optimization. New York, USA: Springer, 1999

[12] Jan Snyman. Practical mathematical optimization, New York, USA: Springer, 2005

[13] Quoc V Le, Jiquan Ngiam, Adam Coates, Abhik Lahiri, Bobby Prochnow, Andrew Y Ng. On optimization methods for deep learning//Proceedings of the International Conference on Machine Learning. Washington, USA, 2011: 265-272

[14] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 2015, 521(7553):436-444

[15] Huan Li, Yiming Dong, and Zhouchen Lin. On the $O(\sqrt{d}T^{-1/4})$ convergence rate of RMSProp and its momentum extension measured by ℓ_1 norm: Better dependence on the dimension. arXiv preprint arXiv:2402. 00389, 2024

[16] John Duchi, Elad Hazan, Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 2011, 12(61): 2121-2159

[17] Diederik P Kingma, Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412. 6980, 2014

[18] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics, 1964, 4(5):1-17

[19] Tijmen Tieleman, Geoffrey Hinton. RMSProp: Divide the gradient by a running average of its recent magnitude. COURSERA Neural Networks for Machine Learning, 2012

[20] Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, Shuicheng Yan. Adan: Adaptive Nesterov momentum algorithm for faster optimizing deep models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(12): 9508-9520

[21] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, Blake Woodworth. Lower bounds for non-convex stochastic optimization. Mathematical Programming, 2023, 199(1):165-214

[22] Léon Bottou, Frank E Curtis, Jorge Nocedal. Optimization methods for large-scale machine learning. SIAM review, 2018, 60(2):223-311

[23] Yushun Zhang, Congliang Chen, Tian Ding, Ziniu Li, Ruoyu Sun, Zhi-Quan Luo. Why Transformers need Adam: A Hessian perspective//Advances in Neural Information Processing Systems. Vancouver, Canada, 2024: 131786-131823

[24] Frederik Kunstner, Jacques Chen, Jonathan Wilder Lavington, Mark Schmidt. Noise is not the main factor behind the gap between SGD and Adam on Transformers, but sign descent might be//Proceedings of the International Conference on Learning Representations. Kigali, Rwanda, 2023

[25] Yan Pan, Yuanzhi Li. Toward understanding why Adam converges faster than SGD for Transformers. arXiv preprint arXiv:2306. 00204, 2023

[26] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms//Advances in Neural Information Processing Systems. New Orleans, USA, 2023: 49205-49233

[27] Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, Wotao Yin. Learning to optimize: A primer and a benchmark. Journal of Machine Learning Research, 2022, 23(189):1-59

[28] Bohan Wang, Huishuai Zhang, Zhiming Ma, Wei Chen. Convergence of AdaGrad for non-convex objectives: Simple proofs and relaxed assumptions//Proceedings of the Thirty Sixth Annual Conference on Learning Theory. Bangalore, India, 2023: 161-190

[29] Zijian Liu, Ta Duy Nguyen, Thien Hang Nguyen, Alina Ene, Huy Nguyen. High probability convergence of stochastic gradient methods//Proceedings of the International Conference on Machine Learning. Honolulu, USA, 2023: 21884-21914

[30] Yusu Hong and Junhong Lin. Revisiting convergence of AdaGrad with relaxed assumptions//Proceedings of the Uncertainty in Artificial Intelligence. Barcelona, Spain, 2024: 1727-1750

[31] Naichen Shi, Dawei Li. RMSProp converges with proper hyperparameter//Proceedings of the International Conference on Learning Representations. Virtual, 2021

[32] Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, Wei Liu. A sufficient condition for convergences of Adam and RMSProp//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, USA, 2019: 11127-11135

[33] Yushun Zhang, Congliang Chen, Naichen Shi, Ruoyu Sun, Zhi-Quan Luo. Adam can converge without any modification on update rules//Advances in Neural Information Processing Systems. New Orleans, USA, 2022: 28386-28399

[34] Bohan Wang, Jingwen Fu, Huishuai Zhang, Nanning Zheng, Wei Chen. Closing the gap between the upper bound and lower bound of Adam's iteration complexity//Advances in Neural Information Processing Systems. New Orleans, USA,

2023: 39006-39032

[35] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, Animashree Anandkumar. SignSGD: Compressed optimisation for non-convex problems//Proceedings of the International Conference on Machine Learning. Stockholm, Sweden, 2018: 560-569

[36] Tao Sun, Qingsong Wang, Dongsheng Li, Bao Wang. Momentum ensures convergence of SignSGD under weaker assumptions//Proceedings of the International Conference on Machine Learning. Honolulu, USA, 2023: 33077-33099

[37] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, Martin Jaggi. Error feedback fixes SignSGD and other gradient compression schemes//Proceedings of the International Conference on Machine Learning. Long Beach, USA, 2019: 3252-3261

[38] Mher Safaryan, Peter Richtárik. Stochastic sign descent methods: New algorithms and better theory//Proceedings of the International Conference on Machine Learning. Virtual, 2021: 9224-9234

[39] Rachel Ward, Xiaoxia Wu, and Léon Bottou. AdaGrad stepsizes: Sharp convergence over nonconvex landscapes. Journal of Machine Learning Research, 2020, 21(219):1-30

[40] Bo Liu, Lemeng Wu, Lizhang Chen, Kaizhao Liang, Jiaxu Zhu, Chen Liang, Raghuraman Krishnamoorthi, and Qiang Liu. Communication efficient distributed training with distributed Lion. arXiv preprint arXiv:2404.00438, 2024

[41] Michael Crawshaw, Mingrui Liu, Francesco Orabona, Wei Zhang, Zhenxun Zhuang. Robustness to unbounded smoothness of generalized SignSGD//Advances in Neural Information Processing Systems. New Orleans, USA, 2022: 9955-9968

[42] Nachuan Xiao, Xiaoyin Hu, Kim-Chuan Toh. Convergence guarantees for stochastic subgradient methods in nonsmooth nonconvex optimization. arXiv preprint arXiv:2307.10053, 2023

[43] Lizhang Chen, Bo Liu, Kaizhao Liang, Qiang Liu. Lion secretly solves constrained optimization: As Lyapunov predicts. arXiv preprint arXiv:2310.05898, 2023

[44] Jan Kukacka, Vladimir Golkov, Daniel Cremers. Regularization for deep learning: A taxonomy. arXiv preprint arXiv: 1710.10686, 2017

[45] Yingjie Tian, Yuqi Zhang. A comprehensive survey on regularization strategies in machine learning. Information Fusion, 2022, 80: 146-166

[46] Lukas Balles, Fabian Pedregosa, and Nicolas Le Roux. The geometry of sign gradient descent. arXiv preprint arXiv: 2002.08056, 2020

[47] Ilya Loshchilov, Frank Hutter. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017

[48] Sashank J Reddi, Satyen Kale, Sanjiv Kumar. On the convergence of Adam and beyond. arXiv preprint arXiv: 1904.09237, 2019

[49] Bohan Wang, Huishuai Zhang, Qi Meng, Ruoyu Sun, Zhi-Ming Ma, Wei Chen. On the convergence of Adam under non-uniform smoothness: Separability from SGDM and beyond. arXiv preprint arXiv:2403.15146, 2024

[50] Shuo Xie, Zhiyuan Li. Implicit bias of AdamW: ℓ_∞ norm constrained optimization. arXiv preprint arXiv: 2404.04454, 2024

[51] Budhayash Gautam. Energy minimization. Homology Molecular Modeling-Perspectives and Applications. London, UK: IntechOpen, 2020

[52] Mila Nikolova. Energy minimization methods. Handbook of Mathematical Methods in Imaging. New York, USA: Springer, 2015: 157-204

[53] David P Williamson. Network flow algorithms. New York, USA: Cambridge University Press, 2019

[54] Martin Treiber, Arne Kesting. Traffic flow dynamics. Traffic Flow Dynamics: Data, Models and Simulation. Berlin, Germany: Springer-Verlag, 2013

[55] Enrico Angelelli, Valentina Morandi, Martin Savelsbergh, Maria Grazia Speranza. System optimal routing of traffic flows with user constraints using linear programming. European Journal of Operational Research, 2021, 293(3):863-879

[56] Jiawei Ren, Mingyuan Zhang, Cunjun Yu, Ziwei Liu. Balanced MSE for imbalanced visual regression//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans, USA, 2022: 7926-7935

[57] Aryan Jadon, Avinash Patil, Shruti Jadon. A comprehensive survey of regression-based loss functions for time series forecasting//Proceedings of the International Conference on Data Management, Analytics & Innovation. Vellore, India, 2024: 117-147

[58] Yu Takagi, Shinji Nishimoto. High-resolution image reconstruction with latent diffusion models from human brain activity//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Vancouver, Canada, 2023: 14453-14463

[59] Titas Anciukevicius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J Mitra, Paul Guerrero. RenderDiffusion: Image diffusion for 3D reconstruction, inpainting and generation//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Vancouver, Canada, 2023: 12608-12618

[60] Yutong Xie, Quanzheng Li. Measurement-conditioned denoising diffusion probabilistic model for under-sampled medical image reconstruction//Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention. Singapore, 2022: 655-664

[61] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical Report, University of Toronto, 2009

[62] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg,

Li Fei-Fei. ImageNet large scale visual recognition challenge. International Journal of Computer Vision, 2015, 115(3): 211-252

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep residual learning for image recognition//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, USA, 2016: 770-778

[64] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv: 2010.11929, 2020

[65] Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, Stefanie Tellex. OpenWebText corpus. http://Skylion007. github. io/OpenWebTextCorpus, 2019

[66] Jacob Devlin. BERT: Pre-training of deep bidirectional Transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018

[67] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019

[68] Stephen Merity, Caiming Xiong, James Bradbury, Richard Socher. Pointer sentinel mixture models. arXiv preprint arXiv:1609.07843, 2016

[69] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, Bryan Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism. arXiv preprint arXiv:1909.08053, 2019

DONG Yi-Ming, Ph. D. candidate. His research interests include deep learning and optimization algorithms.

LI Huan, Ph. D., associate professor. His research interests include machine learning and numerical optimization.

LIN Zhou-Chen, Ph. D., professor, Ph. D. supervisor. His research interests include machine learning and numerical optimization.

Background

The research presented in this paper is focused on the field of deep learning, particularly in the context of large-scale nonconvex optimization problems. In recent years, significant progress has been made internationally in designing optimization algorithms for deep learning tasks, including traditional stochastic gradient descent (SGD) and its adaptive variants such as Adam, RMSProp, and AdaGrad, as well as its signed variants such as SignSGD and LION. While these methods have demonstrated strong practical performance, the theoretical understanding of their convergence rates, especially for novel algorithms like LION, remains incomplete.

This paper addresses these challenges by providing a comprehensive convergence rate analysis of the LION optimizer for a specific type of constrained problem and general unconstrained problems. The results demonstrate that LION achieves a convergence rate of $\mathcal{O}(\sqrt{d}K^{-1/4})$, where d is the problem dimension and is the number of iterations. This rate theoretically matches the current lower bound for stochastic gradient methods in nonconvex optimization with respect to K, and empirically matches the lower bound with respect to d in practical deep learning applications. These findings provide theoretical insights into the behavior of LION, contributing to a deeper understanding of its performance in large-scale deep learning problems.

This paper addresses the core research problem of the project of Science and Technology Innovation 2030: "Major Program on New Generation Artificial Intelligence", Project No. 2022ZD0160300, hosted by Ministry of Science and Technology of the People's Republic of China. Specifically, this work contributes to topic 2 of the major project, titled "Foundational Algorithms for Machine Learning in General Vision", with the topic No. 2022ZD0160302.

The goal of topic 2, "Foundational Algorithms for Machine Learning in General Vision", is to develop and analyze efficient optimization algorithms tailored for large-scale intelligent models, which impose higher demands on training data, computational resources, storage platforms, and energy efficiency. The project emphasizes the design and analysis of highly efficient optimization algorithms specifically suited for large models.

Key objectives include designing single-core optimization algorithms for general nonconvex problems, such as minimization, bilevel, and min-max optimization. By leveraging techniques such as momentum acceleration, this topic aims to improve the convergence properties of stochastic optimization methods and provide well-defined stopping conditions and complexity measures for key computational costs. Furthermore, this topic explores distributed optimization meth-

ods with low communication complexity, enabling efficient multi-machine, multi-core optimization. Additionally, it investigates learning-based optimization approaches that incorporate prior knowledge from training data and neural network architectures to break theoretical complexity lower bounds under certain conditions.

The algorithms developed and analyzed under this topic aim to support the fast training and inference of large-scale models, addressing critical challenges in general machine learning optimization. The results of this paper contribute directly to these objectives by advancing the theoretical understanding of the LION optimizer.