Lion优化器的收敛分析与改进

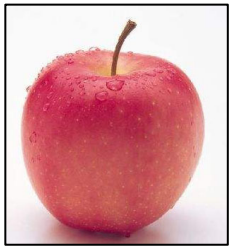
张利军
南京大学

CSIG青年科学家会议
“大模型的高效训练与推理”论坛

目录

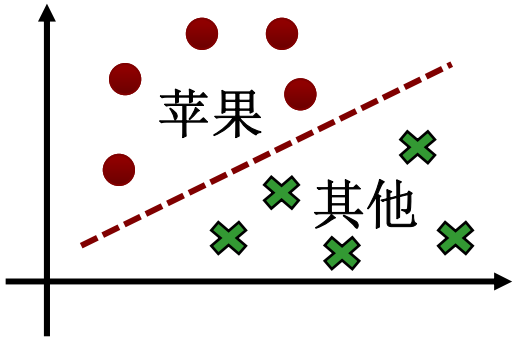
机器学习中的优化问题

➢ 监督学习



苹果

$$(\mathbf{x}_1, y_1)$$
$$\cdots$$
$$(\mathbf{x}_n, y_n) \Rightarrow y \approx h(\mathbf{x})$$



➢ 经验风险最小化

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(\mathbf{x}_i), y_i) + \phi(h)$$

- n是样本数量、$\mathcal{H}$为假设空间，其维度为d
- $\ell(\cdot,\cdot)$为损失函数、$\phi(\cdot)$为正则化因子

典型学习算法

- 最小二乘回归

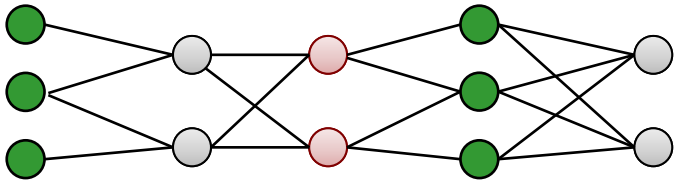
$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^\top \mathbf{x}_i - y_i)^2$$

  - 平方损失：$\ell(u,v) = (u-v)^2$
  - 线性函数空间：$\mathcal{H} = \{h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}\colon\ \mathbf{w} \in \mathbb{R}^d\}$

- 深度学习

  - 交叉熵损失：$\ell(h,y) = -\sum_{c=1}^{K} y_c \log \frac{e^{h_c}}{\sum_{j=1}^{K} e^{h_j}}$
  - 函数空间：神经网络

$$\mathcal{H} = \{h(\mathbf{x};\mathbf{w})\colon \mathbf{w} \in \mathbb{R}^d\}$$



数学优化

- 优化问题

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}^{\top}\mathbf{x}_i, y_i)$$

- 确定优化——梯度下降（GD）

1. 计算梯度

$$\nabla F(\mathbf{w}_t) = \frac{1}{n}\sum_{i=1}^{n} \nabla \ell\left(\mathbf{w}_t^{\top}\mathbf{x}_i, y_i\right)$$

计算复杂度$O(nd)$

2. 更新模型

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla F(\mathbf{w}_t)$$

计算复杂度高×、收敛速率快✓

随机优化

➢ 优化问题

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}^{\top}\mathbf{x}_i, y_i)$$

SGD及其变形（如Adam）被广泛应用

➢ 随机优化——随机梯度下降（SGD）

1. 随机采样 $(\mathbf{x}_t, y_t) \sim \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$

2. 计算随机梯度 $\mathbf{g}_t = \nabla \ell\left(\mathbf{w}_t^{\top}\mathbf{x}_t, y_t\right)$ — 计算复杂度$O(d)$

3. 更新模型 $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{g}_t$

计算复杂度低✓、收敛速率慢×

梯度符号随机优化 [Chen et al., 2014]

- 符号随机梯度下降（signSGD）

1. 随机梯度

$$\mathbf{g}_t = \nabla \ell\left(\mathbf{w}_t^\top \mathbf{x}_t, y_t\right)$$

2. 利用梯度符号更新模型

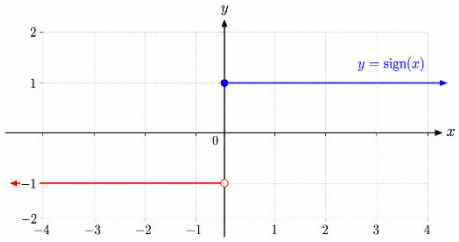
$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathrm{Sign}(\mathbf{g}_t)$$

$$\mathrm{Sign}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

- 优势

降低
通讯代价

抑制
噪声影响



动量符号随机优化 [Bernstein et al., 2018]

➢ 结合动量的符号随机梯度下降（signSGD with Momentum）

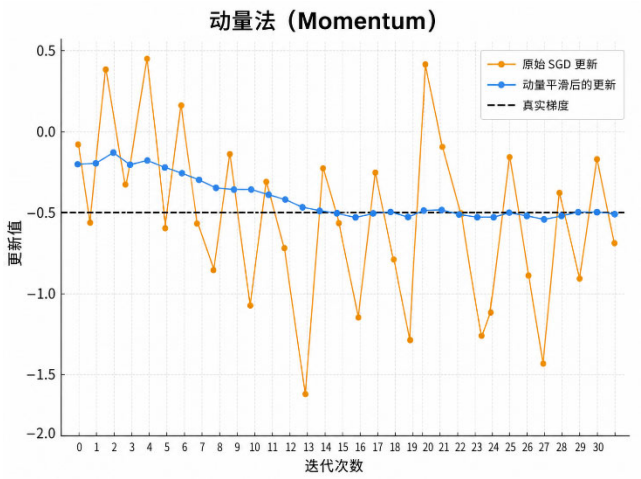
1. 随机梯度

$$\mathbf{g}_t = \nabla \ell\left(\mathbf{w}_t^{\top}\mathbf{x}_t, y_t\right)$$

2. 梯度动量

$$\mathbf{v}_t = (1-\beta)\mathbf{v}_{t-1} + \beta\mathbf{g}_t$$

考虑梯度的累积趋势



3. 利用符号更新模型

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathrm{Sign}(\mathbf{v}_t)$$

Lion [Chen et al., 2023]

➢ EvoLved Sign Momentum （ Lion） Google Brain

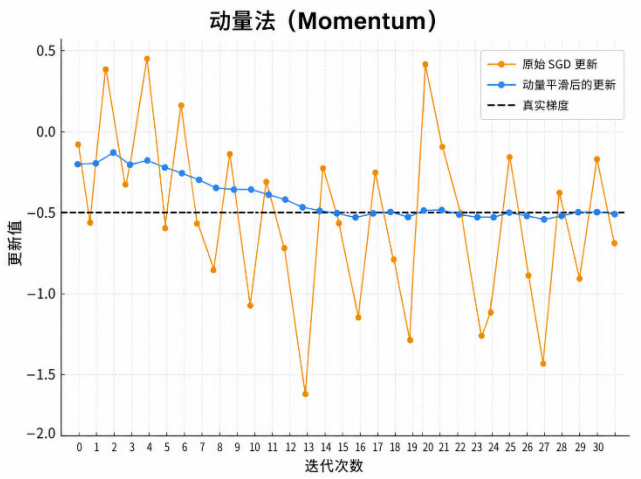
1. 随机梯度

$$\mathbf{g}_t = \nabla \ell\left(\mathbf{w}_t^\top \mathbf{x}_t, y_t\right)$$

2. 梯度动量

$$\mathbf{v}_t = (1-\beta_1)\mathbf{m}_{t-1} + \beta_1 \mathbf{g}_t$$

$$\mathbf{m}_t = (1-\beta_2)\mathbf{m}_{t-1} + \beta_2 \mathbf{g}_t$$



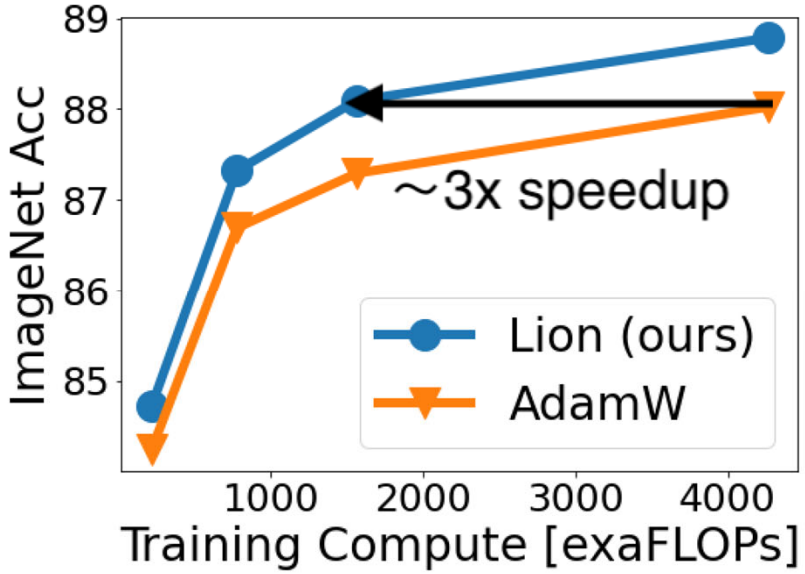
3. 利用符号更新模型

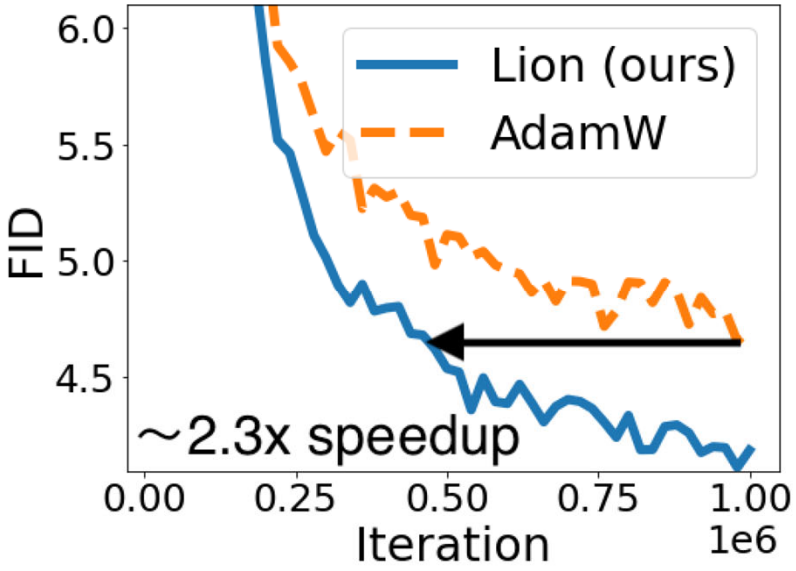
$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t(\mathrm{Sign}(\mathbf{v}_t) + \lambda \mathbf{w}_t)$$

Lion [Chen et al., 2023]

➢ EvoLved Sign Momentum （Lion） Google Brain

- 大模型训练





不仅提高通讯效率，还提升训练效率

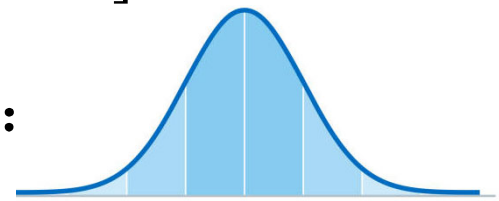
理论研究严重滞后

- 单机场景
  - 依赖大批量采样 [Bernstein et al., 2018]

    收敛速率：$O\left(\frac{1}{T^{1/4}}\right)$　　批量大小：$O\left(\frac{1}{\epsilon^2}\right)$

  - 对噪声添加额外假设 [Bernstein et al., 2019]

    收敛速率：$O\left(\frac{1}{T^{1/4}}\right)$　　单峰对称噪声：

  - 收敛速率慢 [Sun et al., 2023]

    收敛速率：$O\left(\frac{d}{T^{1/4}}\right)$，对维度$d$的依赖过高

理论研究严重滞后

- 分布式场景
  - 收敛速率慢 [Jin et al., 2021]

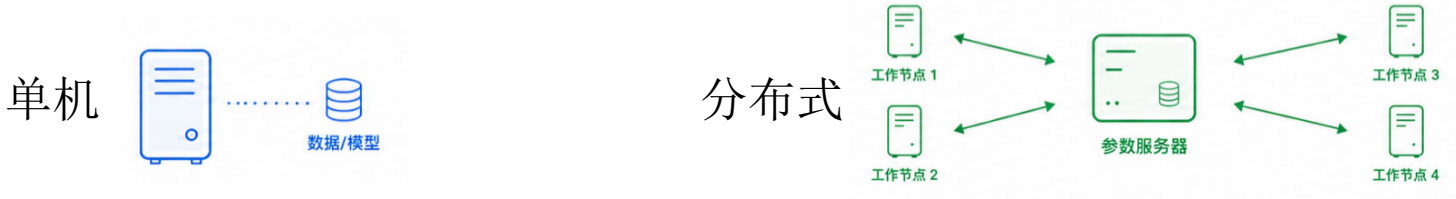
    收敛速率：$O\left(\frac{d^{3/8}}{T^{1/8}}\right)$，对迭代轮数$T$的依赖差

  - 无法收敛到0 [Sun et al., 2023]

    收敛速率：$O\left(\frac{d}{T^{1/4}}+\frac{d}{\sqrt{m}}\right)$，$m$为分布式节点数量

我们的贡献

- signSGD的改进 [Jiang et al., NeurIPS 2024]
  - 利用方差约减提升速率 $O\left(\frac{1}{T^{1/4}}\right) \longrightarrow O\left(\frac{1}{T^{1/3}}\right)$
- signSGD+Momentum的分析 [Jiang et al., Arxiv 2025]
  - 理论分析不需要强假设
- Lion的分析与改进 [Jiang et al., ICML 2026]
  - 理论分析不需要强假设
  - 利用方差约减提升速率 $O\left(\frac{1}{T^{1/4}}\right) \longrightarrow O\left(\frac{1}{T^{1/3}}\right)$



目录

Lion的现有结果

- 算法流程

1. 随机梯度 $\mathbf{g}_t = \nabla \ell\left(\mathbf{w}_t^\top \mathbf{x}_t, y_t\right)$

2. 梯度动量 $\mathbf{v}_t = (1-\beta_1)\mathbf{m}_{t-1} + \beta_1 \mathbf{g}_t$

$$\mathbf{m}_t = (1-\beta_2)\mathbf{m}_{t-1} + \beta_2 \mathbf{g}_t$$

3. 模型更新 $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t(\mathrm{Sign}(\mathbf{v}_t) + \lambda \mathbf{w}_t)$

- 现有结果 [Dong et al., 2024]

强制性假设：$\lim_{\|\mathbf{w}\|\to\infty} F(\mathbf{w}) \to \infty$

收敛率：$O\left(\frac{\sqrt{d}}{T^{1/4}}\right)$

Lion的收敛分析

➢ 理论保障 [Jiang et al., ICML 2026]

假设1：函数平滑

$$\|\nabla F(\mathbf{u}) - \nabla F(\mathbf{v})\| \leq L\|\mathbf{u} - \mathbf{v}\|$$

假设2：方差有界

$$\mathbb{E}_{(\mathbf{x},y)}\left[\left\|\nabla F(\mathbf{w}) - \nabla \ell(\mathbf{w}^\top \mathbf{x}, y)\right\|^2\right] \leq \sigma^2$$

◘收敛速率

$$\mathbb{E}[\|\nabla F(\mathbf{w}_\tau)\|_1] = O\left(\frac{\sqrt{d}}{T^{1/4}}\right)$$

目录

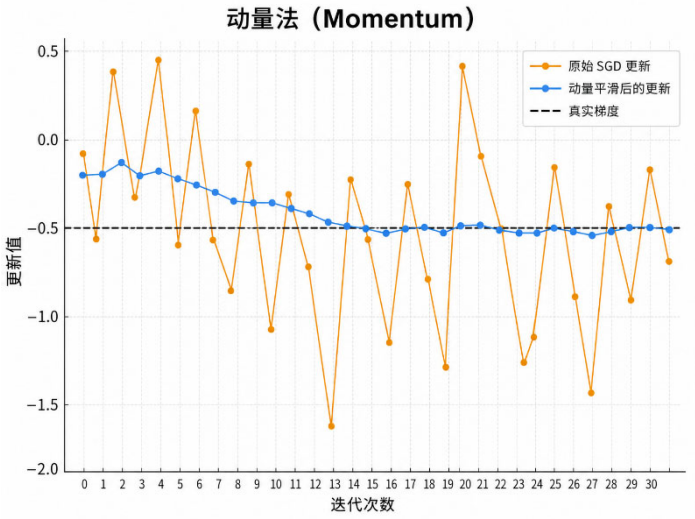
- 研究背景
- 单机Lion
  - 收敛率分析
  - 方差约减
- 分布式Lion
  - 单向压缩
  - 双向压缩
- 总结展望

方差约减

- 动量法（Momentum）
  - 经典技术

$$\mathbf{v}_t = (1-\beta)\mathbf{v}_{t-1} + \beta\mathbf{g}_t$$

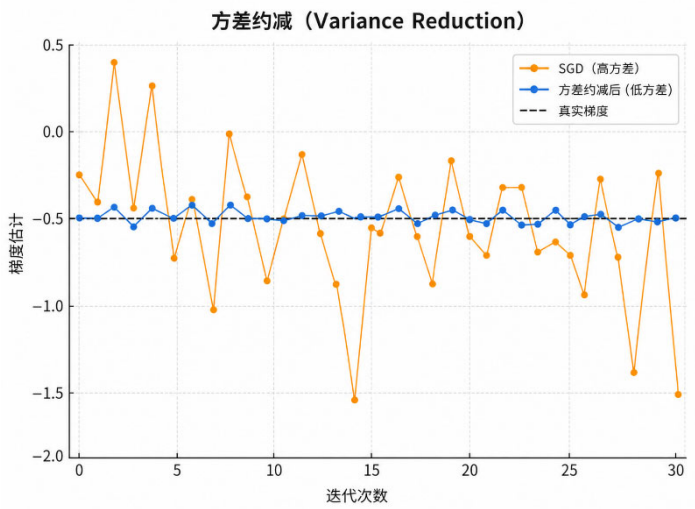
局部光滑



- 方差约减（Variance Reduction）
  - [Zhang et al., NIPS 2013]

$$\mathbf{g}_t(\mathbf{w}_t) - \mathbf{g}_t(\widehat{\mathbf{w}}) + \nabla F(\widehat{\mathbf{w}})$$

降低偏差



方差约减的Lion

➢ 算法流程

1. 随机梯度 $\mathbf{g}_t(\mathbf{w}_t) = \nabla \ell\left(\mathbf{w}_t^\top \mathbf{x}_t, y_t\right)$

2. 方差约减 [Cutkosky and Orabona, 2019]

$$\mathbf{v}_t = (1-\beta_1)\mathbf{m}_{t-1} + \beta_1 \mathbf{g}_t$$

$$\mathbf{m}_t = (1-\beta_2)\mathbf{m}_{t-1} + \beta_2 \mathbf{g}_t$$

梯度作差降低偏差 $+\gamma\left(\mathbf{g}_t(\mathbf{w}_t) - \mathbf{g}_t(\mathbf{w}_{t-1})\right)$

3. 模型更新 $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t(\mathrm{Sign}(\mathbf{v}_t) + \lambda \mathbf{w}_t)$

收敛分析

➢ 理论保障 [Jiang et al., ICML 2026]

假设1：平均平滑

$$\mathbb{E}_{(\mathbf{x},y)}\left[\left\|\nabla\ell(\mathbf{u}^\top\mathbf{x},y)-\nabla\ell(\mathbf{v}^\top\mathbf{x},y)\right\|^2\right]\leq L^2\|\mathbf{u}-\mathbf{v}\|^2$$

假设2：方差有界

$$\mathbb{E}_{(\mathbf{x},y)}\left[\left\|\nabla F(\mathbf{w})-\nabla\ell(\mathbf{w}^\top\mathbf{x},y)\right\|^2\right]\leq\sigma^2$$

◘收敛速率

$$\mathbb{E}[\|\nabla F(\mathbf{w}_\tau)\|_1]=O\left(\frac{\sqrt{d}}{T^{1/3}}\right)$$

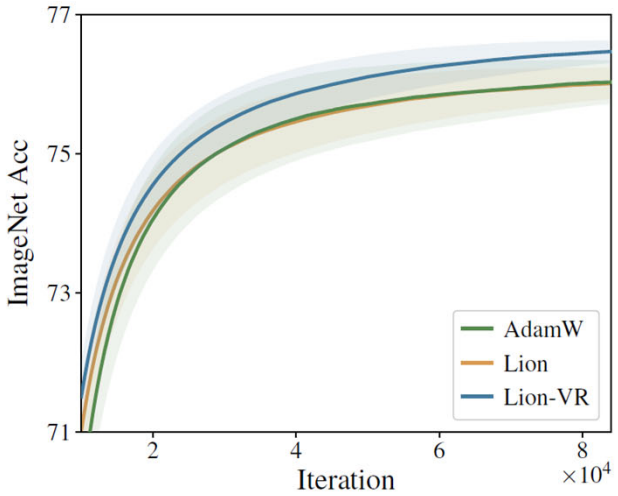
单机理论结果

➢ 收敛率对比

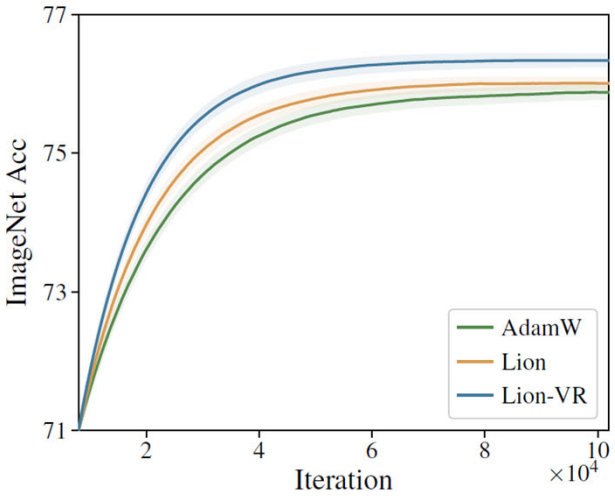
方法	收敛速率	额外假设
Dong et al., 2024	$O\left(\sqrt{d}/T^{1/4}\right)$	强制性
Jiang et al., 2026	$O\left(\sqrt{d}/T^{1/4}\right)$	-
Jiang et al., 2026	$O\left(\sqrt{d}/T^{1/3}\right)$	平均光滑性

实验结果

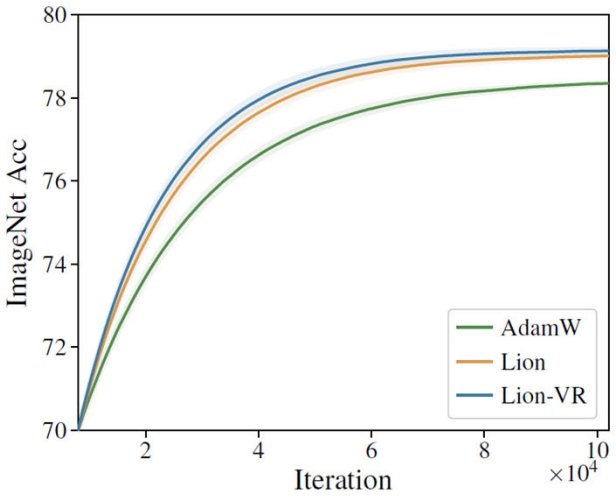
- ImageNet、准确率



ResNet-50



ViT-S without data augmentation



ViT-S with RandAug+MixUp

实验结果

- ImageNet、准确率

Model	Optimizer	RandAug +MixUp	Accuracy(%)
ResNet-50	AdamW	–	76.03±0.31
	Lion		76.01±0.22
	Lion-VR		**76.47±0.16**
ViT-S	AdamW	✗	75.88±0.11
	Lion		76.01±0.09
	Lion-VR		**76.34±0.10**
	AdamW	✓	78.35±0.06
	Lion		79.01±0.08
	Lion-VR		**79.13±0.09**

目录

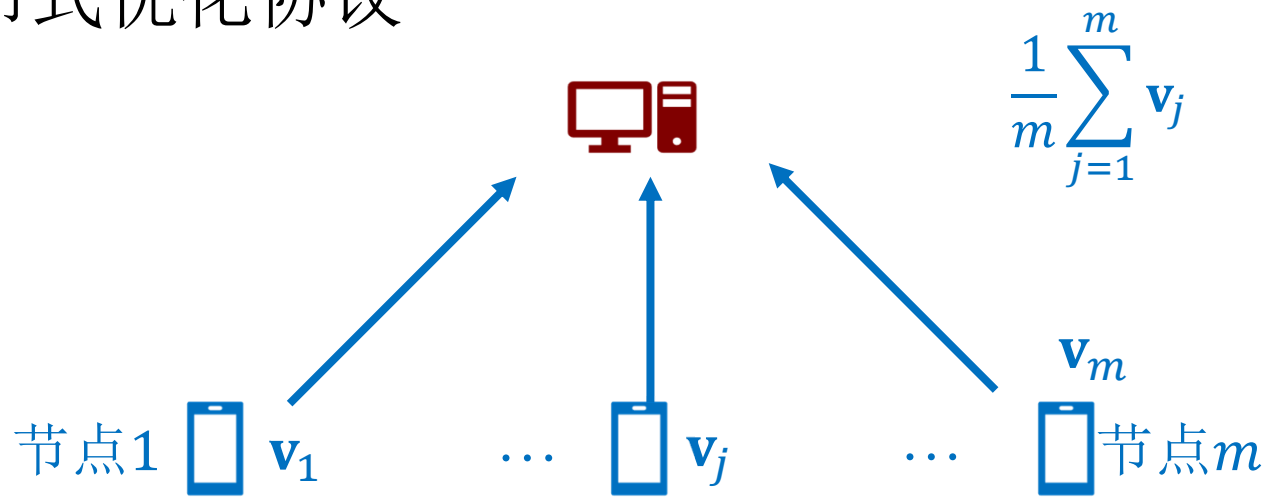
- 研究背景
- 单机Lion
  - 收敛率分析
  - 方差约减
- 分布式Lion
  - 单向压缩
  - 双向压缩
- 总结展望

分布式场景

- 每个函数F_j分布在不同的节点上

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{m}\sum_{j=1}^{m} F_j(\mathbf{w}), \quad F_j(\mathbf{w}) = \mathbb{E}_{\xi^j \sim \mathcal{D}^j}\left[\ell(\mathbf{w}; \xi^j)\right]$$

- 分布式优化协议



$$F_1(\mathbf{w}) = \mathbb{E}_{\xi^1 \sim \mathcal{D}^1}\left[\ell(\mathbf{w}; \xi^1)\right] \qquad F_m(\mathbf{w}) = \mathbb{E}_{\xi^m \sim \mathcal{D}^m}\left[\ell(\mathbf{w}; \xi^m)\right]$$

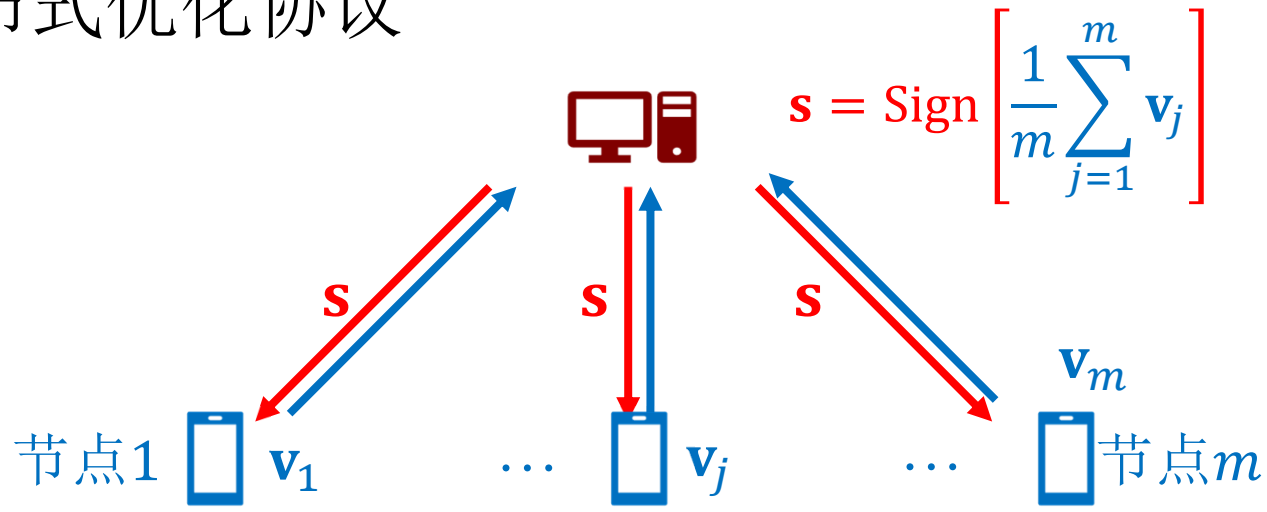
分布式场景

单向压缩

- 每个函数F_j分布在不同的节点上

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{m}\sum_{j=1}^{m} F_j(\mathbf{w}), \quad F_j(\mathbf{w}) = \mathbb{E}_{\xi^j \sim \mathcal{D}^j}\left[\ell(\mathbf{w}; \xi^j)\right]$$

- 分布式优化协议



$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t(\mathbf{s} + \lambda \mathbf{w}_t)$$

分布式Lion（单向）

- 算法流程 [Jiang et al., ICML 2026]

1. 各节点j计算随机梯度

$$\mathbf{g}_t^j = \nabla \ell\left(\mathbf{w}_t^\top \mathbf{x}_t^j, y_t^j\right)$$

2. 各节点j维护梯度动量

$$\mathbf{v}_t^j = (1-\beta_1)\mathbf{m}_{t-1}^j + \beta_1 \mathbf{g}_t^j$$

$$\mathbf{m}_t^j = (1-\beta_2)\mathbf{m}_{t-1}^j + \beta_2 \mathbf{g}_t^j$$

3. 各节点j传递实数$\mathbf{v}_t^j$到服务器

4. 服务器传递聚合符号 $\mathbf{s}_t = \mathrm{Sign}\left(\sum_{j=1}^m \mathbf{v}_t^j\right)$

5. 各节点j更新模型 $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t(\mathbf{s}_t + \lambda \mathbf{w}_t)$

收敛分析

➢ 理论保障 [Jiang et al., ICML 2026]

假设1：各节点函数平滑

$$\left\|\nabla F_j(\mathbf{u}) - \nabla F_j(\mathbf{v})\right\| \leq L\|\mathbf{u} - \mathbf{v}\|$$

假设2：各节点方差有界

$$\mathbb{E}_{(\mathbf{x}^j, y^j)}\left[\left\|\nabla F_j(\mathbf{w}) - \nabla \ell\left(\mathbf{w}^\top \mathbf{x}^j, y^j\right)\right\|^2\right] \leq \sigma^2$$

◘收敛速率

$$\mathbb{E}[\|\nabla F(\mathbf{w}_\tau)\|_1] = O\left(\frac{\sqrt{d}}{m^{1/4} T^{1/4}}\right)$$

方差约减的分布式Lion（单向）

➢ 算法流程 [Jiang et al., ICML 2026]

1. 各节点j计算随机梯度

$$\mathbf{g}_t^j(\mathbf{w}_t) = \nabla \ell\left(\mathbf{w}_t^\top \mathbf{x}_t^j, y_t^j\right)$$

两次
方差约减

2. 各节点j进行方差约减

$$\mathbf{v}_t^j = (1-\beta_1)\mathbf{m}_{t-1}^j + \beta_1 \mathbf{g}_t^j(\mathbf{w}_t) + \gamma_1 \left(\mathbf{g}_t^j(\mathbf{w}_t) - \mathbf{g}_t^j(\mathbf{w}_{t-1})\right)$$

$$\mathbf{m}_t^j = (1-\beta_2)\mathbf{m}_{t-1}^j + \beta_2 \mathbf{g}_t^j(\mathbf{w}_t) + \gamma_2 \left(\mathbf{g}_t^j(\mathbf{w}_t) - \mathbf{g}_t^j(\mathbf{w}_{t-1})\right)$$

3. 各节点j传递实数$\mathbf{v}_t^j$到服务器

4. 服务器传递聚合符号 $\mathbf{s}_t = \mathrm{Sign}\left(\sum_{j=1}^m \mathbf{v}_t^j\right)$

5. 各节点j更新模型 $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t(\mathbf{s}_t + \lambda \mathbf{w}_t)$

收敛分析

➢ 理论保障 [Jiang et al., 2024]

假设1：各节点平均平滑

$$\mathbb{E}_{(\mathbf{x}^j,y^j)}\left[\left\|\nabla\ell\left(\mathbf{u}^\top\mathbf{x}^j,y^j\right)-\nabla\ell\left(\mathbf{v}^\top\mathbf{x}^j,y^j\right)\right\|^2\right]\leq L^2\|\mathbf{u}-\mathbf{v}\|^2$$

假设2：各节点方差有界

$$\mathbb{E}_{(\mathbf{x}^j,y^j)}\left[\left\|\nabla F_j(\mathbf{w})-\nabla\ell\left(\mathbf{w}^\top\mathbf{x}^j,y^j\right)\right\|^2\right]\leq\sigma^2$$

◘收敛速率

$$\mathbb{E}[\|\nabla F(\mathbf{w}_\tau)\|_1]=O\left(\frac{\sqrt{d}}{m^{1/3}T^{1/3}}\right)$$

分布式理论结果

➢ 单向压缩的收敛率对比

方法	收敛速率	额外假设
Jiang et al., 2026	$O\left(\sqrt{d}/m^{1/4}T^{1/4}\right)$	-
Jiang et al., 2026	$O\left(\sqrt{d}/m^{1/3}T^{1/3}\right)$	平均光滑性

目录

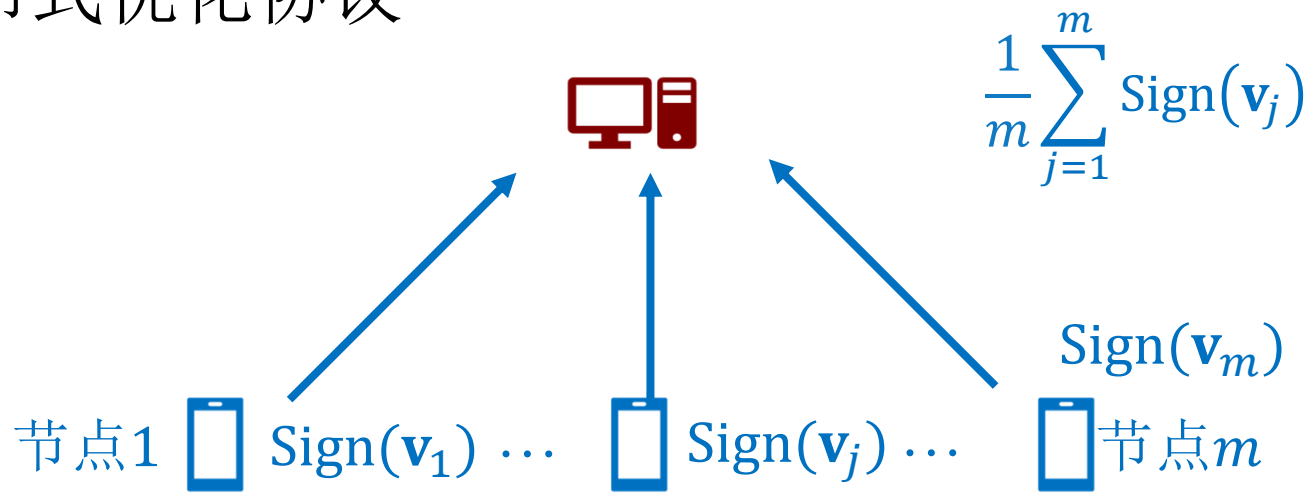
- 研究背景
- 单机Lion
  - 收敛率分析
  - 方差约减
- 分布式Lion
  - 单向压缩
  - 双向压缩
- 总结展望

分布式场景

- 每个函数F_j分布在不同的节点上

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{m}\sum_{j=1}^{m} F_j(\mathbf{w}), \quad F_j(\mathbf{w}) = \mathbb{E}_{\xi^j \sim \mathcal{D}^j}\left[\ell(\mathbf{w}; \xi^j)\right]$$

- 分布式优化协议



$$F_1(\mathbf{w}) = \mathbb{E}_{\xi^1 \sim \mathcal{D}^1}\left[\ell(\mathbf{w}; \xi^1)\right] \qquad F_m(\mathbf{w}) = \mathbb{E}_{\xi^m \sim \mathcal{D}^m}\left[\ell(\mathbf{w}; \xi^m)\right]$$

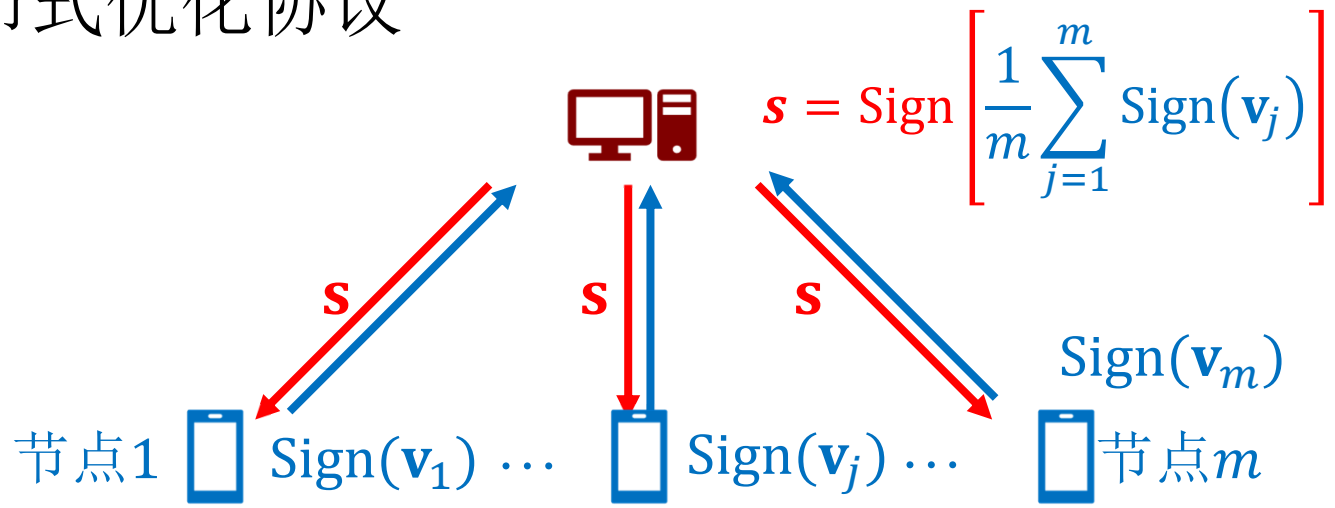
分布式场景

双向压缩？

- 每个函数F_j分布在不同的节点上

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{m}\sum_{j=1}^{m} F_j(\mathbf{w}), \quad F_j(\mathbf{w}) = \mathbb{E}_{\xi^j \sim \mathcal{D}^j}\left[\ell(\mathbf{w}; \xi^j)\right]$$

- 分布式优化协议



$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t(\mathbf{s} + \lambda \mathbf{w}_t)$$

技术挑战

- 两次Sign函数的应用
  - 误差会叠加

$$\mathbf{s} = \mathrm{Sign}\left[\frac{1}{m}\sum_{j=1}^{m}\mathrm{Sign}(\mathbf{v}_j)\right]$$

  - 算法不收敛

- 无偏符号函数
  - 对于向量$\mathbf{v}$，其中$\|\mathbf{v}\|_\infty \le R$，定义

$$[\mathrm{S}_R(\mathbf{v})]_k = \begin{cases} 1, & \text{概率为} \dfrac{R+v_k}{2R} \\ -1, & \text{概率为} \dfrac{R-v_k}{2R} \end{cases}$$

分布式Lion（双向）

- 算法流程 [Jiang et al., ICML 2026]

1. 各节点j计算随机梯度

$$\mathbf{g}_t^j = \nabla \ell\left(\mathbf{w}_t^\top \mathbf{x}_t^j, y_t^j\right)$$

2. 各节点j维护梯度动量

$$\mathbf{v}_t^j = (1-\beta_1)\mathbf{m}_{t-1}^j + \beta_1 \mathbf{g}_t^j$$

$$\mathbf{m}_t^j = (1-\beta_2)\mathbf{m}_{t-1}^j + \beta_2 \mathbf{g}_t^j$$

两次无偏

3. 各节点j传递符号$\mathrm{S}_G\left(\mathbf{v}_t^j\right)$到服务器

4. 服务器传递聚合符号 $\mathbf{s}_t = \mathrm{S}_1\left(\frac{1}{m}\sum_{j=1}^{m} \mathrm{S}_G\left(\mathbf{v}_t^j\right)\right)$

5. 各节点j更新模型 $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t(\mathbf{s}_t + \lambda \mathbf{w}_t)$

收敛分析

➢ 理论保障 [Jiang et al., ICML 2026]

假设1：各节点函数平滑

$$\left\|\nabla F_j(\mathbf{u}) - \nabla F_j(\mathbf{v})\right\| \leq L\|\mathbf{u} - \mathbf{v}\|$$

假设2：各节点方差有界

$$\mathbb{E}_{(\mathbf{x}^j, y^j)}\left[\left\|\nabla F_j(\mathbf{w}) - \nabla \ell\left(\mathbf{w}^\top \mathbf{x}^j, y^j\right)\right\|^2\right] \leq \sigma^2$$

◘收敛速率

$$\mathbb{E}[\|\nabla F(\mathbf{w}_\tau)\|_1] = O\left(\frac{d^{1/4}}{T^{1/4}} + \frac{d^{1/10}}{m^{1/5}T^{1/5}}\right)$$

方差约减的分布式Lion（双向）

➢ 算法流程 [Jiang et al., ICML 2026]

1. 各节点j计算随机梯度

$$\mathbf{g}_t^j(\mathbf{w}_t) = \nabla \ell\left(\mathbf{w}_t^\top \mathbf{x}_t^j, y_t^j\right)$$

2. 各节点j进行方差约减

$$\mathbf{v}_t^j = (1-\beta_1)\mathbf{m}_{t-1}^j + \beta_1 \mathbf{g}_t^j(\mathbf{w}_t)$$

$$\mathbf{m}_t^j = (1-\beta_2)\mathbf{m}_{t-1}^j + \beta_2 \mathbf{g}_t^j(\mathbf{w}_t) + \gamma\left(\mathbf{g}_t^j(\mathbf{w}_t) - \mathbf{g}_t^j(\mathbf{w}_{t-1})\right)$$

一次
方差约减

3. 各节点j传递符号$\mathrm{S}_G\left(\mathbf{v}_t^j\right)$到服务器

4. 服务器传递聚合符号 $\mathbf{s}_t = \mathrm{S}_1\left(\frac{1}{m}\sum_{j=1}^{m} \mathrm{S}_G\left(\mathbf{v}_t^j\right)\right)$

5. 各节点j更新模型 $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t(\mathbf{s}_t + \lambda \mathbf{w}_t)$

收敛分析

➢ 理论保障 [Jiang et al., 2024]

假设1：各节点平均平滑

$$\mathbb{E}_{(\mathbf{x}^j,y^j)}\left[\left\|\nabla\ell\left(\mathbf{u}^\top\mathbf{x}^j,y^j\right)-\nabla\ell\left(\mathbf{v}^\top\mathbf{x}^j,y^j\right)\right\|^2\right]\leq L^2\|\mathbf{u}-\mathbf{v}\|^2$$

假设2：各节点方差有界

$$\mathbb{E}_{(\mathbf{x}^j,y^j)}\left[\left\|\nabla F_j(\mathbf{w})-\nabla\ell\left(\mathbf{w}^\top\mathbf{x}^j,y^j\right)\right\|^2\right]\leq\sigma^2$$

◘收敛速率

$$\mathbb{E}[\|\nabla F(\mathbf{w}_\tau)\|_1]=O\left(\frac{d^{1/4}}{T^{1/4}}\right)$$

分布式理论结果

➢ 单向压缩的收敛率对比

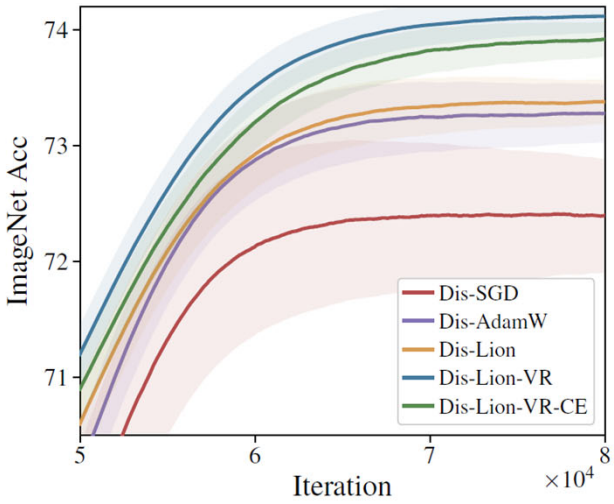
方法	收敛速率	额外假设
Jiang et al., 2026	$O\left(\sqrt{d}/m^{1/4}T^{1/4}\right)$	-
Jiang et al., 2026	$O\left(\sqrt{d}/m^{1/3}T^{1/3}\right)$	平均光滑性

➢ 双向压缩的收敛率对比

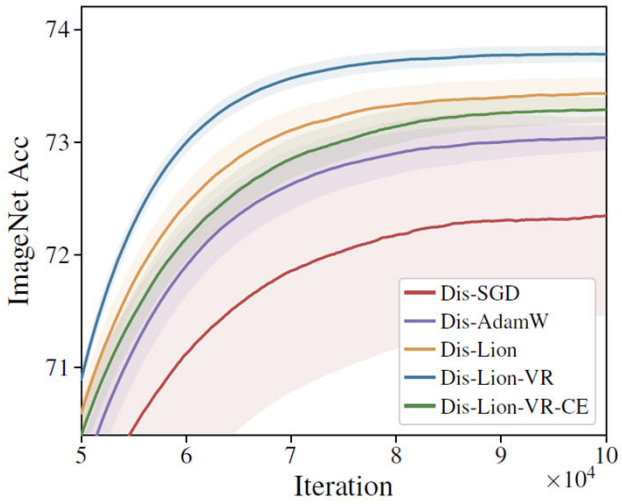
Jin et al., 2021	$O\left(d^{3/8}/T^{1/8}\right)$	-
Jiang et al., 2026	$O\left(d^{1/4}/T^{1/4}+d^{1/10}/n^{1/5}T^{1/5}\right)$	-
Jiang et al., 2026	$O\left(d^{1/4}/T^{1/4}\right)$	平均光滑性

实验结果

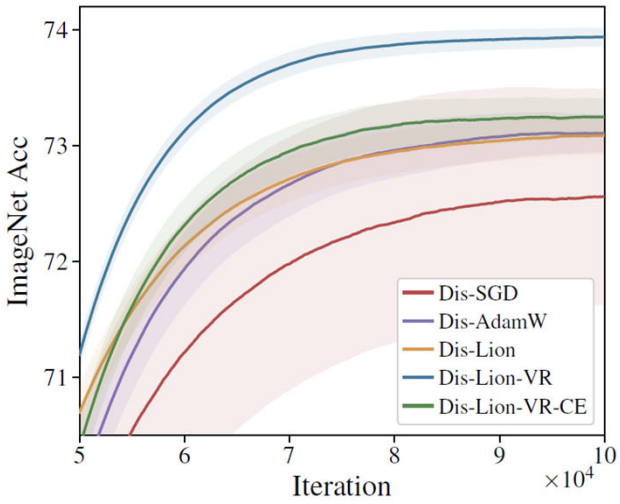
- ImageNet、准确率



ResNet-50



ViT-S without data augmentation



ViT-S with RandAug+MixUp

Dis-Lion-VR：　单向压缩+方差约减
Dis-Lion-VR-CE：双向压缩+方差约减
Dis-Lion：　单向压缩

实验结果

- ImageNet、准确率

Model	Optimizer	RandAug +MixUp	Accuracy (%)
ResNet-50	Dis-SGD	–	72.40±0.49
	Dis-AdamW		73.28±0.25
	Dis-Lion		73.38±0.19
	Dis-Lion-VR		**74.12±0.14**
	Dis-Lion-VR-CE		73.92±0.15
ViT-S	Dis-SGD	✗	72.35±0.88
	Dis-AdamW		73.04±0.11
	Dis-Lion		73.43±0.14
	Dis-Lion-VR		**73.78±0.07**
	Dis-Lion-VR-CE		73.29±0.11
	Dis-SGD	✓	72.56±0.93
	Dis-AdamW		73.11±0.16
	Dis-Lion		73.09±0.16
	Dis-Lion-VR		**73.94±0.08**
	Dis-Lion-VR-CE		73.25±0.16

目录

结果总结

- 单机场景下的Lion优化器

标准假设 $O\left(\frac{\sqrt{d}}{T^{1/4}}\right)$ $\xrightarrow{\text{方差约减}}$ $O\left(\frac{\sqrt{d}}{T^{1/3}}\right)$

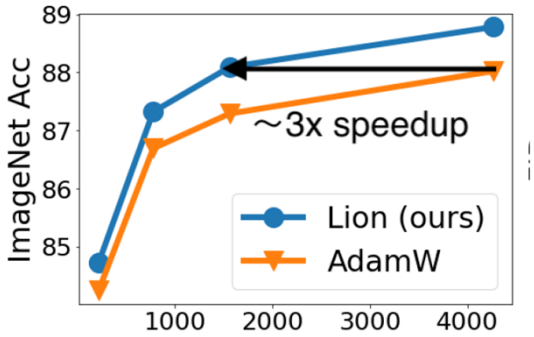
- 分布式场景下的Lion优化器

单向压缩 $O\left(\frac{\sqrt{d}}{m^{1/4}T^{1/4}}\right)$ $\xrightarrow{\text{方差约减}}$ $O\left(\frac{\sqrt{d}}{m^{1/3}T^{1/3}}\right)$

双向压缩 $O\left(\frac{d^{1/4}}{T^{1/4}}+\frac{d^{1/10}}{m^{1/5}T^{1/5}}\right)$ $\xrightarrow{\text{方差约减}}$ $O\left(\frac{d^{1/4}}{T^{1/4}}\right)$

未来展望

- 双向压缩设定下的收敛率有待提升
- 符号优化相比实值优化的额外优势
  - 优化速率与实值类似
  - 实际表现有时候更好
  - 重尾噪声 [Yu et al., 2026]



- 拓展到矩阵变量优化器，如Muon [Jordan, 2024]
  1. 梯度动量 $V_t = \mu V_{t-1} + G_t$
  2. SVD分解 $V_t = U_t \Sigma_t V_t^\top$
  3. 模型更新 $W_{t+1} = W_t - \eta_t U_t \mathrm{Sign}(\Sigma_t) V_t^\top{}_t$

敬请批评指正！

参考文献

- Wei Jiang, Sifan Yang, Wenhao Yang, and Lijun Zhang. Efficient Sign-Based Optimization: Accelerating Convergence via Variance Reduction, NeurIPS 2024.
- Wei Jiang, Dingzhi Yu, Sifan Yang, Wenhao Yang, and Lijun Zhang. Improved Analysis for Sign-based Methods with Momentum Updates, https://arxiv.org/abs/2507.12091, 2025.
- Wei Jiang, Mao Xu, Wenhao Yang, Yibo Wang, Zechao Li, and Lijun Zhang. Convergence Analysis of the Lion Optimizer in Centralized and Distributed Settings, ICML 2026.
- Dingzhi Yu, Hongyi Tao, Yuanyu Wan, Luo Luo, and Lijun Zhang. Sign-Based Optimizers Are Effective Under Heavy-Tailed Noise, https://arxiv.org/abs/2602.07425, 2026.
- Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs, Interspeech 2014.
- Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic Discovery of Optimization Algorithms, NeurIPS 2023.
- Yiming Dong, Huan Li, and Zhouchen Lin. Convergence Rate Analysis of LION, https://arxiv.org/abs/2411.07724, 2024.

参考文献

- Lijun Zhang, Mehrdad Mahdavi, Rong Jin. Linear Convergence with Condition Number Independent Access of Full Gradients, NIPS 2013.
- Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex SGD, NeurIPS 2019.
- Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. SIGNSGD: Compressed Optimisation for Non-Convex Problems, ICML 2018.
- Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signSGD with Majority Vote is Communication Efficient And Fault Tolerant, ICLR 2019.
- Tao Sun, Qingsong Wang, Dongsheng Li, and Bao Wang. Momentum Ensures Convergence of SIGNSGD under Weaker Assumptions, ICML 2023.
- Richeng Jin, Yufan Huang, Xiaofan He, Huaiyu Dai, and Tianfu Wu. Stochastic-Sign SGD for Federated Learning with Theoretical Guarantees, https://arxiv.org/abs/2002.10940, 2021.
- Keller Jordan. Muon: An optimizer for hidden layers in neural networks, https://kellerjordan.github.io/posts/muon, 2024.