自动化学报
ACTA AUTOMATICA SINICA

一种基于单比特通信压缩的大语言模型训练方法

陈楚岩 刘烨 贾维宸 何雨桐 袁坤 王立威

An Effective Algorithm for LLMs Training With One-bit Communication Compression

CHEN Chu-Yan, LIU Ye-Xu, JIA Wei-Chen, HE Yu-Tong, YUAN Kun, WANG Li-Wei

在线阅读 View online: https://doi.org/10.16383/j.aas.c250087

一种基于单比特通信压缩的大语言模型训练方法

陈楚岩 [1] 刘烨谞 [1] 贾维宸 [1] 何雨桐 [1] 袁 坤 [1] 王立威 [1]

摘 要 近年来, 大语言模型研究取得突破性进展. 针对大模型分布式训练中通信开销高、算力利用率低的问题, 提出一种基于 Adam-mini 优化器的单比特通信压缩算法——单比特 Adam-mini. 该算法通过减少二阶动量参数, 使得能够以较小的通信代价精确计算全局二阶动量, 从而简化通信误差补偿机制的设计. 单比特 Adam-mini 不仅避免了现有单比特 Adam 算法中通信开销较大的预热阶段, 还具备可证明的线性加速性质, 确保分布式训练的高效性. 实验结果表明, 该算法在多种任务上表现优异, 并且可以兼容稀疏压缩器, 为大模型训练提供更高效的解决方案.

关键词 大语言模型; 深度学习; 分布式优化; 大规模分布式训练; 通信压缩

引用格式 陈楚岩, 刘烨谞, 贾维宸, 何雨桐, 袁坤, 王立威. 一种基于单比特通信压缩的大语言模型训练方法. 自动化学报, 2026, **52**(8): 1659−1676

DOI 10.16383/j.aas.c250087 **CSTR** 32138.14.j.aas.c250087

An Effective Algorithm for LLMs Training With One-bit Communication Compression

CHEN Chu-Yan[1] LIU Ye-Xu[1] JIA Wei-Chen[1] HE Yu-Tong[1] YUAN Kun[1] WANG Li-Wei[1]

Abstract Recent advancements in large language models (LLMs) research have achieved remarkable breakthroughs. This paper tackles the challenges of high communication costs and low computational efficiency in distributed training for large-scale models by introducing a one-bit communication compression algorithm based on the Adam-mini optimizer, named 1-bit Adam-mini. By significantly reducing the number of second-order momentum parameters through aggregation strategies, our algorithm achieves full-precision global synchronization of second-order momentum with minimal communication overhead. Additionally, by incorporating an error feedback mechanism, 1-bit Adam-mini not only avoids the communication-intensive warm-up phase required by existing 1-bit Adam algorithms but also provides provable linear speedup properties, ensuring the efficiency of distributed training. Experimental results demonstrate that the algorithm exhibits exceptional performance across a variety of tasks and can be seamlessly extended to sparse compressors, presenting a more efficient solution for large-scale model training.

Keywords large language model; deep learning; distributed optimization; large-scale distributed training; communication compression

Citation Chen Chu-Yan, Liu Ye-Xu, Jia Wei-Chen, He Yu-Tong, Yuan Kun, Wang Li-Wei. An effective algorithm for LLMs training with one-bit communication compression. *Acta Automatica Sinica*, 2026, **52**(8): 1659−1676

近年来, 大模型 (如大语言模型和多模态大模型等) 在阅读、理解、推理、规划以及图像生成等多项任务中取得突破性进展[1−3]. 然而, 这类模型通常包含数十亿甚至更多的参数[4−6], 其训练过程需要消耗大量的计算和内存资源, 这为大模型的训练带来巨大挑战. 为提升训练效率, 基于大规模算力集群的分布式方法已成为一种常见的解决方案. 这类算力集群通常由数千至数万个节点组成, 以满足大模型的高性能计算需求. 例如, 字节跳动公司搭建由一万张 GPU 组成的集群用于训练“豆包”大模型, 而美国 Meta 公司则使用两万张 GPU 集群来训练 LLaMA3 大模型.

虽然分布式方法显著提升了大模型训练速度, 但集群的算力利用率却相对较低. 据测算, 英伟达公司在使用 Megatron 框架训练 GPT (generative pre-trained Transformer) 系列模型时算力利用率[1]约为 50%[7−8], OpenAI 在预训练 GPT-3 模型时的算力利用率约为 44%[9], 而 Meta 在预训练 OPT 模型时的算力利用率仅为 30%[10]. 在分布式训练过程中, 各个计算节点需要频繁交换梯度和参数更新信息, 从而造成巨大的带宽开销和通信延迟, 显著降低算力利用率. 因此, 如何减少分布式训练中的通信开销, 提升集群的算力利用率, 成为大模型训练的关键问题.

为缓解大模型训练中的通信开销问题, 近年来

收稿日期 2025-03-06 录用日期 2025-06-09
Manuscript received March 6, 2025; accepted June 9, 2025
本文责任编委 鲁继文
Recommended by Associate Editor LU Ji-Wen
1. 北京大学 北京 100871
1. Peking University, Beijing 100871

[1] 本文中的算力利用率是指集群的实际每秒浮点运算次数与理论峰值每秒浮点运算次数的比值.

研究者们提出多种基于通信压缩的分布式优化算法[11−16]. 与直接传输完整的梯度或模型张量不同, 这些算法在通信前会使用压缩器 (compressor) 对张量进行大幅压缩, 从而显著减少通信量, 实现更高效的通信. 目前, 主流的压缩器主要包括量化 (quantization) 压缩器和稀疏化 (sparsification) 压缩器. 量化压缩器[11, 17−19] 通过将输入张量的每个元素映射到一个较小的离散值集合来实现压缩. 例如, 16 比特或 8 比特量化压缩器将张量的每个元素分别投影到由 16 比特或 8 比特表示的数值范围内. 而稀疏化压缩器[13, 20] 则通过丢弃张量中的部分元素来生成稀疏张量用于通信. 例如, rand-K 压缩器随机保留张量中的 K 个元素, 而 top-K 压缩器则保留张量中模值最大的 K 个元素. 已有研究表明, 基于通信压缩的分布式训练算法能够有效减少通信开销, 并显著提升集群的算力利用率.

尽管基于通信压缩的分布式训练算法已经得到广泛研究, 但现有工作仍存在如下不足之处:

1) 基于 Adam 的压缩算法研究相对欠缺. 自适应梯度算法 Adam[21] 是大模型训练中的主流优化器. 然而, 当前的通信压缩算法大多基于随机梯度下降 (stochastic gradient descent, SGD) 或 Momentum SGD 设计[11, 13−15, 18, 22], 而较少与 Adam 算法相结合. 如何以通信高效的方式有效近似 Adam 算法中的一阶和二阶动量, 是算法设计中的一个本质性难题.

2) 收敛性质的研究尚不成熟. 最近一些研究开始尝试基于 Adam 优化器设计压缩算法, 例如 QAdam[23] 和 Efficient-Adam[24]. 然而, 这些算法的收敛性分析仍不够完善, 尤其是无法证明算法具有线性加速性质[2], 而线性加速正是分布式训练中最为关键的理论特性之一.

3) 通信开销较大的算法预热机制. 为设计基于 Adam 且具有线性加速性质的通信压缩算法, 美国微软公司等机构提出单比特 Adam 算法[25]. 该算法基于 Adam 的二阶动量在训练后期趋于稳定的特性, 避免使用压缩梯度更新二阶动量倒数这一非线性操作, 简化收敛性分析. 然而, 为准确估计二阶动量, 单比特 Adam 需要进行通信无损的预热训练, 导致较大的通信开销, 影响了算法的实用性.

设计基于 Adam 的通信压缩算法的核心挑战在于: 二阶动量张量的倒数运算作为一种非线性操作, 使得压缩误差的补偿变得极为困难, 这也是导致现有研究存在局限性的根本原因. 值得注意的是, 近期提出的一种大模型优化器 Adam-mini[26] 通过调整算法结构, 在显著减少二阶动量参数的同时, 依然保持与 Adam 相当的性能表现. 由于该算法的二阶动量的参数量大幅减小, 本文能够以较小的通信代价精确计算全局二阶动量, 并结合通信误差补偿机制, 设计一种全新的通信压缩算法, 有效地解决上述局限性. 本文的主要贡献如下:

1) 新颖的单比特通信压缩算法. 本文基于 Adam-mini 优化器设计一种新颖的通信压缩训练方法——单比特 Adam-mini. 单比特 Adam-mini 不仅能够达到甚至超越单比特 Adam 算法的性能, 还大幅减少单比特 Adam 中需要全量通信的预热阶段, 从而更加节省通信开销, 实用性也更强.

2) 可证明的线性加速性质. 由于 Adam-mini 算法避免了二阶动量张量求逆的非线性操作, 引入通信压缩后的收敛性分析得以简化. 本文在标准假设下, 为单比特 Adam-mini 算法建立收敛性保证, 并证明了其具有线性加速性质, 从而在理论上保证了算法的性能.

3) 良好的实验表现与可扩展性. 本文在计算机视觉、自然语言处理等多种任务上进行实验. 实验结果表明, 单比特 Adam-mini 能够取得与单比特 Adam 相近甚至更高的测试准确率, 同时具备节省预热阶段全量通信开销的潜力. 此外, 单比特 Adam-mini 可以直接推广至稀疏压缩器, 展现了其良好的扩展性.

表 1 对比了单比特 Adam-mini 算法与当前主流通信压缩算法的性能, 其中, σ 表示梯度噪声, ϵ 为压缩误差上界, T 为迭代轮数. 表 1 中, 单机收敛速度指算法在单个节点上运行时的收敛速度, 而多机收敛速度则表示算法在 n 个节点上运行时的收敛速度. 从表 1 中可以看出, 单比特 Adam-mini 不仅兼容自适应梯度法, 还避免了全量通信预热的需求, 同时具备可证明的线性加速特性. 因此, 单比特 Adam-mini 是一种具有竞争力的大模型通信压缩训练算法.

1 相关工作

1.1 分布式优化

分布式优化旨在通过多机并行训练的方式提升算法的运算效率. 特别是在深度学习任务中, 多机协同训练能够显著提高计算效率、加速算法收敛. 从并行方式来看, 分布式训练主要包括数据并行、流水线并行和张量并行等模式. 其中, 数据并行[29] 通过将数据分配给不同节点来提高梯度采样效率; 流水线并行[30−31] 通过将模型分配给不同节点来克服内存开销; 张量并行[7] 则通过将矩阵乘法等算子拆

[2] 线性加速性质是指, 当算法收敛至指定精度时, 所需的迭代次数随着计算节点数量的增加而线性减少.

表 1　非凸光滑假设下单比特 Adam-mini 与其他通信压缩算法对比

Table 1　Comparison between 1-bit Adam-mini and other communication compression algorithms under the non-convex smoothness assumption

算法	Adam 兼容性	无需预热	压缩种类	单机收敛速率	多机收敛速率
Parallel SGD/Adam	√	√	无压缩	$\frac{\sigma}{\sqrt{T}}+\frac{1}{T}$	$\frac{\sigma}{\sqrt{nT}}+\frac{1}{T}$
DoubleSqueeze[27]	×	√	双向压缩	$\frac{\sigma}{\sqrt{T}}+\frac{\epsilon^{2/3}}{T^{2/3}}+\frac{1}{T}$	$\frac{\sigma}{\sqrt{nT}}+\frac{\epsilon^{2/3}}{T^{2/3}}+\frac{1}{T}$
1-bit Adam[25]	√	×	双向压缩	$\frac{\sigma}{\sqrt{T}}+\frac{\epsilon^{2/3}}{T^{2/3}}+\frac{1}{T}$	$\frac{\sigma}{\sqrt{nT}}+\frac{\epsilon^{2/3}}{T^{2/3}}+\frac{1}{T}$
QAdam[23]	√	√	单向压缩+权重压缩	$\frac{1}{\sqrt{T}}$	$\frac{1}{\sqrt{T}}$
Comp-AMS[28]	√	√	单向压缩	$\frac{\sigma^2+1}{\sqrt{T}}+\frac{1}{T}$	$\frac{\sigma^2+1}{\sqrt{nT}}+\frac{1}{T}$
Efficient-Adam[24]	√	√	双向压缩	$\frac{1}{\sqrt{T}}$	—
1-bit Adam-mini (本文)	√	√	双向压缩	$\frac{\sigma}{\sqrt{T}}+\frac{1}{T}$	$\frac{\sigma}{\sqrt{nT}}+\frac{1}{T}$

分至不同节点进行并行计算. 在理想情况下, 分布式优化的效率提升最多可与节点数量成正比, 即实现“线性加速”. 然而, 随着节点数量的增加, 分布式集群内的通信开销逐渐增大, 相比于逐渐降低的计算开销, 通信开销所占比重逐渐上升, 最终成为效率瓶颈. 为此, 学者们围绕通信高效理念设计多种通信节省机制, 并衍生出许多通信高效的分布式优化算法. 一是采用通信压缩[11, 13] 机制, 通过压缩每次传输的信息向量来减少每轮通信的信息量; 二是采用无中心通信, 使每个计算节点在任意规模的分布式集群中仅与少量邻居节点进行通信[32−40]; 三是采用低频通信, 各节点分别进行多轮梯度下降后再进行融合, 以降低通信频率[41−45].

1.2　通信压缩算法

通信压缩算法是通信高效的分布式优化算法研究中的重要领域. 2017 年, Alistarh 等[11] 提出基于量化思想的压缩器 QSGD, 显著降低有中心分布式优化中随 GPU 节点数量增加而带来的通信开销问题. 此后, 许多学者针对压缩器进行多样化的设计, 并产生一系列相关研究工作 [12, 17, 46−50]. 然而, 这些工作主要使用较为原始的压缩优化算法, 除引入压缩误差外, 并未对算法本身进行实质性改进. 2021 年, EF21 算法[14] 的出现显著改善了原始算法的收敛性问题, 使基于通信压缩的分布式优化算法能够有效克服不同计算节点之间的数据异质性挑战. 由 EF21 算法衍生的误差补偿机制广泛应用于后续的通信压缩算法设计中[51−52], 并在随机优化与确定性优化、凸优化与非凸优化等多类分布式优化问题中取得目前最优的通信复杂度[22, 53−56]. 此外, 采用多轮压缩通信机制的 NEOLITHIC 算法[15, 57] 也在随机凸与非凸优化问题中达到最优通信复杂度.

1.3　误差补偿技巧

误差补偿思想最早于 2014 年提出[18], 其核心是将每次进行梯度通信时产生的压缩误差累积起来, 并累加到下一次通信的梯度中. 然而, 基于该类误差补偿方法衍生的算法在理论与实验中均存在一定的局限性, 其理论证明往往受限于单节点分析、梯度有界等强假设条件[58]. 2019 年, DIANA 算法[59] 首次采用压缩梯度差的误差补偿方式, 有效解决了数据异质性问题. 这一思想随后进一步应用于 2020 年提出的 ADIANA 算法[55] 中. 2021 年, Richtárik 等[14] 提出基于压缩梯度差思想的误差补偿机制 EF21. 该机制在标准假设下具有严格的收敛性保证, 能够显著缓解分布式优化任务中的数据异质性问题, 并且在实验性能上优于 2014 年提出的误差补偿机制, 因此在后续研究中得到广泛采用. EF21 的核心思想是为每个计算节点维护一个局部梯度的追踪器, 当局部梯度随着迭代逐渐收敛时, 其与局部梯度追踪器之间的差异会趋近于零, 通过对该梯度差进行通信, 能够有效降低压缩误差. 2024 年, Fatkhullin 等[22] 提出 EF21-MSGD 算法, 通过引入动量机制, 进一步弥补 EF21 在处理随机性优化问题时的不足.

1.4　自适应优化算法

自适应优化算法旨在利用优化过程中的历史信息动态调整学习率等参数, 从而实现算法的加速收敛. 2011 年, Duchi 等[60] 提出的 AdaGrad 算法通过引入梯度的二阶信息来平衡各维度的步长, 在梯度稀疏的问题中表现优异. 随后, AdaDelta 算法[61] 采用移动平均的思想, 有效缓解了 AdaGrad 算法中步长下降过快的问题. 2014 年, Kingma 等[21] 提出

的 Adam 算法将自适应学习率与一阶动量相结合, 凭借其出色的收敛速度和适应性, 成为深度学习模型训练中的主流优化算法. 2017 年, AdamW 算法[62] 进一步优化 Adam 算法与 ℓ_2 -正则化的兼容性问题. 在大模型时代, Zhang 等[26] 于 2024 年提出的 Adam-mini 算法通过合并二阶动量信息, 在节省一半优化器状态内存的同时, 保持与 Adam 算法相近的性能.

2 问题模型与自适应优化器

2.1 问题模型

本文考虑如下的分布式随机优化问题:

$$\min_{\boldsymbol{x}\in\mathbf{R}^d} f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}[f_i(\boldsymbol{x}) = \mathrm{E}_{\boldsymbol{\zeta}\sim D_i}F_i(\boldsymbol{x},\,\boldsymbol{\zeta})] \tag{1}$$

其中, n 表示计算集群中的节点数量, d 表示向量的维度. 在上述问题中, $\boldsymbol{\zeta}$ 是一个随机变量, 代表节点 i 处可用的本地数据, 且其服从本地分布 D_i. 每个节点 i 可以访问其本地损失函数的随机梯度 $\nabla F_i(\boldsymbol{x},\,\boldsymbol{\zeta})$, 但需要与集群中的其他节点通信以获取这些节点的梯度信息. 实际上, 每个节点的本地数据分布 D_i 通常是不同的, 因此对于任意节点 i 和 j, 往往有 $f_i(\boldsymbol{x}) \neq f_j(\boldsymbol{x})$ 成立, 这种现象称为数据异质性.

2.2 Adam 优化器

Adam[21] 是当前深度学习领域的主流优化算法, 在众多深度学习任务中表现出良好的收敛速度和泛化性能, 同时对超参数的选择具有较强的鲁棒性. Adam 主要通过结合一阶动量和二阶动量来自动调整学习率. 假设 $\boldsymbol{g}_t$ 是 t 时刻的随机梯度, Adam 算法的迭代式为

$$\boldsymbol{m}_{t+1} = \beta_1\boldsymbol{m}_t + (1-\beta_1)\boldsymbol{g}_t \tag{2a}$$

$$\boldsymbol{v}_{t+1} = \beta_2\boldsymbol{v}_t + (1-\beta_2)(\boldsymbol{g}_t)^2 \tag{2b}$$

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma\frac{\boldsymbol{m}_{t+1}}{\sqrt{\boldsymbol{v}_{t+1}}+\epsilon} \tag{2c}$$

其中, γ 是学习率, ϵ 是一个较小的常数, β_1 和 β_2 分别是累积一阶动量和二阶动量的移动平均参数. 此外, 式 (2b) 中的 $(\boldsymbol{g}_t)^2$ 与式 (2c) 中的 $\frac{\boldsymbol{m}_{t+1}}{\sqrt{\boldsymbol{v}_{t+1}}+\epsilon}$ 均为逐元素操作的运算符.

2.3 Adam-mini 优化器

虽然 Adam 优化器性能稳定且优异, 但其迭代过程需要存储一阶和二阶动量, 占用较高的显存空间. 为节省显存, Adam-mini 优化器[26] 对 Adam 进行改进. Adam-mini 将 Adam 中的二阶动量信息进行合并, 在减少大部分二阶动量参数的同时, 保持与 Adam 相近的性能. 具体而言, Adam-mini 算法对参数 $\boldsymbol{x}$ 进行分组, 每组参数共享同一个二阶动量标量. 假设参数分为 B 组, 对于第 b 组 ($b = 1,\,\cdots,\,B$), 记该组参数的数目为 d_b, 梯度为 $\boldsymbol{g}^{(b)} \in \mathbf{R}^{d_b}$, 则该组参数的更新共用同一个二阶动量标量 $v^{(b)}$. Adam-mini 的迭代式为

$$\boldsymbol{m}_{t+1}^{(b)} = \beta_1\boldsymbol{m}_t^{(b)} + (1-\beta_1)\boldsymbol{g}_t^{(b)} \tag{3a}$$

$$v_{t+1}^{(b)} = (1-\beta_2)\cdot\mathrm{mean}(\boldsymbol{g}_t^{(b)}\odot\boldsymbol{g}_t^{(b)}) + \beta_2\cdot v_t^{(b)} \tag{3b}$$

$$\boldsymbol{x}_{t+1}^{(b)} = \boldsymbol{x}_t^{(b)} - \gamma\frac{\boldsymbol{m}_{t+1}^{(b)}}{\sqrt{v_{t+1}^{(b)}}+\epsilon} \tag{3c}$$

Adam 优化器需要存储 d 维向量 $\boldsymbol{v}$, 而 Adam-mini 优化器只需存储 $\{v^{(b)}\}_{b=1}^{B}$ 这 B 个标量. 由于 $B \ll d$, Adam-mini 显著降低了内存开销. 大量实验表明, Adam-mini 优化器的实际性能与 Adam 几乎一致, 未出现显著的性能损失[26]. 在 Adam-mini 的实际应用中, 分组策略对其收敛性能有显著影响[26]. 一般来说, 对于模型中越重要的层, 会采用更细粒度的分组策略, 并分配更多的二阶动量. 在 Adam-mini 中, 对于卷积神经网络, 每个矩阵使用一个二阶动量参数进行更新. 对于 Transformer 结构的网络, 嵌入层和输出层中的每个参数单独维护一个二阶动量值, 多头注意力层中每个注意力头的参数维护一个二阶动量值, 对于其他部分, 则每一个矩阵维护一个二阶动量值. 由于 Transformer 模型的参数主要集中在多头自注意力层和前馈神经网络中, 这种分组方式仍然能够节省绝大部分二阶动量. 在后续单比特 Adam-mini 的实现过程中, 本文将沿用这一分组策略.

3 单比特 Adam 算法及其局限性

3.1 单比特 Adam 算法

单比特 Adam[25] 是一种基于自适应优化器的通信压缩算法. 该算法的核心思想在于, 每个工作节点对动量进行压缩处理, 随后将压缩后的动量传输至服务器以执行求平均操作. 具体而言, 单比特 Adam 中每个工作节点 i 按照下式计算局部梯度并更新局部动量:

$$\boldsymbol{g}_t^{(i)} = \nabla F_i(\boldsymbol{x}_t,\,\boldsymbol{\zeta}_t^{(i)}) \tag{4}$$

$$\boldsymbol{m}_t^{(i)} = \beta_1 \boldsymbol{m}_{t-1}^{(i)} + (1-\beta_1)\boldsymbol{g}_t^{(i)} \tag{5}$$

随后各节点压缩动量 $\hat{\boldsymbol{m}}_t^{(i)}$, 计算补偿误差 $\boldsymbol{\delta}_t^{(i)}$, 并将压缩动量传输至参数服务器:

$$\hat{\boldsymbol{m}}_t^{(i)} = \boldsymbol{C}_\omega[\boldsymbol{m}_t^{(i)} + \boldsymbol{\delta}_{t-1}^{(i)}] \tag{6}$$

$$\boldsymbol{\delta}_t^{(i)} = \boldsymbol{m}_t^{(i)} + \boldsymbol{\delta}_{t-1}^{(i)} - \hat{\boldsymbol{m}}_t^{(i)} \tag{7}$$

其中, $\boldsymbol{C}_\omega[\cdot]$ 代表单比特压缩算子, 其定义为 $\boldsymbol{C}_\omega[\boldsymbol{x}] = (\|\boldsymbol{x}\|_1/d)\cdot \mathrm{sign}(\boldsymbol{x})$. 参数服务器接收所有工作节点的压缩动量, 并计算其平均值以得到全局动量 $\frac{1}{n}\sum_{i=1}^n \hat{\boldsymbol{m}}_t^{(i)}$, 其中 d 是向量 $\boldsymbol{x}$ 的维度. 由于全局动量需要发送给工作节点以更新参数, 因此在发送前需对全局动量进行压缩并执行误差补偿:

$$\overline{\boldsymbol{m}}_t = \boldsymbol{C}_\omega\left[\frac{1}{n}\sum_{i=1}^n \hat{\boldsymbol{m}}_t^{(i)} + \overline{\boldsymbol{\delta}}_{t-1}\right] \tag{8}$$

$$\overline{\boldsymbol{\delta}}_t = \frac{1}{n}\sum_{i=1}^n \hat{\boldsymbol{m}}_t^{(i)} + \overline{\boldsymbol{\delta}}_{t-1} - \overline{\boldsymbol{m}}_t \tag{9}$$

最后, 每个工作节点收到 $\overline{\boldsymbol{m}}_t$ 后更新参数 $\boldsymbol{x}_t$:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma \frac{\overline{\boldsymbol{m}}_t}{\sqrt{\boldsymbol{v}_{T_w}} + \epsilon} \tag{10}$$

其中, 向量 $\boldsymbol{v}_{T_w}$ 表示第 T_w 次迭代时的二阶动量. 由于单比特 Adam 算法未对二阶动量进行压缩, 每次传输二阶动量将导致巨大的通信开销. 文献 [25] 指出, 二阶动量 $\boldsymbol{v}_{T_w}$ 在训练初期变化显著, 而在训练后期趋于稳定. 为节省通信开销, 当二阶动量稳定时, 单比特 Adam 算法不再对其进行更新, 从而避免对二阶动量的计算和通信需求. 超参数 T_w 是通过运行完整 Adam 算法 (2a) ~ (2c) 确定的二阶动量稳定时刻, 本文称这一阶段为算法预热阶段.

3.2　局限性一: 预热阶段通信开销巨大

由迭代式 (10) 可以看出, 为获得二阶动量 $\boldsymbol{v}_{T_w}$, 单比特 Adam 算法需要执行 T_w 步的标准 Adam 算法进行预热. 当预热步数 T_w 较大时, 这一过程将产生较大的通信开销.

在 RTE 数据集上使用单比特 Adam 训练 RoBERTa-base 模型, 得到的累计通信量 (以比特为单位) 随迭代步数的变化曲线如图 1 所示. 在预热阶段, 由于传输的是全量梯度, 曲线呈现陡峭上升趋势; 而在压缩阶段, 由于梯度的每个元素被压缩至单比特, 总通信量随迭代步数的增速显著放缓. 转折点处的纵坐标值代表预热阶段的通信量. 可以观察到, 当预热步数占总步数的 38% 时, 预热阶段的通信量占总通信量的 56%.

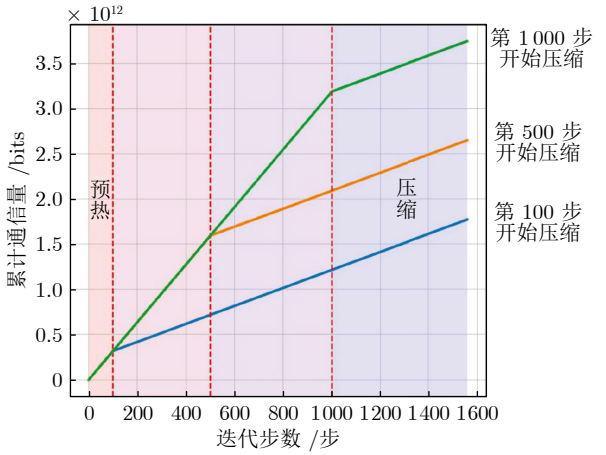


图 1　单比特 Adam 算法累积通信量与迭代步数的关系图

Fig. 1　Relationship between accumulated communication and iteration steps for 1-bit Adam

图 2 在相同实验设定下展示了模型训练后的准确率、总通信量与预热步数的关系. 可以观察到, 在相同的预热步数下, 单比特 Adam 算法随着预热步数的增加, 总通信量上升, 预热步数所占比例也显著提高, 同时准确率也有所提升. 这表明, 减少通信量需要以牺牲算法效果为代价. 与此相比, 本文设计的新算法 (单比特 Adam-mini) 能够在少量预热步数下, 既保持较低的总通信量, 又兼具较高的准确率.

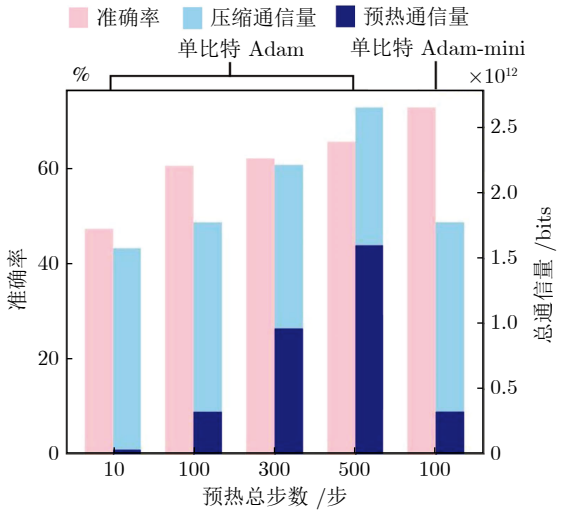


图 2　单比特 Adam 和单比特 Adam-mini 在不同预热步数下的总通信量开销比例与准确率对比图

Fig. 2　Accuracy and proportion of total communication overhead comparison between the 1-bit Adam algorithm and the 1-bit Adam-mini algorithm under different preheating steps

3.3　局限性二: 二阶动量倒数的非稳定性

单比特 Adam 在迭代步数 $t > T_w$ 后, 使用 T_w

时刻的二阶动量 $\boldsymbol{v}_{T_w}$ 代替 Adam 中的二阶动量 $\boldsymbol{v}_t$ 进行更新. 尽管文献 [25] 中的实验表明, 当标准 Adam 算法在迭代步数 t 足够大时, 二阶动量 $\boldsymbol{v}_t$ 的二范数 $\|\boldsymbol{v}_t\|_2$ 趋于稳定. 然而, 根据式 (10), 二阶动量在实际算法中以 $(\sqrt{\boldsymbol{v}_t}+\epsilon)^{-1}$ 的形式与更新量相乘. 即使 $\boldsymbol{v}_t$ 有微小变化, $(\sqrt{\boldsymbol{v}_t}+\epsilon)^{-1}$ 仍可能产生较大的波动.

为验证上述结论, 使用 Adam 算法在 RTE 任务上微调 RoBERTa-base 模型. 令 $\boldsymbol{h}_t=(\sqrt{\boldsymbol{v}_t}+\epsilon)^{-1}$ 和 $\boldsymbol{h}_{T_w}=(\sqrt{\boldsymbol{v}_{T_w}}+\epsilon)^{-1}$, 图 3 展示了 $\|\boldsymbol{h}_t-\boldsymbol{h}_{T_w}\|/\|\boldsymbol{h}_{T_w}\|$ 随迭代步数的变化关系, 其中 $T_w=300$. 结果表明, 即使在训练后期, $\boldsymbol{h}_t$ 与 $\boldsymbol{h}_{T_w}$ 之间的差距仍然显著. 这意味着使用 $\boldsymbol{v}_{T_w}$ 代替 $\boldsymbol{v}_t$ 将引入较大的误差. 因此, 单比特 Adam 算法冻结二阶动量的策略会导致较大的动量误差, 从而降低算法的性能表现.

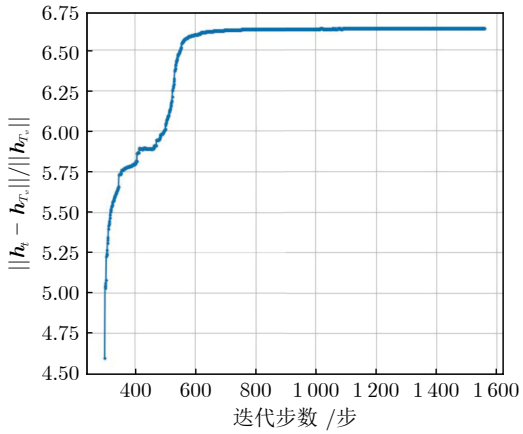


图 3 $\|\boldsymbol{h}_t-\boldsymbol{h}_{T_w}\|/\|\boldsymbol{h}_{T_w}\|$ 与迭代步数关系图

Fig. 3 Relationship between $\|\boldsymbol{h}_t-\boldsymbol{h}_{T_w}\|/\|\boldsymbol{h}_{T_w}\|$ and iteration steps

4 单比特 Adam-mini 算法

1) 核心思想. 单比特 Adam 算法的两个局限性源于无法以通信高效的方式准确计算全局二阶动量 $\boldsymbol{v}_t$. 若要精确计算二阶动量, 单比特 Adam 算法需要在每次迭代时传输完整梯度; 而若要节省通信开销, 单比特 Adam 只能使用冻结的二阶动量, 从而引入动量偏差. 因此, 如何设计一种能够以较低通信开销计算全局二阶动量的方法是算法设计的核心挑战.

本文注意到, 使用 Adam-mini 算法可以解决这一核心挑战. 从式 (3b) 可以看出, Adam-mini 的二阶动量仅包含 B 个标量. 传输这 B 个标量 $\{v^{(b)}\}_{b=1}^{B}$, 而非 d 维向量 $\boldsymbol{v}$, 具有极高的通信效率. 因此, 基于 Adam-mini 计算全局二阶动量 $\{v^{(b)}\}_{b=1}^{B}$ 天然具备通信节省的优势. 由于二阶动量的参数量显著减少, 可以在每次迭代中直接传输 Adam-mini 的二阶动量, 而无需像单比特 Adam 算法那样使用冻结的二阶动量, 从而有效缓解单比特 Adam 的预热问题以及二阶动量冻结导致的误差问题.

2) 算法设计. 单比特 Adam-mini 算法与单比特 Adam 算法十分相似, 主要区别在于二阶动量的更新方式. 由于 Adam-mini 算法的二阶动量参数较少, 单比特 Adam-mini 未对二阶动量进行压缩或冻结. 将单比特 Adam-mini 的二阶动量更新方式设计为

$$v_t^{(i,\,b)}=\mathrm{mean}(\boldsymbol{g}_t^{(i,\,b)}\odot\boldsymbol{g}_t^{(i,\,b)}) \tag{11}$$

$$v_t^{(b)}=\frac{1-\beta_2}{n}\sum_{i=1}^{n}v_t^{(i,\,b)}+\beta_2\cdot v_{t-1}^{(b)} \tag{12}$$

其中, $\boldsymbol{g}_t^{(i,\,b)}=\nabla F(\boldsymbol{x}_t^{(b)},\,\boldsymbol{\zeta}_t^{(i,\,b)})$ 表示节点 i 在本地参数 $\boldsymbol{x}_t$ 的第 b 个分组 $\boldsymbol{x}_t^{(b)}$ 上的随机梯度. 单比特 Adam-mini 算法的参数分组方式与 Adam-mini 一致, 具体细节可参见第 2.3 节. 由于 $v_t^{(i,\,b)}$ 是标量, 计算 $(1/n)\sum_{i=1}^{n}v_t^{(i,\,b)}$ 的通信开销非常小. 单比特 Adam-mini 的迭代细节详见算法 1. 尽管算法 1 的迭代格式以服务器与工作节点的方式描述, 但其可以在高性能计算集群中以更高效的方式运行, 更多实现细节参见附录 A.

算法 1. 单比特 Adam-mini 算法

1: **初始化.** 模型初始参数 $\boldsymbol{x}_0$, 初始累积误差 $\boldsymbol{\delta}_{-1}^{(i)}=\boldsymbol{0}$ $(i=1,\,2,\,\cdots,\,n)$, $\overline{\boldsymbol{\delta}}_{-1}=\boldsymbol{0}$, 初始一阶动量 $\boldsymbol{m}_{-1}=\boldsymbol{0}$, 初始二阶动量 $\boldsymbol{v}_{-1}=\boldsymbol{0}$; 学习率 γ, Adam 优化器参数 β_1, β_2, 优化器状态分组数目 B 以及分组策略 $\boldsymbol{\Pi}_B$, 总迭代次数 T.

2: 按照分组策略 $\boldsymbol{\Pi}_B$, 运行 T_w 步标准的 Adam-mini 算法, 并存储得到的二阶动量 $\overline{v}_{T_w}^{(b)}$ $(b=1,\,\cdots,\,B)$.

3: **for** $t=T_w,\,\cdots,\,T$ **do**

4: 在工作节点 i 执行:

5: 随机抽取 $\boldsymbol{\zeta}_t^{(i)}$ 并计算局部随机梯度 $\boldsymbol{g}_t^{(i)}=\nabla F_i(\boldsymbol{x}_t,\,\boldsymbol{\zeta}_t^{(i)})$.

6: 使用分组策略 $\boldsymbol{\Pi}_B$ 将 $\boldsymbol{g}_t^{(i)}$, $\boldsymbol{m}_t^{(i)}$, $\boldsymbol{x}_t$ 划分成 $\boldsymbol{g}_t^{(i,\,b)}$, $\boldsymbol{m}_t^{(i,\,b)}$, $\boldsymbol{x}_t^{(b)}$ $(b=1,\,\cdots,\,B)$.

7: **for** $b=1,\,\cdots,\,B$ **do**

8: 更新局部动量变量 $\boldsymbol{m}_t^{(i,\,b)}=\beta_1\boldsymbol{m}_{t-1}^{(i,\,b)}+(1-\beta_1)\boldsymbol{g}_t^{(i,\,b)}$.

9: 压缩 $\hat{\boldsymbol{m}}_t^{(i,\,b)}=\boldsymbol{C}_\omega[\boldsymbol{m}_t^{(i,\,b)}+\boldsymbol{\delta}_{t-1}^{(i,\,b)}]$.

10: 通过 $\boldsymbol{\delta}_t^{(i,\,b)}=\boldsymbol{m}_t^{(i,\,b)}+\boldsymbol{\delta}_{t-1}^{(i,\,b)}-\hat{\boldsymbol{m}}_t^{(i,\,b)}$ 更新压缩误差.

11: 计算梯度的均方值 $v_t^{(i,\,b)}=\mathrm{mean}(\boldsymbol{g}_t^{(i,\,b)}\odot\boldsymbol{g}_t^{(i,\,b)})$.

12: 将 $\hat{\boldsymbol{m}}_t^{(i,\,b)}$, $v_t^{(i,\,b)}$ 发送给服务器.

13: 在服务器端执行:

14: 更新全局二阶动量 $\overline{v}_t^{(b)} = \beta_2\, \overline{v}_{t-1}^{(b)} + \frac{1-\beta_2}{n} \times \sum_{i=1}^n v_t^{(i,\,b)}$.

15: 压缩全局动量为 $\overline{\boldsymbol{m}}_t^{(b)} = \boldsymbol{C}_\omega\big[\frac{1}{n}\sum_{i=1}^n \hat{\boldsymbol{m}}_t^{(i,\,b)} + \overline{\boldsymbol{\delta}}_{t-1}^{(b)}\big]$.

16: 更新压缩误差 $\overline{\boldsymbol{\delta}}_t^{(b)} = \frac{1}{n}\sum_{i=1}^n \hat{\boldsymbol{m}}_t^{(i,\,b)} + \overline{\boldsymbol{\delta}}_{t-1}^{(b)} - \overline{\boldsymbol{m}}_t^{(b)}$.

17: 将 $\overline{\boldsymbol{m}}_t^{(b)}$, $\overline{v}_t^{(b)}$ 发送给所有工作节点.

18: 在工作节点 i 执行:

19: 更新本地模型 $\boldsymbol{x}_{t+1}^{(b)} = \boldsymbol{x}_t^{(b)} - \gamma\, \overline{\boldsymbol{m}}_t^{(b)}/(\sqrt{\overline{v}_t^{(b)}} + \epsilon)$.

20: **end for.**

21: **end for.**

22: **输出.** $\boldsymbol{x}_T$.

5　单比特 Adam-mini 算法的收敛性分析

为建立 Adam-mini 算法的收敛性, 本文参考 AMSGrad 算法[63], 将二阶动量的更新方式修改为 $\hat{v}_t^{(b)} = \max\{\hat{v}_{t-1}^{(b)},\, \overline{v}_t^{(b)}\}$, 即使用 $\hat{v}_t^{(b)}$ 作为二阶动量. 参数更新式为 $\boldsymbol{x}_{t+1}^{(b)} = \boldsymbol{x}_t^{(b)} - \gamma\, \overline{\boldsymbol{m}}_t^{(b)}/(\sqrt{\hat{v}_t^{(b)}} + \epsilon)$. 这样的修改主要是为能够在 Adam 类算法中进行收敛性分析, 同时作如下的假设.

假设 1 (光滑性). $\forall i \in [n]$, f_i 为 L 光滑的, 即 $\forall \boldsymbol{x},\, \boldsymbol{y} \in \mathbf{R}^d$, $\|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\| \le L\|\boldsymbol{x} - \boldsymbol{y}\|$.

假设 2 (无偏有界的随机梯度). $\forall t \ge 1$, $\forall i \in [n]$, $\mathrm{E}[\boldsymbol{g}_t^{(i)}] = \nabla f_i(\boldsymbol{x}_t)$, $\|\boldsymbol{g}_t^{(i)}\| \le G$, 并且 $\boldsymbol{g}_t^{(i)}$ 之间相互独立.

假设 3 (有界方差). $\forall t \ge 1$, $\forall i \in [n]$, $\mathrm{E}[\|\boldsymbol{g}_t^{(i)} - \nabla f_i(\boldsymbol{x}_t)\|^2] \le \sigma^2$.

假设 4 (收缩性质的压缩器). 压缩器 $\boldsymbol{C}_\omega : \mathbf{R}^d \to \mathbf{R}^d$, $\forall \boldsymbol{x} \in \mathbf{R}^d$ 有 $\|\boldsymbol{C}_\omega[\boldsymbol{x}] - \boldsymbol{x}\| \le q\|\boldsymbol{x}\|$. 其中常数 $q \in [0,\, 1)$.

以上假设是合理的. 其中, 假设 1 和假设 2 中的无偏性以及假设 3 是分布式优化中的标准假设, 假设 4 是压缩通信算法的标准假设, 同时假设 2 中的有界部分是 Adam 算法收敛性分析中常用的假设[63]. 满足假设 4 的压缩器也比较常见. 在实验部分, 本文主要使用如下两类压缩器.

1) 单比特压缩器. 该压缩器的数学表达式为 $\boldsymbol{C}_\omega[\boldsymbol{x}] = \frac{\|\boldsymbol{x}\|_1}{d}\mathrm{sign}(\boldsymbol{x})$, 对应的系数 $q^2 = 1 - 1/d$.

2) top-K 压缩器. 若 $\boldsymbol{x}_i$ 为分量绝对值前 K 大的元素, 则 $\boldsymbol{C}_\omega[\boldsymbol{x}]_i = \boldsymbol{x}_i$; 否则, $\boldsymbol{C}_\omega[\boldsymbol{x}]_i = 0$, 对应的系数为 $q^2 = 1 - K/d$.

在上述假设下, 我们可以得到关于单比特 Adam-mini 算法的收敛性结果. 具体证明细节详见附录 B.

定理 1. 记

$$C_0 = G + \epsilon$$

$$C_1 = \frac{\beta_1}{1-\beta_1} + \sqrt{\frac{15q^2}{(1-q)^3(1-\beta_1)^2(1-r)}}$$

$$f^* = \min f$$

其中, $r = \max\{(\beta_1+1)/2,\, (q+3)/4\}$.

在假设 1 ~ 4 成立的条件下, 算法 1 满足:

$$\left(1 - \frac{2\gamma C_0 L}{\epsilon^2}\right)\frac{1}{T+1}\sum_{t=0}^{T}\mathrm{E}[\|\nabla f(\boldsymbol{x}_t)\|^2] \le 2C_0\left(\frac{\mathrm{E}[f(\boldsymbol{x}_0)] - f^*}{(T+1)\gamma} + \frac{\gamma^2 C_1^2 L^2 G^2}{2\epsilon^3} + \frac{\gamma L\sigma^2}{n\epsilon^2} + \frac{\gamma C_1(2C_1+1)LG^2 d}{(T+1)\epsilon^2} + \frac{(C_1+1)G^2 d}{(T+1)\epsilon}\right)$$

基于上述定理, 将学习率 γ 设为适当的值后, 可以得到如下收敛速度:

推论 1. 取 $\gamma = \min\{\frac{\epsilon^2}{4C_0 L},\, \frac{\sqrt{n}}{\sigma\sqrt{T}}\}$, 则在假设 1 ~ 4 条件下, 算法 1 满足:

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathrm{E}[\|\nabla f(\boldsymbol{x}_t)\|^2] \le \mathrm{O}\left(\frac{\sigma}{\sqrt{nT}} + \frac{1}{T}\right)$$

注 1 (线性加速). 推论 1 的结果表明, 在合理的假设与参数选择下, 当迭代次数 T 足够大时, 算法 1 的收敛速度主导项为 $\sigma/\sqrt{nT}$. 这意味着单比特 Adam-mini 算法收敛至精度 δ 需要 $\mathrm{O}(1/(n\delta^2))$ 次迭代. 当计算节点数 n 增大时, 算法所需的收敛次数线性减少, 因此称单比特 Adam-mini 算法具有线性加速性质. 与单比特 Adam-mini 算法相比, 通信压缩算法 QAdam[23] 和 Efficient-Adam[24] 均未证明其具有线性加速性质 (具体对比参见表 1).

注 2 (与无压缩算法的对比). 未使用通信压缩的分布式 SGD 与 Adam 算法的收敛速度阶次为 $\mathrm{O}(\sigma/\sqrt{nT} + 1/T)$. 与此相比, 单比特 Adam-mini 算法的压缩误差并未对算法的收敛性造成阶次性影响, 仅引入常数级别的影响 (具体对比参见表 1). 这体现了单比特 Adam-mini 算法对压缩误差的鲁棒性.

6　实验与结果分析

本文在多个机器学习任务上评估单比特 Adam-mini 算法的实际效果, 包括在 CIFAR-10 数据集上预训练 ResNet18 模型[64]、微调 ViT-L/16 模型[65], 在 GLUE 数据集[66] 和 SQuAD 数据集[3] 上微调 RoBERTa-base 模型[67], 利用 C4 数据集[4] 预训练 LLa-

[3] https://rajpurkar.github.io/SQuAD-explorer/

[4] https://huggingface.co/datasets/allenai/c4

MA 模型, 在 GLUE 数据集上微调十亿参数量级的 DeBERTa-v2-xxlarge 模型[68] 以及在 Math-10K 数据集上微调接近百亿参数量级的 LLaMA-7B 模型[69]. 实验结果说明在预热步数少的情形下, 单比特 Adam-mini 的性能表现能够超过单比特 Adam. 此外, 本文还验证了单比特 Adam-mini 能够推广至稀疏压缩器, 具有良好的扩展性.

6.1 图像分类模型对比实验

本文评估单比特 Adam-mini 在 CIFAR-10 数据集[5] 上预训练的性能, 采用的 ResNet18 模型是一个具有约 1 100 万个参数的卷积神经网络, 其通过残差连接提高训练效率, 广泛应用于图像分类当中. 我们在实验中使用 4 张 Nvidia RTX-4 090 GPU, 并将每张 GPU 批量大小设置为 32, 总共训练 50 个周期. 为公平比较, 采用带有 10% 预热的线性衰减学习率调度器, 并将最大学习率设置为 10^{-3}, 预热步数为 1 000 步. Adam 优化器的超参数 β_1 和 β_2 分别设置为 0.9 和 0.99. 表 2 展示了实验结果.

表 2 在 CIFAR-10 数据集上预训练 ResNet18 模型的结果
Table 2 Results of pretraining ResNet18 model on CIFAR-10 dataset

算法	损失	准确率 (%)	时间 (s/步)
Adam	0.011 8	92.06	0.072 4
Adam-mini	0.014 1	91.55	0.072 8
1-bit Adam	0.013 6	91.02	0.055 7
Eficient-Adam	**0.004 8**	91.68	0.053 0
Comp-AMS	0.017 1	92.02	0.053 8
1-bit Adam-mini	0.023 5	**92.08**	0.059 8

注: 加粗字体表示最优结果.

根据表 2 的实验结果, 在相同有限预热步数条件下, 单比特 Adam-mini 展现出两方面的优势: 1) 其准确率显著优于单比特 Adam 算法; 2) 内存占用量降低了 18.6% (归功于 Adam-mini 算法对二阶矩估计的优化). 特别地, 在预热步数为 2 000 时, 单比特 Adam 训练直接崩溃, 准确率也接近随机结果 10%, 而算法 1 没有出现崩溃的情况, 并且得到 91.82% 的测试准确率, 高于单比特 Adam 算法在有限预热步数下的最好结果.

此外, 本文在目前多模态领域主流的视觉模型上也进行了泛化实验的检验. Vision Transformer (ViT) 是多模态领域经典且基础的模型, 其在提取图像特征后, 输入语言模型 Transformer 进行运算, 使模型具有跨模态的能力. 本文实验使用 ViT-L/16 模型进行微调, 模型参数大小约 300 MB. 本文在 CIFAR-10 数据集上对比测试了 Adam、单比特 Adam 和单比特 Adam-mini 算法的效果, 优化器预热步数为 10, 其余设定与前文 CIFAR-10 实验一致, 实验结果和学习率等超参数详见表 3.

表 3 在 CIFAR-10 数据集上微调 ViT-L/16 模型的结果
Table 3 Results of finetuned ViT-L/16 model on CIFAR-10 dataset

算法	学习率	训练循环数	准确率 (%)	损失	预热步数
Adam	1×10^{-5}	10	97.79	0.000 3	—
1-bit Adam	1×10^{-5}	10	13.19	2.229 0	10
1-bit Adam-mini	1×10^{-5}	10	**97.51**	**0.000 6**	10
Adam	5×10^{-5}	10	97.63	0.000 2	—
1-bit Adam	5×10^{-5}	10	11.76	2.296 0	10
1-bit Adam-mini	5×10^{-5}	10	**96.70**	**0.000 3**	10

注: 加粗字体表示最优结果.

根据表 3 的结果, 我们发现单比特 Adam-mini 算法在 ViT 模型下也取得远好于单比特 Adam 的效果, 并与 Adam 结果相近, 说明我们的新算法在视觉任务上也有不错的泛化能力.

6.2 微调文本分类模型对比实验

本文评估了单比特 Adam-mini 算法在 GLUE 数据集上微调 RoBERTa-base 模型的表现. RoBERTa-base 是一个包含约 1.25 亿个参数的预训练语言模型, 广泛应用于自然语言理解任务. 我们使用 4 张 Nvidia RTX-4 090 GPU 进行训练, 每张 GPU 批量大小设置为 8. 参考相关工作[70], 本文采用线性衰减学习率调度器训练 10 个迭代周期, 并将二阶动量的预热设置为 1 000 步数 (SST2、QNLI) 与 100 步 (RTE、STS-B). 我们的优化器学习率在 SST2、QNLI 和 STS-B 任务上设为 2×10^{-5}, RTE 任务上设为 1×10^{-5}, Adam 优化器的超参数 β_1 和 β_2 分别设置为 0.9 和 0.999, 表 4 展示了具体结果.

根据表 4 的实验结果, 在 SST2、QNLI、RTE、STS-B 多个任务中单比特 Adam-mini 算法都相比单比特 Adam 表现出更好的效果, 与不压缩的全量方法 Adam 效果接近. 综合以上实验, 单比特 Adam-mini 算法在节省预热步数的同时兼具优良的性能. 此外, 结合表 2 和表 4 的结果可知, 尽管 Comp-AMS 与 Efficient-Adam 优化器取得较好的分类效果, 然而在与标准 Adam 算法和单比特 Adam-mini 算法对比时, 无论是在图像分类任务还是在文本分类任务上都仍有一定差距. 从而体现出单比特 Adam-mini 算法在与其他通信压缩算法相比时, 不仅在理论上具有收敛性优势, 也兼具实际表现中的良好性能.

[5] https://www.cs.toronto.edu/kriz/cifar.html

表 4　在 GLUE 数据集上微调 RoBERTa-base 模型的测试准确率 (SST2, QNLI, RTE) 和皮尔森系数 (STS-B)

Table 4　Test accuracy (SST2, QNLI, RTE) and Pearson coefficient (STS-B) of finetuned RoBERTa-base model on GLUE dataset

算法	SST2 (%)	QNLI (%)	RTE (%)	STS-B	时间 (s/步)
Adam	94.84	92.57	73.65	0.90	0.2757
Adam-mini	93.69	89.07	48.74	0.72	0.2800
vanilla 1-bit Adam	90.60	87.79	55.96	0.66	0.1405
1-bit Adam	93.58	91.47	60.65	−0.03	0.1412
Comp-AMS	92.43	88.83	70.40	0.89	0.1401
Efficient-Adam	92.78	88.80	68.23	0.89	0.1563
1-bit Adam-mini	**94.15**	**92.09**	**72.92**	**0.90**	0.1482

注: 加粗字体表示最优结果.

6.3　预训练语言模型对比实验

本文在 C4 数据集预训练 LLaMA 模型的任务上评估单比特 Adam-mini 算法. C4 数据集是一个经过清洗的大规模网页文本数据集, 常用于自然语言处理任务. 我们参照文献 [70] 选择较小的 130 M 的 LLaMA 模型进行预训练. 使用 4 张 Nvidia RTX-4090 GPU 进行训练, 每张 GPU 批量大小为 128, 采用 20000 步的迭代与 2000 步预热的余弦衰减学习率调度器, 并设置最大学习率为 2×10^{-3}, 序列长度设置为 256. 本文汇总了预热步数为 1000 的单比特 Adam-mini、单比特 Adam 和标准 Adam 优化器的对比结果. 由于预热步数为 1 000 步时单比特 Adam 会训练失败 (曲线发散), 我们展示了预热步数为 2000 的单比特 Adam-mini 的训练曲线. 训练损失函数值和通信量的关系如图 4 所示.

图 4 结果显示, 相比单比特 Adam 和标准 Adam 算法, 本文设计的单比特 Adam-mini 的损失函数能够在相同通信量情形下收敛到更低的值, 说明算法 1 在预训练任务中也能够起到节省通信的效果.

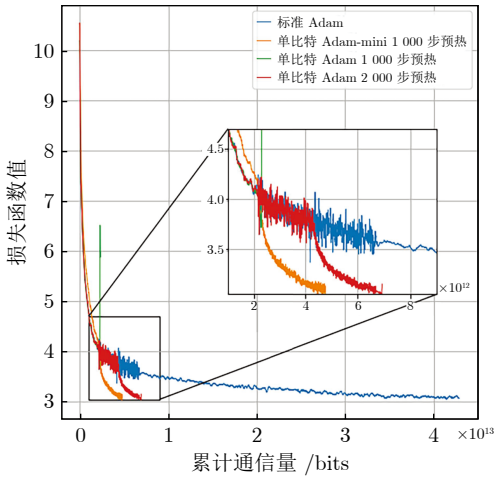


图 4　LLaMA 预训练任务中训练损失与通信量关系曲线

Fig. 4　Relationship curves between training loss and communication bits in LLaMA pretraining tasks

6.4　十亿至百亿参数规模语言模型微调实验

本文在 GLUE 数据集上对 DeBERTa-v2-xxlarge 模型进行微调. DeBERTa-v2-xxlarge 是具有约 15 亿参数的预训练语言模型, 其通过解纠缠注意力机制与增强掩码解码器技术, 显著提升对文本语义及位置关系的建模能力. 本文评估了标准 Adam、单比特 Adam-mini、单比特 Adam 算法的效果. 对每一种算法, 都采用如下两组不同优化器超参数中的效果较好者, 分别是 575 步的迭代与 57 步预热的线性衰减学习率调度器, 并设置最大学习率为 2×10^{-6}, 以及 1380 步的迭代与余弦退火学习率调度器, 并设置最大学习率为 3×10^{-6}. 本文使用 4 张 Nvidia A100 GPU 进行训练, 每张 GPU 的批量大小设置为 8, 序列长度为 256. 本文汇总了预热步数为 30 的标准 Adam 优化器、单比特 Adam-mini、单比特 Adam 的对比结果, 表 5 展示了具体结果.

表 5　在 GLUE 数据集 (子集: MRPC) 上的微调 DeBERTa-v2-xxlarge 模型的测试准确率与 F1 分数

Table 5　Test accuracy and F1 score of finetuned DeBERTa-v2-xxlarge model on GLUE dataset (subset: MRPC)

优化算法	准确率 (%)	F1 分数 (%)	运行时间 (s/步)
Adam	91.18	93.68	2.2085
1-bit Adam	89.95	92.79	0.9488
Comp-AMS	83.09	88.56	0.9421
Efficient-Adam	84.07	89.15	1.1588
1-bit Adam-mini	**90.44**	**93.07**	1.0039

注: 加粗字体表示最优结果.

实验结果表明, 在十亿级参数规模的语言模型微调中, 单比特 Adam-mini 相较于单比特 Adam 在子集 MRPC 任务的准确率和 F1 值上均表现出更优的性能.

此外, 本文在数学推理任务中, 对参数量接近百亿的 LLaMA-7B 模型进行微调. 微调使用 Math-10K 数据集[71], 评估微调后的 LLaMA-7B 大模型在 GSM8k[72]、SVAMP[73]、AQuA[74] 和 MultiArith[75] 测试集上的准确率, 学习率设为 3×10^{-4}. 本文使用 8 张 Nvidia RTX-4090 GPU 进行训练, 总批量大小设置为 16, 本文汇总了预热步数为 1600 步的标准 Adam、单比特 Adam、单比特 Adam-mini 优化器的对比结果, 详见表 6 所示.

上述实验表明, 在接近百亿级参数规模的语言模型微调中, 单比特 Adam-mini 同样相较于单比

表 6 在 Math-10K 数据集上微调 LLaMA-7B 模型的测试准确率 (GSM8K, SVAMP, AQuA, MultiArith) (%)
Table 6 Test accuracy (GSM8K, SVAMP, AQuA, MultiArith) of finetuned LLaMA-7B model on Math-10K dataset (%)

算法	GSM8K	SVAMP	AQuA	MultiArith
Adam	32.60	43.1	17.71	92.33
1-bit Adam	27.14	36.5	13.38	88.17
1-bit Adam-mini	**31.16**	**40.4**	**14.56**	**91.50**

注: 加粗字体表示最优结果.

特 Adam 在多个数学任务上表现出更高的准确率, 与全量 Adam 算法效果相近.

6.5 稀疏压缩器推广实验

本文使用单比特 Adam-mini 算法对 BERT-large 模型进行微调, 并将压缩器 $\boldsymbol{C}_\omega[\cdot]$ 替换成满足假设 4 的 top 5% 压缩器, 用以观察不同压缩器下单比特 Adam-mini 算法的效果. 本文使用 4 张 Nvidia RTX-4090 GPU 训练 3 个周期, 每张 GPU 批量大小设置为 24, 梯度累积设置为 8. 使用 10% 预热的线性衰减学习率调度器 Adam, 优化器的超参数 β_1 和 β_2 分别设置为 0.9 和 0.99, 表 7 展示了实验的结果.

表 7 在 SQuAD 数据集上微调 BERT-large 模型的测试准确率 (%)
Table 7 Test accuracy of finetuned BERT-large model on SQuAD dataset (%)

优化算法	Adam		Adam-mini	
压缩器	1-bit	top 5%	1-bit	top 5%
预热步数 = 20	78.32	—	**86.51**	**85.74**
预热步数 = 100	86.28	85.67	**86.71**	**86.17**
预热步数 = $+\infty$	**86.93**	**86.93**	86.49	86.49

注: 加粗字体表示最优结果.

根据表 7 结果, 将单比特 Adam-mini 算法替换成满足收缩性质的稀疏压缩器 (例如 top 5% 压缩器), 在预热步数为 20 和 100 的情形下, 微调实验仍然能够取得很好的效果, 说明单比特 Adam-mini 算法具有良好的扩展性.

更多的补充实验参见附录 C.

7 结束语

本文提出单比特 Adam-mini 算法, 有效解决了大模型通信压缩训练中通信开销大和计算效率低的问题. 通过引入 Adam-mini 优化器和通信压缩机制, 单比特 Adam-mini 不仅缓解了单比特 Adam 算法中预热阶段带来的巨大通信开销问题, 还具备理论上可证明的线性加速性质. 实验结果表明, 该算法在计算机视觉与自然语言处理任务中表现优异, 相较于传统的单比特 Adam 算法, 在相同通信开销下能够实现更好的训练效果. 此外, 单比特 Adam-mini 还具有良好的扩展性, 能够有效推广到稀疏压缩器等其他压缩策略.

附录 A 系统实现细节

在算法 1 中, 本文采用服务器与工作节点的描述方式, 然而该算法可以在高性能计算集群中以更高效的方式实现[25]. 在数据并行场景下, 每台机器既充当工作节点参与局部梯度的计算, 也充当服务器节点聚合局部梯度并分发全局梯度. NCCL (Nvidia collective communications library) 是英伟达公司提供的高效集体通信库, 专为多 GPU 并行计算优化, 支持广播、规约等操作. 它与 PyTorch 紧密兼容, 显著提高了大规模深度学习任务的数据传输效率. 在实验部分, 基于 2.27.5 版本的 NCCL 通信库, 本文实现了数据并行场景下与算法 1 等效的算法.

在算法 1 中, 主要的通信开销来源于计算一阶动量或二阶动量的全局平均值. 其中由于单比特 Adam-mini 算法中二阶动量参数较少, 在计算全局平均值时, 采用全量通信的策略 (参考式 (12)). 因此, 在计算 $v_t^{(i,\,b)}$ 的全局平均值时, 本文使用高效的环形 All-Reduce 通信原语来实现上述功能.

在计算一阶动量全局平均值时, 单比特 Adam-mini 算法在通信时引入压缩操作, 参见算法 1 的第 9 ~ 17 行. 在数据并行场景下, 利用 All-to-All 和 All-Gather 通信原语实现了与算法 1 完全等价的通信效果, 具体实现参见算法 A1. 由于误差补偿技巧不会影响通信行为, 为简化描述, 在算法 A1 中省略误差补偿相关的内容.

算法 A1. 数据并行场景下双向压缩等价算法

1: **输入.** 工作节点 i 上的局部张量 $\boldsymbol{x}^{(i)} \in \mathbf{R}^d$.
2: 在工作节点 i 上并行执行:
3: 将 $\boldsymbol{x}^{(i)}$ 划分成 n 份长度为 d/n 的张量 $\boldsymbol{x}_j^{(i)}$ $(j = 1, \cdots, n)$, 对每一份张量压缩得到 $\hat{\boldsymbol{x}}_j^{(i)} = \boldsymbol{C}_\omega[\boldsymbol{x}_j^{(i)}]$ $(j = 1, \cdots, n)$
4: 使用 All-to-All 通信原语, 接收所有工作节点的第 i 份压缩张量 $\hat{\boldsymbol{x}}_i^{(j)}$ $(j = 1, \cdots, n)$
5: 计算全局张量 $\overline{\boldsymbol{x}}_i = \boldsymbol{C}_\omega[\frac{1}{n}\sum_{j=1}^n \hat{\boldsymbol{x}}_i^{(j)}]$
6: 使用 All-Gather 原语, 接收所有工作节点的全局张量 $\overline{\boldsymbol{x}}_j$ $(j = 1, \cdots, n)$
7: 拼接得到完整的全局张量

$$\overline{\boldsymbol{x}} = \text{concat}[\overline{\boldsymbol{x}}_1, \cdots, \overline{\boldsymbol{x}}_n] \in \mathbf{R}^d$$

8: **输出.** $\overline{\boldsymbol{x}}$.

附录 B 证明细节

本部分给出定理 1 的详细证明. 首先考虑 $\overline{\boldsymbol{m}}_t$ 的实际迭代方式. 记 $\boldsymbol{\delta}_t = \frac{1}{n}\sum_{i=1}^n \boldsymbol{\delta}_t^{(i)} + \overline{\boldsymbol{\delta}}_t$, $\boldsymbol{g}_t = \frac{1}{n}\sum_{i=1}^n \boldsymbol{g}_t^{(i)}$, 则

$$\overline{\boldsymbol{m}}_{t+1} = \frac{1}{n}\sum_{i=1}^n \hat{\boldsymbol{m}}_{t+1}^{(i)} + \overline{\boldsymbol{\delta}}_t - \overline{\boldsymbol{\delta}}_{t+1} =$$

$$\frac{1}{n}\sum_{i=1}^n \left(\boldsymbol{m}_{t+1}^{(i)} + \boldsymbol{\delta}_t^{(i)} - \boldsymbol{\delta}_{t+1}^{(i)}\right) + \overline{\boldsymbol{\delta}}_t - \overline{\boldsymbol{\delta}}_{t+1} =$$

$$\frac{1}{n}\sum_{i=1}^n \left(\beta_1 \boldsymbol{m}_t^{(i)} + (1-\beta_1)\boldsymbol{g}_{t+1}^{(i)} + \boldsymbol{\delta}_t^{(i)} - \boldsymbol{\delta}_{t+1}^{(i)}\right) +$$

$$\overline{\boldsymbol{\delta}}_t - \overline{\boldsymbol{\delta}}_{t+1} =$$

$$\beta_1 \overline{\boldsymbol{m}}_t + (1-\beta_1)\boldsymbol{g}_{t+1} + \left(\frac{1}{n}\sum_{i=1}^n \boldsymbol{\delta}_t^{(i)} + \overline{\boldsymbol{\delta}}_t\right) -$$

$$\left(\frac{1}{n}\sum_{i=1}^n \boldsymbol{\delta}_{t+1}^{(i)} + \overline{\boldsymbol{\delta}}_{t+1}\right) =$$

$$\beta_1 \overline{\boldsymbol{m}}_t + (1-\beta_1)\boldsymbol{g}_{t+1} + \boldsymbol{\delta}_t - \boldsymbol{\delta}_{t+1}$$

递归定义 $\Delta_{-1} = 0$, $\Delta_t = \beta_1 \Delta_{t-1} + \boldsymbol{\delta}_t$ $(t \geq 0)$; 递归定义 $\boldsymbol{m}'_{-1} = 0$, $\boldsymbol{m}'_t = \beta_1 \boldsymbol{m}'_{t-1} + (1-\beta_1)\boldsymbol{g}_t$ $(t \geq 0)$ 表示使用真实梯度累积计算的动量. 通过计算可知, $\overline{\boldsymbol{m}}_t - \boldsymbol{m}'_t + \Delta_t - \Delta_{t-1} = 0$. 为方便表述, 将所有的 $\hat{\boldsymbol{v}}_t^{(b)}$ 写为一个向量 $\hat{\boldsymbol{v}}_t$, $\hat{\boldsymbol{v}}_t$ 在 b 分块对应的分量中均为 $\hat{\boldsymbol{v}}_t^{(b)}$. 这样每轮整体参数的更新方式可以写为 $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma \frac{\overline{\boldsymbol{m}}_t}{\sqrt{\hat{\boldsymbol{v}}_t + \epsilon}}$.

为证明定理 1, 接下来需要引入如下引理.

B.1 引理及证明

引理 B1. $\forall t \geq 0$, 以下不等式成立

$$\|\boldsymbol{g}_t\| \leq G, \quad \mathrm{E}\|\boldsymbol{g}_t\|^2 \leq \frac{\sigma^2}{n} + \|\nabla f(\boldsymbol{x}_t)\|^2$$

证明. 对于第一个不等式, 由假设 2 以及 $\boldsymbol{g}_t = \frac{1}{n}\sum_{i=1}^n \boldsymbol{g}_t^{(i)}$ 可以得到 $\|\boldsymbol{g}_t\| \leq G$. 对于第二个不等式, 由假设 2 和假设 3 知

$$\mathrm{E}[\|\boldsymbol{g}_t\|^2] = \mathrm{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{g}_t^{(i)} - \nabla f(\boldsymbol{x}_t) + \nabla f(\boldsymbol{x}_t)\right\|^2\right] =$$

$$\mathrm{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n (\boldsymbol{g}_t^{(i)} - \nabla f_i(\boldsymbol{x}_t))\right\|^2\right] + \|\nabla f(\boldsymbol{x}_t)\|^2 \leq$$

$$\frac{\sigma^2}{n} + \|\nabla f(\boldsymbol{x}_t)\|^2$$

□

引理 B2. $\|\boldsymbol{m}'_t\| \leq G$, $\forall t \geq 0$.

证明. 由假设 2 以及 $\boldsymbol{m}'_t$ 的表达式, 得

$$\|\boldsymbol{m}'_t\| = \|(1-\beta_1)\sum_{i=0}^t \beta_1^{t-i}\boldsymbol{g}_i\| \leq G$$

□

引理 B3. $\boldsymbol{\delta}_t$, Δ_t 如前述定义, 则以下不等式成立:

$$\|\boldsymbol{\delta}_t\|^2 \leq \frac{15q^2}{(1-q)^3(1-r)}G^2$$

$$\|\Delta_t\|^2 \leq \frac{15q^2}{(1-q)^3(1-\beta_1)^2(1-r)}G^2, \ \forall t \geq 0$$

证明. 首先证明如下不等式. 给定向量 $a_i \in \mathbf{R}^d, i = 0, 1, 2, \cdots, k$. 定义 $b_l = (1-s)\sum_{i=0}^l s^{l-i}a_i$, $l = 0, 1, 2, \cdots, k$, 其中, $s \in [0, 1)$. 则对于任意的 $r > s$, 有

$$\sum_{i=0}^k r^{k-i}\|b_i\|^2 \leq \frac{1-s}{1-\frac{s}{r}}\sum_{i=0}^k r^{k-i}\|a_i\|^2 \tag{B1}$$

由 Cauchy 不等式

$$\|b_l\|^2 \leq (1-s)^2\left(\sum_{i=0}^l s^{l-i}\right)\left(\sum_{i=0}^l s^{l-i}\|a_i\|^2\right) \leq$$

$$(1-s)\sum_{i=0}^l s^{l-i}\|a_i\|^2$$

于是应用到原不等式左边

$$\sum_{i=0}^k r^{k-i}\|b_i\|^2 \leq (1-s)\sum_{i=0}^k r^{k-i}\sum_{j=0}^i s^{i-j}\|a_j\|^2 \leq$$

$$(1-s)\sum_{i=0}^k \|a_i\|^2 \sum_{j=i}^k r^{k-j}s^{j-i} =$$

$$(1-s)\sum_{i=0}^k r^{k-i}\|a_i\|^2 \frac{1-(\frac{s}{r})^{k+1-i}}{1-\frac{s}{r}} \leq$$

$$\frac{1-s}{1-\frac{s}{r}}\sum_{i=0}^k r^{k-i}\|a_i\|^2$$

回到原引理. 由假设 4 并且 $\boldsymbol{\delta}_t^{(i)} = \boldsymbol{m}_t^{(i)} + \boldsymbol{\delta}_{t-1}^{(i)} - \boldsymbol{C}_\omega[\boldsymbol{m}_t^{(i)} + \boldsymbol{\delta}_{t-1}^{(i)}], \forall i \in [n]$, 得到

$$\|\boldsymbol{\delta}_t^{(i)}\|^2 \leq q^2\|\boldsymbol{m}_t^{(i)} + \boldsymbol{\delta}_{t-1}^{(i)}\|^2 =$$

$$q^2\|\beta_1\overline{\boldsymbol{m}}_{t-1} + (1-\beta_1)\boldsymbol{g}_t^{(i)} + \boldsymbol{\delta}_{t-1}^{(i)}\|^2 =$$

$$q^2\|\beta_1(\overline{\boldsymbol{m}}_{t-1} - \boldsymbol{m}'_{t-1}) + \boldsymbol{\delta}_{t-1}^{(i)} + \beta_1\boldsymbol{m}'_{t-1} +$$

$$(1-\beta_1)\boldsymbol{g}_t^{(i)}\|^2 =$$

$$q^2\|(1-\beta_1)\Delta_{t-1} - \boldsymbol{\delta}_{t-1} + \boldsymbol{\delta}_{t-1}^{(i)} + \beta_1\boldsymbol{m}'_{t-1} +$$

$$(1-\beta_1)\boldsymbol{g}_t^{(i)}\|^2 \overset{(a)}{\leq}$$

$$q\|(1-\beta_1)\Delta_{t-1} - \boldsymbol{\delta}_{t-1} + \boldsymbol{\delta}_{t-1}^{(i)}\|^2 +$$

$$\frac{q^2}{1-q}\|\beta_1\boldsymbol{m}'_{t-1} + (1-\beta_1)\boldsymbol{g}_t^{(i)}\|^2 \leq$$

$$q\|(1-\beta_1)\Delta_{t-1} - \boldsymbol{\delta}_{t-1} + \boldsymbol{\delta}_{t-1}^{(i)}\|^2 + \frac{q^2}{1-q}G^2 \tag{B2}$$

其中, $\overset{(a)}{\leq}$ 用到不等式 $\|a+b\|^2 \leq \frac{1}{q}\|a\|^2 + \frac{1}{1-q}\|b\|^2$. 将式 (B2) 对全体 $i \in [n]$ 相加并除以 n. 记 $M_t = \frac{1}{n}\sum_{i=1}^n \|\boldsymbol{\delta}_t^{(i)}\|^2$, $\boldsymbol{\delta}_t^* = \sum_{i=1}^n \boldsymbol{\delta}_t^{(i)}$, 则有

$$
\begin{aligned}
M_t = \frac{1}{n}\sum_{i=1}^n \|\boldsymbol{\delta}_t^{(i)}\|^2 \leq q\Bigg(\frac{1}{n}\sum_{i=1}^n \|(1-\beta_1)\Delta_{t-1} - \\
\overline{\boldsymbol{\delta}}_{t-1} - \boldsymbol{\delta}_{t-1}^* + \boldsymbol{\delta}_{t-1}^{(i)}\|^2\Bigg) + \frac{q^2}{1-q}G^2 = \\
q(\|(1-\beta_1)\Delta_{t-1} - \overline{\boldsymbol{\delta}}_{t-1}\|^2 - \|\boldsymbol{\delta}_{t-1}^*\|^2 + \\
\frac{1}{n}\sum_{i=1}^n \|\boldsymbol{\delta}_{t-1}^{(i)}\|^2) + \frac{q^2}{1-q}G^2 = \\
q(\beta_1\|\overline{\boldsymbol{\delta}}_{t-1}\|^2 - \beta_1(1-\beta_1)\|\Delta_{t-1}\|^2 + \\
(1-\beta_1)\|\Delta_{t-1} - \overline{\boldsymbol{\delta}}_{t-1}\|^2 - \\
\|\boldsymbol{\delta}_{t-1}^*\|^2 + M_{t-1}) + \frac{q^2}{1-q}G^2 \qquad \text{(B3)}
\end{aligned}
$$

由假设 4 以及 $\overline{\boldsymbol{\delta}}_t = \frac{1}{n}\sum_{i=1}^n \hat{\boldsymbol{m}}_t^{(i)} + \overline{\boldsymbol{\delta}}_{t-1} - \boldsymbol{C}_\omega \times [\frac{1}{n}\sum_{i=1}^n \hat{\boldsymbol{m}}_t^{(i)} + \overline{\boldsymbol{\delta}}_{t-1}]$, 有

$$
\begin{aligned}
\|\overline{\boldsymbol{\delta}}_t\|^2 \leq q^2\left\|\frac{1}{n}\sum_{i=1}^n \hat{\boldsymbol{m}}_t + \overline{\boldsymbol{\delta}}_{t-1}\right\|^2 = \\
q^2\|\overline{\boldsymbol{m}}_t + \boldsymbol{\delta}_{t-1}^* - \boldsymbol{\delta}_t^* + \overline{\boldsymbol{\delta}}_{t-1}\|^2 = \\
q^2\|(1-\beta_1)\Delta_{t-1} - \boldsymbol{\delta}_t^* + \beta_1\boldsymbol{m}_{t-1}' + \\
(1-\beta_1)\boldsymbol{g}_t\|^2 \leq \\
q\|(1-\beta_1)\Delta_{t-1} - \boldsymbol{\delta}_t^*\|^2 + \frac{q^2}{1-q}\|\beta_1\boldsymbol{m}_{t-1}' + \\
(1-\beta_1)\boldsymbol{g}_t\|^2 \leq \\
q\|(1-\beta_1)\Delta_{t-1} - \boldsymbol{\delta}_t^*\|^2 + \frac{q^2}{1-q}G^2 = \\
q\Bigg(\frac{1}{\beta_1}\|\boldsymbol{\delta}_t^*\|^2 + (1-\beta_1)\|\Delta_{t-1}\|^2 - \\
\frac{1-\beta_1}{\beta_1}\|\beta_1\Delta_{t-1} + \boldsymbol{\delta}_t^*\|^2\Bigg) + \frac{q^2}{1-q}G^2 = \\
q\Bigg(\frac{1}{\beta_1}\|\boldsymbol{\delta}_t^*\|^2 + (1-\beta_1)\|\Delta_{t-1}\|^2 - \\
\frac{1-\beta_1}{\beta_1}\|\Delta_t - \overline{\boldsymbol{\delta}}_t\|^2\Bigg) + \frac{q^2}{1-q}G^2 \qquad \text{(B4)}
\end{aligned}
$$

利用式 (B3) 和式 (B4), 得到

$$
\begin{aligned}
M_t + \beta_1\|\overline{\boldsymbol{\delta}}_{t-1}\|^2 \leq q(\beta_1\|\overline{\boldsymbol{\delta}}_{t-1}\|^2 + \\
(1-\beta_1)\beta_1(\|\Delta_{t-2}\|^2 - \|\Delta_{t-1}\|^2) + \\
M_{t-1}) + \frac{q^2}{1-q}(1+\beta_1)G^2 \qquad \text{(B5)}
\end{aligned}
$$

对 $t = 0, 1, 2, \cdots, l$, 式 (B5) 乘上系数 r^{l-t} 进行求和 (所有下标小于 -1 的变量均为 0), 即

$$
\begin{aligned}
\sum_{t=0}^l r^{l-t}(M_t + \beta_1\|\overline{\boldsymbol{\delta}}_{t-1}\|^2) \leq \\
q\Bigg(\beta_1\sum_{t=0}^{l-1} r^{l-t-1}\|\overline{\boldsymbol{\delta}}_{t-1}\|^2 + \\
(1-\beta_1)\beta_1\left(\frac{1}{r}-1\right)\sum_{t=0}^{l-1} r^{l-t-1}\|\Delta_{t-1}\|^2 + \\
\sum_{t=0}^{l-1} r^{l-t-1}M_t\Bigg) + \frac{q^2}{1-q}(1+\beta_1)G^2\frac{1}{1-r}
\end{aligned}
$$

于是有

$$
\begin{aligned}
M_l \leq \frac{q^2(1+\beta_1)}{(1-q)(1-r)}G^2 - \Bigg((r-q)\sum_{t=0}^{l-1} r^{l-t-1}M_t + \\
\beta_1(1-q)\sum_{t=0}^{l-1} r^{l-t-1}\|\overline{\boldsymbol{\delta}}_{t-1}\|^2 - \\
(1-\beta_1)\beta_1\left(\frac{1}{r}-1\right)\sum_{t=0}^{l-1} r^{l-t-1}\|\Delta_{t-1}\|^2\Bigg)
\end{aligned}
$$

注意到

$$
\begin{aligned}
\sum_{t=0}^{l-1} r^{l-t-1}((r-q)M_t + \beta_1(1-q)\|\overline{\boldsymbol{\delta}}_t\|^2) \geq \\
\sum_{t=0}^{l-1} r^{l-t-1}\frac{3}{4}\beta_1(1-q)(M_t + \|\overline{\boldsymbol{\delta}}_t\|^2) \geq \\
\sum_{t=0}^{l-1} r^{l-t-1}\frac{3}{4}\beta_1(1-q)\frac{1}{2}\|\boldsymbol{\delta}_t\|^2 \overset{(a)}{\geq} \\
\frac{3}{8}\beta_1(1-q)\frac{1-\frac{\beta_1}{r}}{1-\beta_1}\sum_{t=0}^{l-1} r^{l-t-1}\|\Delta_t\|^2 \overset{(b)}{\geq} \\
(1-\beta_1)\beta_1\left(\frac{1}{r}-1\right)\sum_{t=0}^{l-1} r^{l-t-1}\|\Delta_t\|^2
\end{aligned}
$$

其中, $\overset{(a)}{\geq}$ 用到引理最开始证明的式 (B1); $\overset{(b)}{\geq}$ 是因为由 $r \geq (\beta_1+1)/2$, $r \geq (q+3)/4$ 得到 $(1-\frac{\beta_1}{r})/(1-\beta_1) \geq 1-\beta_1$, $(r-q)/2 \geq \frac{1}{r}-1$, 从而得到

$$
M_t \leq \frac{q^2(1+\beta_1)}{(1-q)(1-r)}G^2 \leq \frac{2q^2}{(1-q)(1-r)}G^2
$$

于是, $\|\boldsymbol{\delta}_t^*\|^2 \leq M_t \leq \frac{2q^2}{(1-q)(1-r)}G^2$. 进一步

$$\|\overline{\boldsymbol{\delta}}_t\| \leq q\|(1-\beta_1)\Delta_{t-1} - \boldsymbol{\delta}_t^* + \beta_1\boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t\| \leq q((1-\beta_1)\sum_{i=0}^{t-1}\beta_1^{t-i-1}(\|\overline{\boldsymbol{\delta}}_i\| + \|\boldsymbol{\delta}_i^*\|) + \|\beta_1\boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t\|) \leq q(1-\beta_1)\sum_{i=0}^{t-1}\beta_1^{t-i-1}\|\overline{\boldsymbol{\delta}}_i\| + q\sqrt{\frac{2q^2}{(1-q)(1-r)}}G + qG \leq q(1-\beta_1)\sum_{i=0}^{t-1}\beta_1^{t-i-1}\|\overline{\boldsymbol{\delta}}_i\| + q\sqrt{\frac{6}{(1-q)(1-r)}}G$$

使用归纳可以得到

$$\|\overline{\boldsymbol{\delta}}_t\|^2 \leq \frac{6q^2}{(1-q)^3(1-r)}G^2$$

因此, 得到如下结果:

$$\|\boldsymbol{\delta}_t\|^2 \leq (\|\boldsymbol{\delta}_t^*\| + \|\overline{\boldsymbol{\delta}}_t\|)^2 \leq \left(\sqrt{\frac{6q^2}{(1-q)^3(1-r)}} + \sqrt{\frac{2q^2}{(1-q)(1-r)}}\right)^2 G^2 \leq \frac{15q^2}{(1-q)^3(1-r)}G^2$$

注意到 $\Delta_t = \sum_{i=0}^t \beta_1^{t-i}\boldsymbol{\delta}_i$, 进一步可以得到 $\|\Delta_t\|^2 \leq \frac{15q^2}{(1-q)^3(1-\beta_1)^2(1-r)}G^2$. □

引理 B4. $\hat{\boldsymbol{v}}_{t,\,i} \leq G^2$, $\forall t \geq 0$.

证明. 由式 (11) $v_t^{(i,\,b)}$ 的更新方式以及假设 2 知

$$v_t^{(i,\,b)} = \frac{\|\boldsymbol{g}_t^{(i,\,b)}\|^2}{d_b} \leq G^2$$

其中, d_b 表示分块后的 $\boldsymbol{g}_t^{(i,\,b)}$ 的维数.

于是由式 (12) 知

$$\overline{v}_t^{(b)} = \frac{1-\beta_2}{n}\sum_{i=1}^n v_t^{(i,\,b)} + \beta_2\overline{v}_{t-1}^{(b)} \leq (1-\beta_2)G^2 + \beta_2\overline{v}_{t-1}^{(b)}$$

利用归纳可以得到 $\overline{v}_t^{(b)} \leq G^2$. 再根据 $\hat{\boldsymbol{v}}_t$ 的更新方式有 $\hat{\boldsymbol{v}}_{t,\,i} \leq G^2$. □

引理 B5. 定义 $D_t = \frac{1}{\sqrt{\hat{\boldsymbol{v}}_{t-1}+\epsilon}} - \frac{1}{\sqrt{\hat{\boldsymbol{v}}_t+\epsilon}}$, 则

$$\sum_{t=0}^T \|D_t\|_1 \leq \frac{d}{\epsilon}, \quad \sum_{t=0}^T \|D_t\|^2 \leq \frac{d}{\epsilon^2}$$

证明. 根据 $\hat{\boldsymbol{v}}_t$ 的更新方式有 $\hat{\boldsymbol{v}}_t \leq \hat{\boldsymbol{v}}_{t-1}$, $\forall t$. 于是

$$\sum_{t=0}^T \|D_t\|_1 = \sum_{t=0}^T\sum_{i=1}^d \left(\frac{1}{\sqrt{\hat{\boldsymbol{v}}_{t-1,\,i}+\epsilon}} - \frac{1}{\sqrt{\hat{\boldsymbol{v}}_{t,\,i}+\epsilon}}\right) = \sum_{i=1}^d \left(\frac{1}{\sqrt{\hat{\boldsymbol{v}}_{0,\,i}+\epsilon}} - \frac{1}{\sqrt{\hat{\boldsymbol{v}}_{T,\,i}+\epsilon}}\right) \leq \frac{d}{\epsilon}$$

$$\sum_{t=0}^T \|D_t\|^2 = \sum_{t=0}^T\sum_{i=1}^d \left(\frac{1}{\sqrt{\hat{\boldsymbol{v}}_{t-1,\,i}+\epsilon}} - \frac{1}{\sqrt{\hat{\boldsymbol{v}}_{t,\,i}+\epsilon}}\right)^2 \leq \sum_{t=0}^T\sum_{i=1}^d \frac{1}{(\sqrt{\hat{\boldsymbol{v}}_{t-1,\,i}+\epsilon})^2} - \frac{1}{(\sqrt{\hat{\boldsymbol{v}}_{t,\,i}+\epsilon})^2} = \sum_{i=1}^d \left(\frac{1}{(\sqrt{\boldsymbol{v}_{0,\,i}+\epsilon})^2} - \frac{1}{(\sqrt{\boldsymbol{v}_{T,\,i}+\epsilon})^2}\right) \leq \frac{d}{\epsilon^2}$$

□

B.2　定理 1 的证明

定义 $\boldsymbol{y}_t = \boldsymbol{x}_t - \gamma\frac{\beta_1}{1-\beta_1}\frac{\boldsymbol{m}'_{t-1}}{\sqrt{\hat{\boldsymbol{v}}_{t-1}+\epsilon}} - \gamma\frac{\Delta_{t-1}}{\sqrt{\hat{\boldsymbol{v}}_{t-1}+\epsilon}}$. 下面考虑 $\boldsymbol{y}_t$ 的迭代.

$$\boldsymbol{y}_{t+1} = \boldsymbol{x}_{t+1} - \gamma\frac{\beta_1}{1-\beta_1}\frac{\boldsymbol{m}'_t}{\sqrt{\hat{\boldsymbol{v}}_t+\epsilon}} - \gamma\frac{\Delta_t}{\sqrt{\hat{\boldsymbol{v}}_t+\epsilon}}$$

$$\boldsymbol{y}_t = \boldsymbol{x}_t - \gamma\frac{\beta_1}{1-\beta_1}\frac{\boldsymbol{m}'_{t-1}}{\sqrt{\hat{\boldsymbol{v}}_{t-1}+\epsilon}} - \gamma\frac{\Delta_{t-1}}{\sqrt{\hat{\boldsymbol{v}}_{t-1}+\epsilon}}$$

相减后得到

$$\begin{aligned}\boldsymbol{y}_{t+1} - \boldsymbol{y}_t = {}& \boldsymbol{x}_{t+1} - \boldsymbol{x}_t - \gamma\frac{\beta_1}{1-\beta_1}\frac{\boldsymbol{m}'_t - \boldsymbol{m}'_{t-1}}{\sqrt{\hat{\boldsymbol{v}}_t+\epsilon}} - \gamma\frac{\Delta_t - \Delta_{t-1}}{\sqrt{\hat{\boldsymbol{v}}_t+\epsilon}} + \gamma\frac{\beta_1}{1-\beta_1}D_t\boldsymbol{m}'_{t-1} + \gamma D_t\Delta_{t-1} = \\ & -\left(\gamma\frac{\boldsymbol{m}_t - \boldsymbol{m}'_t}{\sqrt{\hat{\boldsymbol{v}}_t+\epsilon}} + \gamma\frac{\Delta_t - \Delta_{t-1}}{\sqrt{\hat{\boldsymbol{v}}_t+\epsilon}}\right) - \left(\gamma\frac{\boldsymbol{m}'_t}{\sqrt{\hat{\boldsymbol{v}}_t+\epsilon}} + \gamma\frac{\beta_1}{1-\beta_1}\frac{\boldsymbol{m}'_t - \boldsymbol{m}'_{t-1}}{\sqrt{\hat{\boldsymbol{v}}_t+\epsilon}}\right) + \\ & \gamma\frac{\beta_1}{1-\beta_1}D_t\boldsymbol{m}'_{t-1} + \gamma D_t\Delta_{t-1} = \\ & -\gamma\frac{\boldsymbol{g}_t}{\sqrt{\hat{\boldsymbol{v}}_t+\epsilon}} + \gamma\frac{\beta_1}{1-\beta_1}D_t\boldsymbol{m}'_{t-1} + \gamma D_t\Delta_{t-1}\end{aligned}$$

应用假设 1 函数 f 的光滑性, 有

$$f(\boldsymbol{y}_{t+1}) \leq f(\boldsymbol{y}_t) + \langle\nabla f(\boldsymbol{y}_t),\, \boldsymbol{y}_{t+1} - \boldsymbol{y}_t\rangle + \frac{L}{2}\|\boldsymbol{y}_{t+1} - \boldsymbol{y}_t\|^2$$

对 $\boldsymbol{g}_t$ 的随机性取期望, 得到

$$\mathrm{E}[f(\boldsymbol{y}_{t+1})]-f(\boldsymbol{y}_t)\leq \underbrace{-\gamma\mathrm{E}[\langle\nabla f(\boldsymbol{y}_t),\ \frac{\boldsymbol{g}_t}{\sqrt{\hat{\boldsymbol{v}}_t}+\epsilon}\rangle]}_{\mathrm{I}}+\underbrace{\gamma\mathrm{E}[\langle\nabla f(\boldsymbol{y}_t),\ \frac{\beta_1}{1-\beta_1}D_t\boldsymbol{m}'_{t-1}+D_t\Delta_{t-1}\rangle]}_{\mathrm{II}}+\underbrace{\frac{\gamma^2L}{2}\mathrm{E}[\|\frac{\boldsymbol{g}_t}{\sqrt{\hat{\boldsymbol{v}}_t}+\epsilon}-\frac{\beta_1}{1-\beta_1}D_t\boldsymbol{m}'_{t-1}-D_t\Delta_{t-1}\|^2]}_{\mathrm{III}}$$

接下来, 分别对 I, II, III 进行放缩. 回顾 $C_1=\frac{\beta_1}{1-\beta_1}+\sqrt{\frac{15q^2}{(1-q)^3(1-\beta_1)^2(1-r)}}$. 利用引理 B2 和引理 B3, 可以得到

$$\|\frac{\beta_1}{1-\beta_1}\boldsymbol{m}'_{t-1}+\Delta_{t-1}\|\leq\frac{\beta_1}{1-\beta_1}G+\sqrt{\frac{15q^2}{(1-q)^3(1-\beta_1)^2(1-r)}}G=C_1G \quad \text{(B6)}$$

利用式 (B6), 可以得到

$$\|\nabla f(\boldsymbol{y}_t)-\nabla f(\boldsymbol{x}_t)\|\leq L\|\boldsymbol{y}_t-\boldsymbol{x}_t\|=\gamma L\|\frac{\beta_1}{1-\beta_1}\frac{\boldsymbol{m}'_{t-1}}{\sqrt{\hat{\boldsymbol{v}}_{t-1}}+\epsilon}+\frac{\Delta_{t-1}}{\sqrt{\hat{\boldsymbol{v}}_{t-1}}+\epsilon}\|\leq\frac{\gamma C_1L}{\epsilon}G \quad \text{(B7)}$$

对 I 的放缩过程为

$$\begin{aligned}\mathrm{I}=&-\gamma\mathrm{E}\left[\left\langle\nabla f(\boldsymbol{y}_t),\ \frac{\boldsymbol{g}_t}{\sqrt{\hat{\boldsymbol{v}}_{t-1}}+\epsilon}\right\rangle\right]+\gamma\mathrm{E}[\langle\nabla f(\boldsymbol{y}_t)-\nabla f(\boldsymbol{x}_t),\ D_t\boldsymbol{g}_t\rangle]+\gamma\mathrm{E}[\langle\nabla f(\boldsymbol{x}_t),\ D_t\boldsymbol{g}_t\rangle]\overset{(a)}{\leq}\\&-\gamma\mathrm{E}\left[\left\langle\nabla f(\boldsymbol{y}_t),\ \frac{\nabla f(\boldsymbol{x}_t)}{\sqrt{\hat{\boldsymbol{v}}_{t-1}}+\epsilon}\right\rangle\right]+\frac{\gamma^2C_1LG^2}{\epsilon}\mathrm{E}[\|D_t\|_1]+\gamma G^2\mathrm{E}[\|D_t\|_1]=\\&-\frac{\gamma}{2}\mathrm{E}\left[\left\langle\nabla f(\boldsymbol{x}_t),\ \frac{\nabla f(\boldsymbol{x}_t)}{\sqrt{\hat{\boldsymbol{v}}_{t-1}}+\epsilon}\right\rangle\right]-\frac{\gamma}{2}\mathrm{E}\left[\left\langle\nabla f(\boldsymbol{y}_t),\ \frac{\nabla f(\boldsymbol{y}_t)}{\sqrt{\hat{\boldsymbol{v}}_{t-1}}+\epsilon}\right\rangle\right]+\frac{\gamma}{2}\mathrm{E}\left[\left\langle\nabla f(\boldsymbol{x}_t)-\nabla f(\boldsymbol{y}_t),\ \frac{\nabla f(\boldsymbol{x}_t)-\nabla f(\boldsymbol{y}_t)}{\sqrt{\hat{\boldsymbol{v}}_{t-1}}+\epsilon}\right\rangle\right]+\frac{\gamma^2C_1LG^2}{\epsilon}\mathrm{E}[\|D_t\|_1]+\\&\gamma G^2\mathrm{E}[\|D_t\|_1]\overset{(b)}{\leq}-\frac{\gamma}{2C_0}\|\nabla f(\boldsymbol{x}_t)\|^2+\frac{\gamma^3C_1^2L^2G^2}{2\epsilon^3}+\frac{\gamma^2C_1LG^2}{\epsilon}\mathrm{E}[\|D_t\|_1]+\gamma G^2\mathrm{E}[\|D_t\|_1]\end{aligned} \quad \text{(B8)}$$

其中, $\overset{(a)}{\leq}$ 用到式 (B7), 以及不等式 $\langle\boldsymbol{a},\boldsymbol{b}\odot\boldsymbol{c}\rangle\overset{(b)}{\leq}\|\boldsymbol{a}\|\|\boldsymbol{c}\|\|\boldsymbol{b}\|_1$ (向量 $\boldsymbol{b}\odot\boldsymbol{c}$ 表示向量 $\boldsymbol{b}$, $\boldsymbol{c}$ 逐点相乘), $\leq$ 用到式 (B7) 以及引理 B4.

对 II 的放缩过程为

$$\mathrm{II}=\gamma\mathrm{E}[\langle\nabla f(\boldsymbol{y}_t)-\nabla f(\boldsymbol{x}_t),\ \frac{\beta_1}{1-\beta_1}D_t\boldsymbol{m}'_{t-1}+D_t\Delta_t\rangle]+\gamma\mathrm{E}[\langle\nabla f(\boldsymbol{x}_t),\ \frac{\beta_1}{1-\beta_1}D_t\boldsymbol{m}'_{t-1}+D_t\Delta_t\rangle]\overset{(a)}{\leq}\frac{\gamma^2C_1^2LG^2}{\epsilon}\mathrm{E}[\|D_t\|_1]+\gamma C_1G^2\mathrm{E}[\|D_t\|_1] \quad \text{(B9)}$$

其中, $\overset{(a)}{\leq}$ 用到式 (B6) 和式 (B7), 以及不等式 $\langle\boldsymbol{a},\boldsymbol{b}\odot\boldsymbol{c}\rangle\leq\|\boldsymbol{a}\|\|\boldsymbol{c}\|\|\boldsymbol{b}\|_1$.

对 III 的放缩过程为

$$\begin{aligned}\mathrm{III}\overset{(a)}{\leq}&\gamma^2L\mathrm{E}\left[\left\|\frac{\boldsymbol{g}_t}{\sqrt{\hat{\boldsymbol{v}}_t}+\epsilon}\right\|^2\right]+\gamma^2L\mathrm{E}\left[\left\|\frac{\beta_1}{1-\beta_1}D_t\boldsymbol{m}'_{t-1}+D_t\Delta_t\right\|^2\right]\overset{(b)}{\leq}\\&\frac{\gamma^2L}{\epsilon^2}\|\nabla f(\boldsymbol{x}_t)\|^2+\frac{\gamma^2L\sigma^2}{n\epsilon^2}+\gamma^2C_1^2LG^2\mathrm{E}[\|D_t\|^2]\end{aligned} \quad \text{(B10)}$$

其中, $\overset{(a)}{\leq}$ 用到 $\frac{1}{2}\|\boldsymbol{a}+\boldsymbol{b}\|^2\leq\|\boldsymbol{a}\|^2+\|\boldsymbol{b}\|^2$; $\overset{(b)}{\leq}$ 用到引理 B1, 式 (B6) 以及 $\|\boldsymbol{a}\odot\boldsymbol{b}\|^2\leq\|\boldsymbol{a}\|^2\|\boldsymbol{b}\|^2$ (向量 $\boldsymbol{a}\odot\boldsymbol{b}$ 表示向量 $\boldsymbol{a}$, $\boldsymbol{b}$ 逐点相乘).

将式 (B8) ~ (B10) 进行整合, 得到

$$\begin{aligned}\mathrm{E}[f(\boldsymbol{y}_{t+1})]-f(\boldsymbol{y}_t)\leq&\left(-\frac{\gamma}{2C_0}+\frac{\gamma^2L}{\epsilon^2}\right)\|\nabla f(\boldsymbol{x}_t)\|^2+\frac{\gamma^3C_1^2L^2G^2}{2\epsilon^3}+\frac{\gamma^2L\sigma^2}{n\epsilon^2}+\\&\left(\frac{\gamma^2C_1(C_1+1)LG^2}{\epsilon}+\gamma(C_1+1)G^2\right)\mathrm{E}[\|D_t\|_1]+\\&\gamma^2C_1^2LG^2\mathrm{E}[\|D_t\|^2]\end{aligned}$$

接下来, 对 $t=0,\ 1,\ 2,\ \cdots,\ T$ 求和并取期望, 得

$$\begin{aligned}&\mathrm{E}[f(\boldsymbol{y}_{T+1})-f(\boldsymbol{y}_0)]\leq\\&\left(-\frac{\gamma}{2C_0}+\frac{\gamma^2L}{\epsilon^2}\right)\sum_{t=0}^{T}\mathrm{E}[\|\nabla f(\boldsymbol{x}_t)\|^2]+\\&\frac{(T+1)\gamma^3C_1^2L^2G^2}{2\epsilon^3}+\frac{(T+1)\gamma^2L\sigma^2}{n\epsilon^2}+\\&\left(\frac{\gamma^2C_1(C_1+1)LG^2}{\epsilon}+\gamma(C_1+1)G^2\right)\sum_{t=0}^{T}\mathrm{E}[\|D_t\|_1]+\end{aligned}$$

$$\gamma^2 C_1^2 L G^2 \sum_{t=0}^{T} \mathrm{E}[\|D_t\|^2] \overset{(a)}{\leq}$$

$$\left(-\frac{\gamma}{2C_0}+\frac{\gamma^2 L}{\epsilon^2}\right)\sum_{t=0}^{T}\mathrm{E}[\|\nabla f(\boldsymbol{x}_t)\|^2]+$$

$$\frac{(T+1)\gamma^3 C_1^2 L^2 G^2}{2\epsilon^3}+\frac{(T+1)\gamma^2 L\sigma^2}{n\epsilon^2}+$$

$$\frac{\gamma^2 C_1(2C_1+1)LG^2 d}{\epsilon^2}+\frac{\gamma(C_1+1)G^2 d}{\epsilon}$$

其中, $\overset{(a)}{\leq}$ 用到引理 B5.

整理后, 得到如下结果:

$$\left(1-\frac{2\gamma C_0 L}{\epsilon^2}\right)\frac{1}{T+1}\sum_{t=0}^{T}\mathrm{E}[\|\nabla f(\boldsymbol{x}_t)\|^2]\leq$$

$$2C_0\left(\frac{\mathrm{E}[f(\boldsymbol{y}_0)-f(\boldsymbol{y}_{T+1})]}{(T+1)\gamma}+\frac{\gamma^2 C_1^2 L^2 G^2}{2\epsilon^3}+\frac{\gamma L\sigma^2}{n\epsilon^2}+\right.$$

$$\left.\frac{\gamma C_1(2C_1+1)LG^2 d}{(T+1)\epsilon^2}+\frac{(C_1+1)G^2 d}{(T+1)\epsilon}\right)\leq$$

$$2C_0\left(\frac{\mathrm{E}[f(\boldsymbol{x}_0)]-f^*}{(T+1)\gamma}+\frac{\gamma^2 C_1^2 L^2 G^2}{2\epsilon^3}+\frac{\gamma L\sigma^2}{n\epsilon^2}+\right.$$

$$\left.\frac{\gamma C_1(2C_1+1)LG^2 d}{(T+1)\epsilon^2}+\frac{(C_1+1)G^2 d}{(T+1)\epsilon}\right)$$

□

附录 C 补充实验

在收敛性分析部分, 参考 AMSGrad 算法[63] 对单比特 Adam-mini 算法进行改进. 具体而言, 将算法 1 第 19 行中更新的二阶动量从 $\overline{v}_t^{(b)}$ 替换为 $\max\{\hat{v}_{t-1}^{(b)}, \overline{v}_t^{(b)}\}$, 这一修改旨在使 Adam 类算法能够进行收敛性分析.

为验证上述改进的实际实验效果, 本文基于第 6.5 节的实验设置实现改进算法. 实验结果如表 C1 所示, 表 C1 中使用后缀 w/AMS 表示采用 AMSGrad 改进的算法, 后缀 wo/AMS 表示原始版本.

表 C1　在 SQuAD 数据集上微调 BERT-large 模型的测试准确率

Table C1　Test accuracy of finetuned BERT-large models on SQuAD dataset

优化算法	1-bit Adam-mini	
是否使用 AMS 技巧	w/AMS	wo/AMS
预热步数 = 20	85.76	**86.51**
预热步数 = 100	86.32	**86.71**
预热步数 = $+\infty$	**86.49**	86.45

注: 加粗字体表示最优结果.

如表 C1 所示, 在不同预热步数配置下, 采用 AMS 改进的算法 (w/AMS) 与原始算法 (wo/AMS) 的实验效果未呈现显著差异. 值得注意的是, 原始算法在某些配置下 (如预热步数 = 20, 100) 反而表现出更优的性能. 基于上述结果, 第 6 节的实验部分均选择实际效果更好的 wo/AMS 版本作为最终实验方案.

参考文献

1 Grattafiori A, Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, et al. The LLaMA 3 herd of models. arXiv preprint arXiv: 2407.21783, 2024.

2 Mesnard T, Hardin C, Dadashi R, Bhupatiraju S, Pathak S, Sifre L, et al. Gemma: Open models based on gemini research and technology. arXiv preprint arXiv: 2403.08295, 2024.

3 Liu H, Li C, Wu Q, Lee Y J. Visual instruction tuning. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans, USA: Curran Associates, Inc. 2023. 34892−34916

4 Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman F L, et al. GPT-4 Technical Report. arXiv preprint arXiv: 2303.08774, 2023.

5 Liu A, Feng B, Xue B, Wang B, Wu B, Lu C, et al. DeepSeek-V3 Technical Report. arXiv preprint arXiv: 2412.19437, 2024.

6 Anil R, Borgeaud S, Alayrac J B, Yu J, Soricut R, Schalkwyk J, et al. Gemini: A family of highly capable multimodal models. arXiv preprint arXiv: 2312.11805, 2023.

7 Shoeybi M, Patwary M, Puri R, LeGresley P, Casper J, Catanzaro B. Megatron-LM: Training multi-billion parameter language models using model parallelism. arXiv preprint arXiv: 1909.08053, 2019.

8 Narayanan D, Shoeybi M, Casper J, LeGresley P, Patwary M, Korthikanti V, et al. Efficient large-scale language model training on GPU clusters using Megatron-LM. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. Saint Louis, USA: ACM. 2021. 1−15

9 Brown T, Mann B, Ryder N, Subbiah M, Kaplan J D, Dhariwal P, et al. Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. Vancouver, Canada: Curran Associates, Inc. 2020. 1877−1901

10 Zhang S, Roller S, Goyal N, Artetxe M, Chen M, Chen S, et al. OPT: Open pre-trained transformer language models. arXiv preprint arXiv: 2205.01068, 2022.

11 Alistarh D, Grubic D, Li J Z, Tomioka R, Vojnovic M. QSGD: Communication-efficient SGD via gradient quantization and encoding. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, USA: Curran Associates, Inc. 2017. 1707−1718

12 Bernstein J, Wang Y X, Azizzadenesheli K, Anandkumar A. SignSGD: Compressed optimisation for non-convex problems. In: Proceedings of the 35th International Conference on Machine Learning. Stockholm, Sweden: PMLR, 2018. 560−569

13 Stich S U, Cordonnier J B, Jaggi M. Sparsified SGD with memory. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal, Canada: Curran Associates, Inc. 2018. 4452−4463

14 Richtárik P, Sokolov I, Fatkhullin I. EF21: A new, simpler, theoretically better, and practically faster error feedback. In: Pro-

ceedings of the 35th International Conference on Neural Information Processing Systems. Virtual Event: Curran Associates, Inc. 2021. 4384−4396

15 Huang X, Chen Y, Yin W, Yuan K. Lower bounds and nearly optimal algorithms in distributed learning with communication compression. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. New Orleans, USA: Curran Associates, Inc. 2022. 18955−18969

16 Ramasinghe S, Ajanthan T, Avraham G, Zuo Y, Long A. Protocal models: Scaling decentralized training with communication-efficient model parallelism. arXiv preprint arXiv: 2506.01260, 2025.

17 Horvóth S, Ho C Y, Horvath L, Sahu A N, Canini M, Richtárik P. Natural compression for distributed deep learning. *Mathematical and Scientific Machine Learning*. Beijing, China: PMLR, 2022. 129−141

18 Seide F, Fu H, Droppo J, Li G, Yu D. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In: Proceedings of Interspeech. Singapore: ISCA. 2014. 1058−1062

19 Li S, Xu W, Wang H, Tang X, Qi Y, Xu S, et al. FedBAT: Communication-efficient federated learning via learnable binarization. In: Proceedings of the 41st International Conference on Machine Learning. Vienna, Austria: PMLR, 2024. 29074−29095

20 Wangni J, Wang J, Liu J, Zhang T. Gradient sparsification for communication-efficient distributed optimization. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal, Canada: Curran Associates, Inc. 2018. 1306−1316

21 Kingma D P, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv: 1412.6980, 2014.

22 Fatkhullin I, Tyurin A, Richtárik P. Momentum provably improves error feedback! In: Proceedings of the 38th International Conference on Neural Information Processing Systems. Vancouver, Canada: Curran Associates, Inc. 2024. 76444−76495

23 Chen C, Shen L, Huang H, Liu W. Quantized Adam with error feedback. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2021, **12**(5): 1−26

24 Chen C, Shen L, Liu W, Luo Z Q. Efficient-Adam: Communication-efficient distributed Adam. *IEEE Transactions on Signal Processing*, 2023, **71**: 3257−3266

25 Tang H, Gan S, Awan A A, Rajbhandari S, Li C, Lian X, et al. 1-bit Adam: Communication efficient large-scale training with Adam's convergence speed. In: Proceedings of the 38th International Conference on Machine Learning. Virtual Event: PMLR, 2021. 10118−10129

26 Zhang Y, Chen C, Li Z, Ding T, Wu C, Kingma D P, et al. Adam-mini: Use fewer learning rates to gain more. arXiv preprint arXiv: 2406.16793, 2024.

27 Tang H, Yu C, Lian X, Zhang T, Liu J. DoubleSqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. In: Proceedings of the 36th International Conference on Machine Learning. Long Beach, USA: PMLR, 2019. 6155−6165

28 Li X, Karimi B, Li P. On distributed adaptive optimization with gradient compression. arXiv preprint arXiv: 2205.05632, 2022

29 Dean J, Corrado G, Monga R, Chen K, Devin M, Mao M, et al. Large scale distributed deep networks. In: Proceedings of the 26th International Conference on Neural Information Processing Systems. Nevada, USA: Curran Associates, Inc. 2012. 1223−1231

30 Huang Y, Cheng Y, Bapna A, Firat O, Chen D, Chen M, et al. GPipe: Efficient training of giant neural networks using pipeline parallelism. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. Vancouver, Canada: Curran Associates, Inc. 2019. 103−112

31 Narayanan D, Harlap A, Phanishayee A, Seshadri V, Devanur N R, Ganger G R, et al. PipeDream: Generalized pipeline parallelism for DNN training. In: Proceedings of the 27th ACM Symposium on Operating Systems Principles. Huntsville, Canada: ACM. 2019. 1−15

32 Nedic A, Ozdaglar A. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 2009, **54**(1): 48−61

33 Lopes C G, Sayed A H. Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Transactions on Signal Processing*, 2008, **56**(7): 3122−3136

34 Xu J, Zhu S, Soh Y C, Xie L. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. *IEEE Conference on Decision and Control (CDC)*, 2015: 2055−2060

35 Nedic A, Olshevsky A, Shi W. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 2017, **27**(4): 2597−2633

36 Shi W, Ling Q, Wu G, Yin W. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 2015, **25**(2): 944−966

37 Yuan K, Ying B, Zhao X, Sayed A H. Exact diffusion for distributed optimization and learning——Part I: Algorithm development. *IEEE Transactions on Signal Processing*, 2018, **67**(3): 708−723

38 Kong B, Zhu S, Lu S, Huang X, Yuan K. Decentralized bilevel optimization over graphs: Loopless algorithmic update and transient iteration complexity. arXiv preprint arXiv: 2402.03167, 2024.

39 Zhu S, Kong B, Lu S, Huang X, Yuan K. SPARKLE: A unified single-loop primal-dual framework for decentralized bilevel optimization. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. Vancouver, Canada: Curran Associates, Inc. 2024. 62912−62987

40 He Y, Shang Q, Huang X, Liu J, Yuan K. A mathematics-inspired learning-to-optimize framework for decentralized optimization. arXiv preprint arXiv: 2410.01700, 2024.

41 McMahan B, Moore E, Ramage D, Hampson S, y Arcas B A. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Fort Lauderdale, USA: PMLR, 2017. 1273−1282

42 Karimireddy S P, Kale S, Mohri M, Reddi S, Stich S, Suresh A T. SCAFFOLD: Stochastic controlled averaging for federated learning. In: Proceedings of the 37th International Conference on Machine Learning. Virtual Event: PMLR, 2020. 5132−5143

43 Mishchenko K, Malinovsky G, Stich S, Richtárik P. ProxSkip: Yes! local gradient steps provably lead to communication acceleration! finally! In: Proceedings of the 39th International Conference on Machine Learning. Baltimore, USA: PMLR, 2022. 15750−15769

44 Condat L, Maranjyan A, Richtárik P. LoCoDL: Communication-efficient distributed learning with local training and compression. arXiv preprint arXiv: 2403.04348, 2024.

45 Douillard A, Feng Q, Rusu A A, Chhaparia R, Donchev Y, Kuncoro A, et al. DiLoCo: Distributed low-communication training of language models. arXiv preprint arXiv: 2311.08105, 2023.

46 Wang H, Sievert S, Liu S, Charles Z, Papailiopoulos D, Wright S. ATOMO: Communication-efficient learning via atomic sparsification. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal, Canada: Curran Associates, Inc. 2018. 9872−9883

47 Mishchenko K, Wang B, Kovalev D, Richtárik P. IntSGD: Adaptive floatless compression of stochastic gradients. arXiv pre-

print arXiv: 2102.08374, 2021.

48 Mayekar P, Tyagi H. RATQ: A universal fixed-length quantizer for stochastic optimization. In: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics. Virtual Event: PMLR, 2020. 1399−1409

49 Wen W, Xu C, Yan F, Wu C, Wang Y, Chen Y, et al. TernGrad: Ternary gradients to reduce communication in distributed deep learning. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, USA: Curran Associates, Inc. 2017. 1508−1518

50 Gandikota V, Kane D, Maity R K, Mazumdar A. vqSGD: Vector quantized stochastic gradient descent. In: Proceedings of the 24th International Conference on Artificial Intelligence and Statistics. Virtual Evenvt: PMLR, 2021. 2197−2205

51 He Y, Hu J, Huang X, Lu S, Wang B, Yuan K. Distributed bilevel optimization with communication compression. In: Proceedings of the 41st International Conference on Machine Learning. Vienna, Austria: PMLR, 2024. 17877−17920

52 Modoranu I V, Safaryan M, Malinovsky G, Kurtić E, Robert T, Richtárik P, et al. MicroAdam: Accurate adaptive optimization with low space overhead and provable convergence. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. Vancouver, Canada: Curran Associates, Inc. 2024. 1−43

53 Gorbunov E, Burlachenko K P, Li Z, Richtárik P. MARINA: Faster non-convex distributed learning with compression. In: Proceedings of the 38th International Conference on Machine Learning. Virtual Event: PMLR, 2021. 3788−3798

54 Li Z, Richtárik P. CANITA: Faster rates for distributed convex optimization with communication compression. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. Virtual Event: Curran Associates, Inc. 2021. 13770−13781

55 Li Z, Kovalev D, Qian X, Richtárik P. Acceleration for compressed gradient descent in distributed and federated optimization. arXiv preprint arXiv: 2002.11364, 2020.

56 He Y, Huang X, Yuan K. Unbiased compression saves communication in distributed optimization: When and how much? In: Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans, USA: Curran Associates, Inc. 2023. 47991−48020

57 He Y, Huang X, Chen Y, Yin W, Yuan K. Lower bounds and accelerated algorithms in distributed stochastic optimization with communication compression. arXiv preprint arXiv: 2305.07612, 2023.

58 Zhao H, Xie X, Fang C, Lin Z. SEPARATE: A simple low-rank projection for gradient compression in modern large-scale model training process. In: Proceedings of the 13th International Conference on Learning Representations. Singapore: OpenReview.net. 2025.

59 Mishchenko K, Gorbunov E, Takáč M, Richtárik P. Distributed learning with compressed gradient differences. *Optimization Methods and Software*, 2024: 1−16

60 Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 2011, **12**(7): 2121−2159

61 Zeiler M D. ADADELTA: An adaptive learning rate method. arXiv preprint arXiv: 1212.5701, 2012.

62 Loshchilov I, Hutter F. Decoupled weight decay regularization. arXiv preprint arXiv: 1711.05101, 2017.

63 Reddi S J, Kale S, Kumar S. On the convergence of Adam and beyond. arXiv preprint arXiv: 1904.09237, 2019.

64 He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, USA: IEEE, 2016. 770−778

65 Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. arXiv preprint arXiv: 2010.11929, 2020.

66 Wang A, Singh A, Michael J, Hill F, Levy O, Bowman S R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv: 1804.07461, 2018.

67 Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. RoBERTa: A robustly optimized bert pretraining approach. arXiv preprint arXiv: 1907.11692, 2019.

68 He P, Liu X, Gao J, Chen W. DeBERTa: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv: 2006.03654, 2020.

69 Touvron H, Lavril T, Izacard G, Martinet X, Lachaux M A, Lacroix T, et al. LLaMA: Open and efficient foundation language models. arXiv preprint arXiv: 2302.13971, 2023.

70 Robert T, Safaryan M, Modoranu I V, Alistarh D. LDAdam: Adaptive optimization from low-dimensional gradient statistics. arXiv preprint arXiv: 2410.16103, 2024.

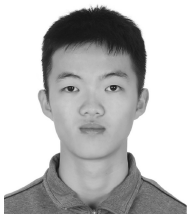
71 Hu Z, Wang L, Lan Y, Xu W, Lim E P, Bing L, et al. LLM-Adapters: An adapter family for parameter-efficient fine-tuning of large language models. arXiv preprint arXiv: 2304.01933, 2023.

72 Cobbe K, Kosaraju V, Bavarian M, Chen M, Jun H, Kaiser L, et al. Training verifiers to solve math word problems. arXiv preprint arXiv: 2110.14168, 2021.

73 Patel A, Bhattamishra S, Goyal N. Are NLP models really able to solve simple math word problems? In: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics. Virtual Event: Association for Computational Linguistics. 2021. 2080−2094

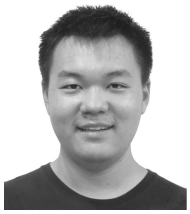
74 Ling W, Yogatama D, Dyer C, Blunsom P. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver, Canada: Association for Computational Linguistics. 2017. 158−167

75 Roy S, Roth D. Solving general arithmetic word problems. arXiv preprint arXiv: 1608.01413, 2016.



陈楚岩 北京大学数学科学学院硕士研究生. 主要研究方向为优化算法和模型压缩.
E-mail: chuyanchen@stu.pku.edu.cn
(**CHEN Chu-Yan** Master student at the School of Mathematical Sciences, Peking University. His research interests include optimization algorithm and model compression.)

刘烨谞 北京大学数学科学学院硕士研究生. 主要研究方向为最优化和机器学习.
E-mail: 2200010762@stu.pku.edu.cn
(**LIU Ye-Xu** Master student at the School of Mathematical Sciences, Peking University. His research interests include optimization and machine learning.)



贾维宸 北京大学生命科学学院硕士研究生. 主要研究方向为机器学习, 基因组学分析和植物生殖生物学.
E-mail: 2100012106@stu.pku.edu.cn
(**JIA Wei-Chen** Master student at the School of Life Sciences, Peking University. His research interests include machine learning, genomic analysis, and plant reproductive biology.)

何雨桐 北京大学大数据科学研究中心博士研究生. 2022 年获得北京大学学士学位. 主要研究方向为分布式优化和大语言模型.
E-mail: yutonghe@pku.edu.cn
(**HE Yu-Tong** Ph.D. candidate at the Center for Data Science, Peking University. He received his bachelor degree from Peking University in 2022. His research interests include distributed optimization and large language models.)

袁 坤 北京大学国际机器学习研究中心助理教授. 主要研究方向为面向大规模机器学习问题的最优化算法. 本文通信作者.
E-mail: kunyuan@pku.edu.cn
(**YUAN Kun** Assistant professor at the Center for Machine Learning Research, Peking University. His research interests include optimization algorithm for large-scale machine learning. Corresponding author of this paper.)

王立威 北京大学智能学院教授. 主要研究方向为机器学习基础理论和科学智能.
E-mail: wanglw@pku.edu.cn
(**WANG Li-Wei** Professor at the School of Intelligence Science and Technology, Peking University. His research interests include theory foundations of machine learning and artificial intelligence for science.)