南京大学
NANJING UNIVERSITY

南京鼓楼医院
南京大学医学院附属鼓楼医院
Nanjing Drum Tower Hospital of the Affiliated Hospital of Nanjing University Medical School

高质效分布式机器学习

李武军

https://cs.nju.edu.cn/lwj/

南京大学计算机学院

南京大学软件新技术全国重点实验室

南京大学医学院附属鼓楼医院医学大数据中心

2026年5月9日

目录

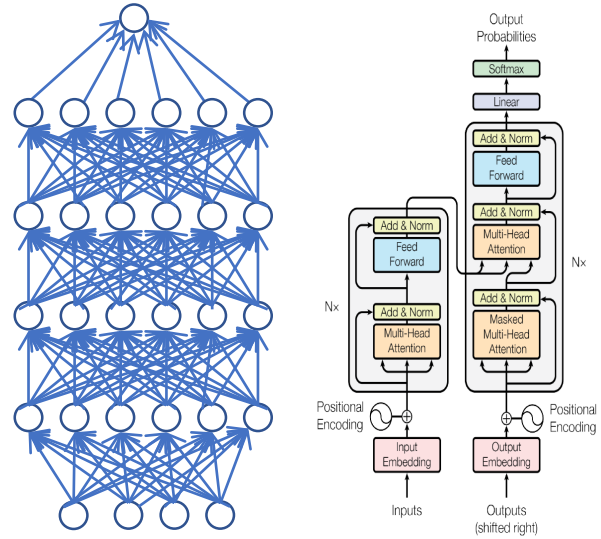
南京大学 NANJING UNIVERSITY

机器学习

给定训练样本集 $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$，解如下优化问题

$$\min_{\mathbf{w}} f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w})$$

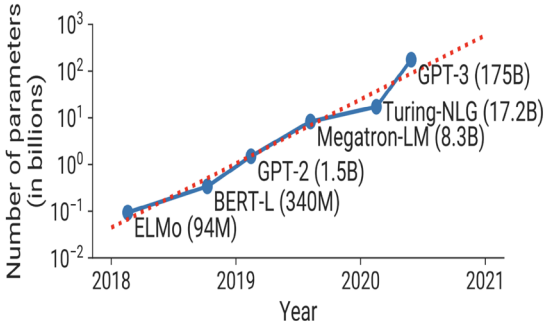
- LR: $f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}[\log(1+e^{-y_i\mathbf{x}_i^T\mathbf{w}}) + \frac{\lambda}{2}\|\mathbf{w}\|^2]$
- SVM: $f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}[\max\{0, 1-y_i\mathbf{x}_i^T\mathbf{w}\} + \frac{\lambda}{2}\|\mathbf{w}\|^2]$
- 深度学习模型
- 其他非监督模型，如PCA和矩阵分解等



$$a_i^l = f(\sum_{j=1}^{n_{l-1}} W_{ij}^l a_j^{l-1} + b_i^l)$$

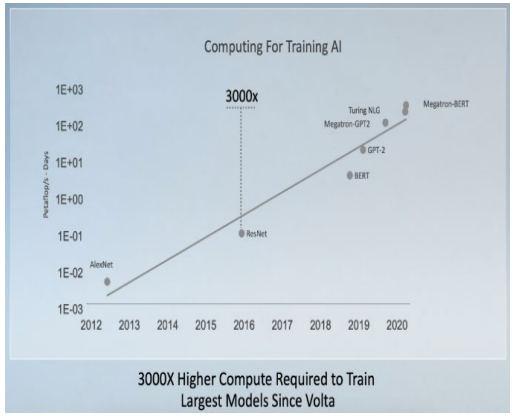
大模型

- 机器学习大模型：大模型+大数据
  - 视觉：ResNet50 (2500万) /ViT (220亿)+ImageNet (1400万)
  - 语言：GPT-3 (1750亿)+维基百科 (25亿)　（ChatGPT、GPT-4、DeepSeek）



[Narayanan et. al., SC'21]

- 挑战：算力（存储/计算/通信）
  - GTC 2020, 英伟达 CEO黄仁勋表示，自2017年底发布 Tesla V100之后，训练最大模型的算力需求增长了3000倍
  - 2025年OpenAI的营收130亿美元，训练成本83亿美元，推理成本84亿美元，导致亏损超140亿美元
  - 2025年智谱总收入7.24 亿元，经调整净亏损31.82 亿元。研发费用31.80 亿元，其中员工薪酬（含股份支付）13.63 亿元，第三方算力服务费约 18.17 亿元



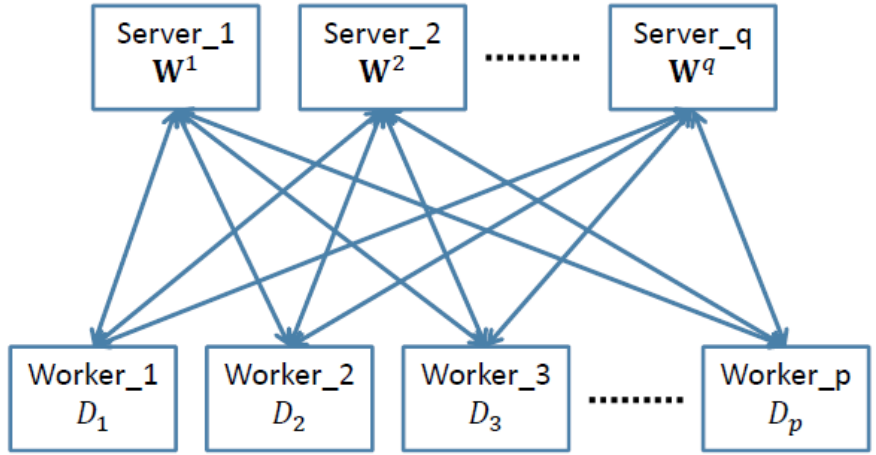
- 在“规模定律（Scaling Law)”指引下，巨大的算力成本已成为大模型和人工智能可持续发展的主要障碍

分布式机器学习（分布式训练/分布式优化）

南京大学 NANJING UNIVERSITY

- 单卡、甚至单机多卡不足以支撑大模型的训练（也叫学习或优化）
  - 存储：存不下
  - 计算：算得慢

$$\min_{\mathbf{w}} f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w})$$



- 基于多机多卡的分布式机器学习成为必需
  - 地理上集中数据
    - 数据中心集群
  - 地理上分散数据
    - 联邦学习
    - 恶意攻击、隐私泄露、通信拥塞、数据Non-IID、异步参与

常见问题

- 老师/领导，现在20（100）张卡没法训练 x B大小的模型，显存溢出了，能否再增加20（100）张卡的预算?

- 老师/领导，现在20（100）张卡训练 x B大小的模型太慢了，能否再增加20（100）张卡的预算?

- 老师/领导，某某国产卡显存又小、计算峰值又低、网络带宽也低、经常出错，根本就没法训练大模型，能否换N公司的A/H/B卡?

常见问题

南京大学 NANJING UNIVERSITY

- 8节点 × 4 DCU/节点

Model	Batch size	Training throughput (samples/s)				#infeasible[5]	#candidate[6]
		Top-1[1]	Top-2[2]	Slowest[3]	Median[4]		
Llama-7B	8	2.01	1.92	0.22	0.82	41	64
Llama-13B	4	0.82	0.58	0.27	0.42	42	48

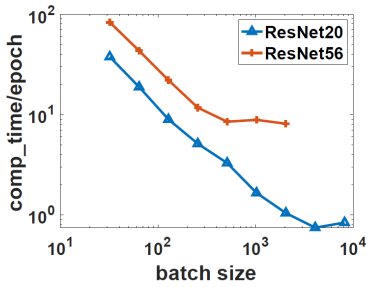
分布式训练技术门槛高
不同的分布式机器学习算法质效差别巨大

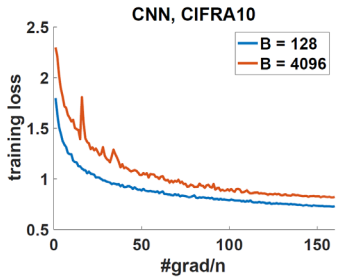
如何设计高质效训练（学习）算法和平台

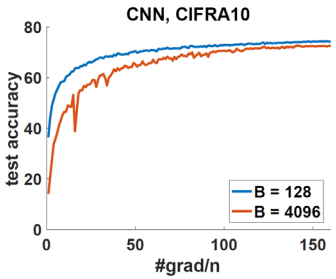
高质：高精度
高效：低成本（高易用性）

分布式机器学习中的挑战

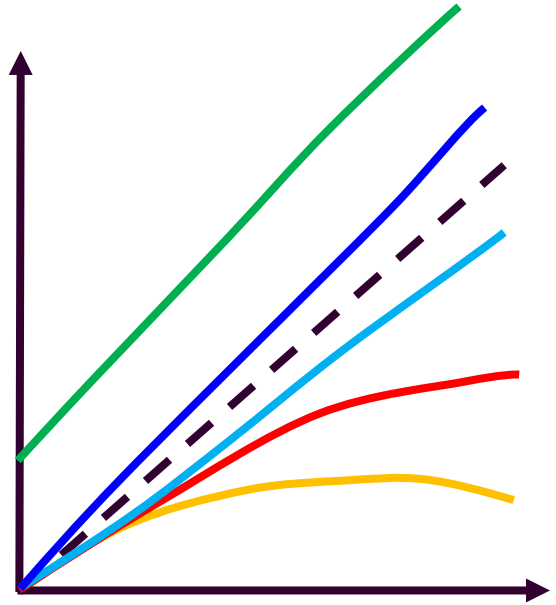
- 质效挑战
  - 算力利用率
    - 单位时间计算的梯度次数
  - 计算（梯度）有效率
    - 单位梯度获得的精度提升
  - 系统容错性
    - 单位时间系统的重启次数
  - 系统易用性







- 算力有限、算力充足都需要解决的挑战
  - 算力有限：如何以有限的算力探索更大的模型
  - 算力充足：如何降本增效



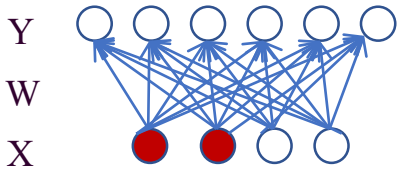
实际算力（质效）增大倍数 = 理论算力增大倍数 * W

W为算法加权因子，可以大于1，也可以小于1

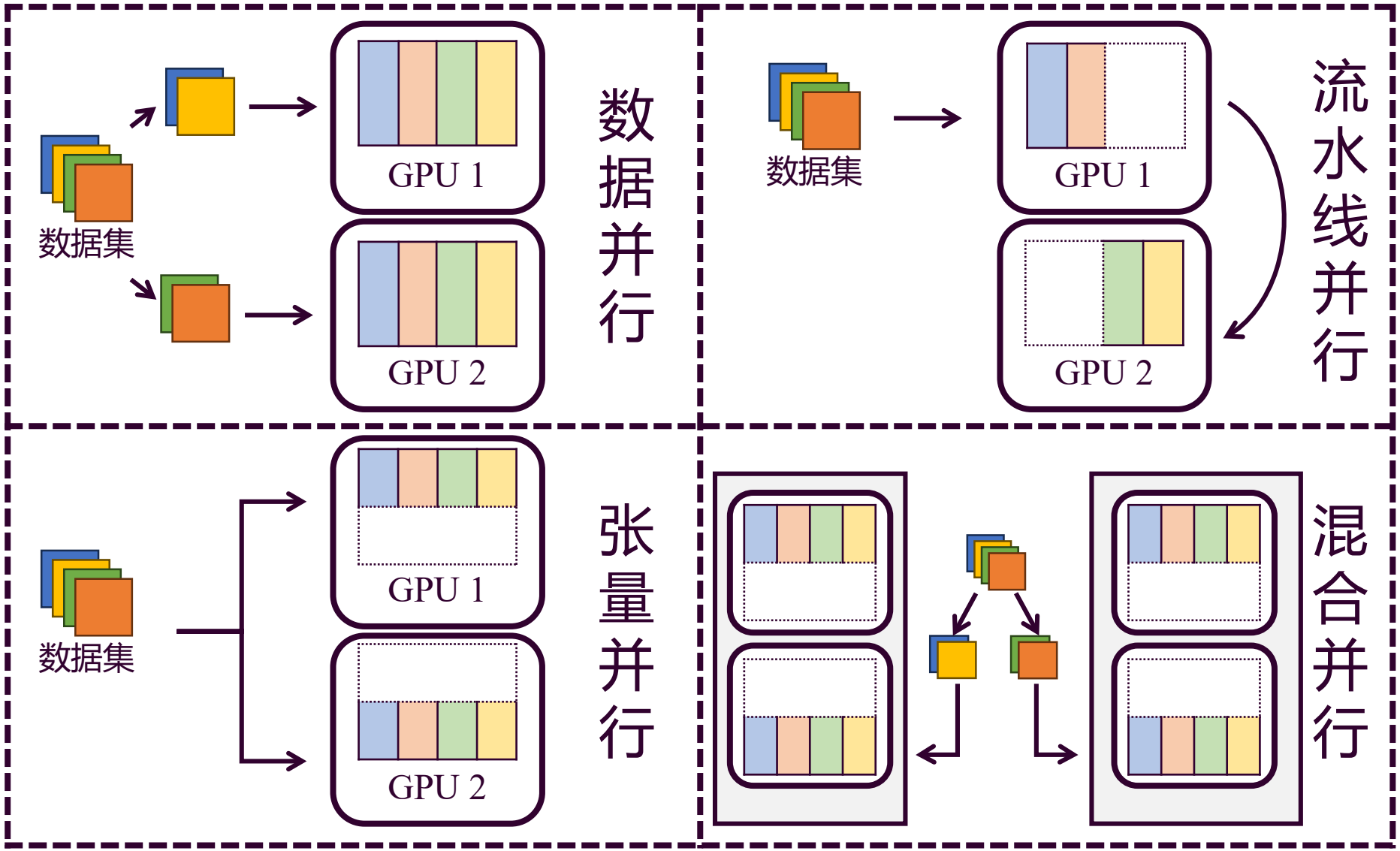
- 高质效分布式机器学习算法
  - ChatGPT：模型、数据、算力、算法

目录

分布式机器学习



- 数据并行
- 模型并行
  - 张量并行
  - 流水线并行
- 混合并行
- 自动并行



分布式机器学习

- 数据并行
- 模型并行
- 混合并行
- 自动并行

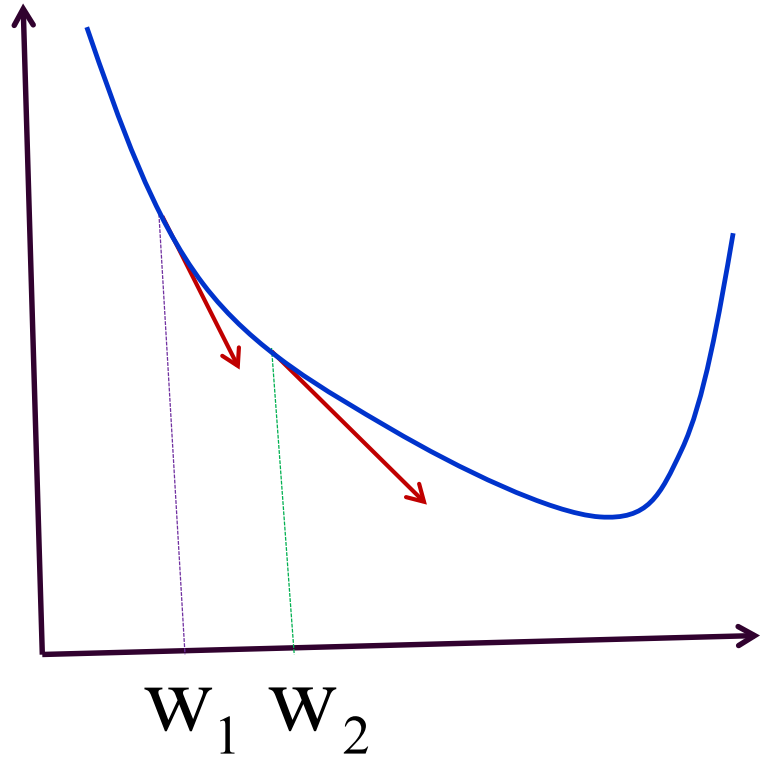
（序列式）梯度下降法(GD)

$$w_{t+1} \leftarrow w_t - \eta_t \left[\frac{1}{n} \sum_{i=1}^{n} \nabla f_i(w_t) \right]$$

- 收敛率：$\mathcal{O}(\rho^T)$ (强凸)

  $\mathcal{O}\left(\frac{1}{T}\right)$ (一般凸)

- 每次迭代开销：$\mathcal{O}(nd)$



经典随机优化方法（序列式）

随机梯度下降法(SGD)

$$w_{t+1} \leftarrow w_t - \eta_t \nabla f_{i_t}(w_t)$$

- 通过随机采样样本梯度取代全梯度
- 收敛率：$\mathcal{O}\left(\frac{1}{T}\right)$ (强凸)

  $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ (一般凸)
- 每次迭代开销：$\mathcal{O}(d)$

小批量随机梯度下降法(mini-batch SGD)

$$w_{t+1} \leftarrow w_t - \frac{\eta_t}{|\mathcal{S}_t|} \sum_{i \epsilon \mathcal{S}_t} \nabla f_i(w_t)$$

直观而常用的数据并行算法

分布式随机梯度下降算法流程(Distributed SGD)：

完全等价于序列式mini-batch SGD

初始化：K个工作节点参与；每个节点保存整个模型的拷贝；数据集$\mathcal{D}$被划分到各个节点上，节点k上的部分数据集记作$\mathcal{D}_k$；

for $t = 0,1,2,\dots,T-1$ do:

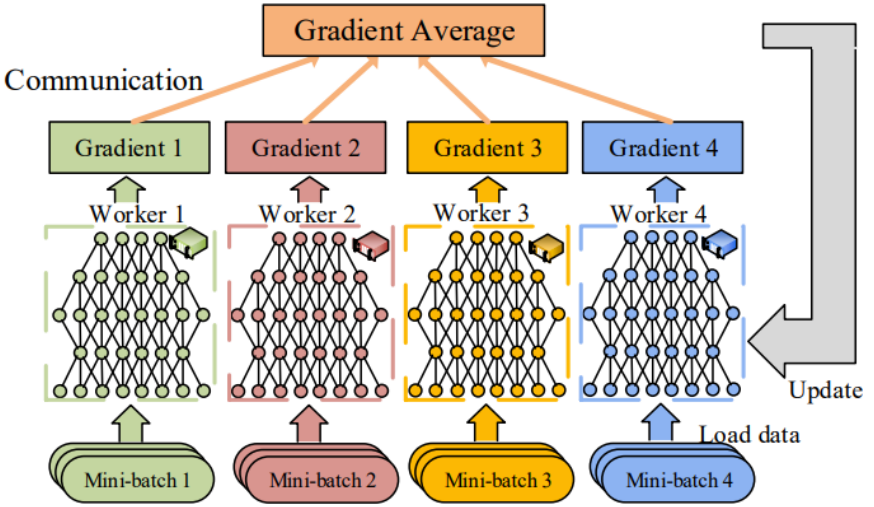
  for $k = 0,1,2,\dots,K-1$ do:

    节点k从本地数据集 $\mathcal{D}_k$随机 采样批量大小为b的数据 $\mathcal{S}_{t,k}$；

    计算随机梯度$g_{t,k} = \frac{1}{b}\sum_{i\in\mathcal{S}_{t,k}} \nabla f_i(w_t)$；

    通信$g_{t,k}$，聚合梯度$g_t = \frac{1}{K}\sum_{k=0}^{K-1} g_{t,k}$；

    进行参数更新$w_{t+1} = w_t - \eta g_t$；



存在的问题：

1. 通信频次高
2. 模型参数量多，单次通信数量巨大
3. 可能存在中心节点通信拥塞
4. 整个大模型单节点存不下、算得慢

数据并行

- 通信频次优化算法
  - 本地学习：LocalSGD、SCOPE、pSCOPE 、OrLoMo
  - 大批量学习：LARS/LAMB、SNGD、SNGM
- 通信数量优化算法
  - 量化/稀疏、EF-signSGD、DGC、GMC
- 通信模式优化算法
  - OrMo、OrLoMo

通信频次优化算法

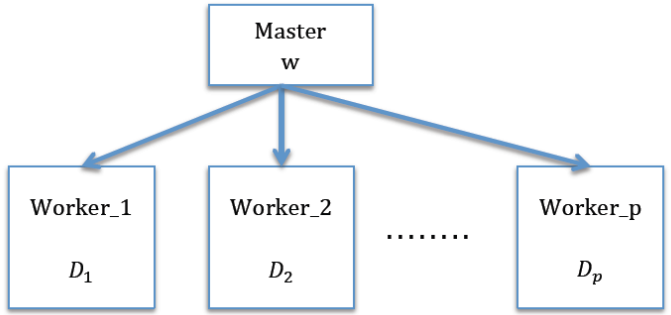
南京大学 NANJING UNIVERSITY

SCOPE

Task of Workers in SCOPE:

Initialization: initialize η and $c > 0$;
For the Worker_k:
for $t = 0, 1, 2, \ldots, T$ **do**
  Wait until it gets the newest parameter $\mathbf{w}_t$ from the Master;
  Let $\mathbf{u}_{k,0} = \mathbf{w}_t$, compute the local gradient sum $\mathbf{z}_k = \sum_{i \in \mathcal{D}_k} \nabla f_i(\mathbf{w}_t)$, and then send $\mathbf{z}_k$ to the Master;
  Wait until it gets the full gradient $\mathbf{z}$ from the Master;
  for $m = 0$ to $M - 1$ **do**
    Randomly pick up an instance with index $i_{k,m}$ from $\mathcal{D}_k$;
    $\mathbf{u}_{k,m+1} = \mathbf{u}_{k,m} - \eta(\nabla f_{i_{k,m}}(\mathbf{u}_{k,m}) - \nabla f_{i_{k,m}}(\mathbf{w}_t) + \mathbf{z} + c(\mathbf{u}_{k,m} - \mathbf{w}_t))$;
  end for
  Send $\mathbf{u}_{k,M}$ or the average of these $\{\mathbf{u}_{k,m}\}$, which is called the locally updated parameter and denoted as $\tilde{\mathbf{u}}_k$, to the Master;
end for

本地学习策略



Shen-Yi Zhao, …, Wu-Jun Li. "SCOPE: Scalable composite optimization for learning on Spark." Proceedings of the AAAI Conference on Artificial Intelligence. 2017.

通信频次优化算法

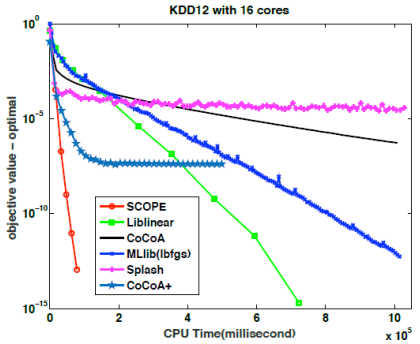
南京大学 NANJING UNIVERSITY

理论分析

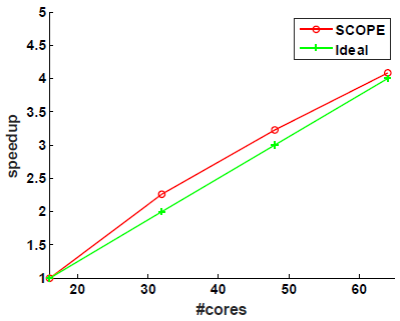
- 线性收敛率（强凸） $\mathbb{E}\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2 \leq (\alpha^M + \frac{\beta}{1-\alpha})\mathbb{E}\|\mathbf{w}_t-\mathbf{w}^*\|^2$
- 通信开销：传统方法：$\mathcal{O}(nT_1)$； SCOPE：$\mathcal{O}(T_2)$

实验效果

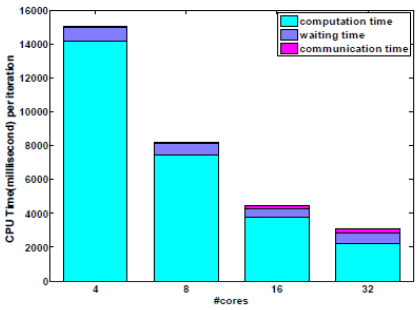
- 在Spark平台上实现
- 7千万样本、百万维特征(稀疏)，8台机器，1分钟内完成LR的训练



(c) KDD12



(e) Speedup



(f) Synchronization cost

Shen-Yi Zhao, …, Wu-Jun Li. "SCOPE: Scalable composite optimization for learning on Spark." Proceedings of the AAAI Conference on Artificial Intelligence. 2017.

南京大学 NANJING UNIVERSITY

实验效果

➢基于参数服务器（Parameter Server）架构

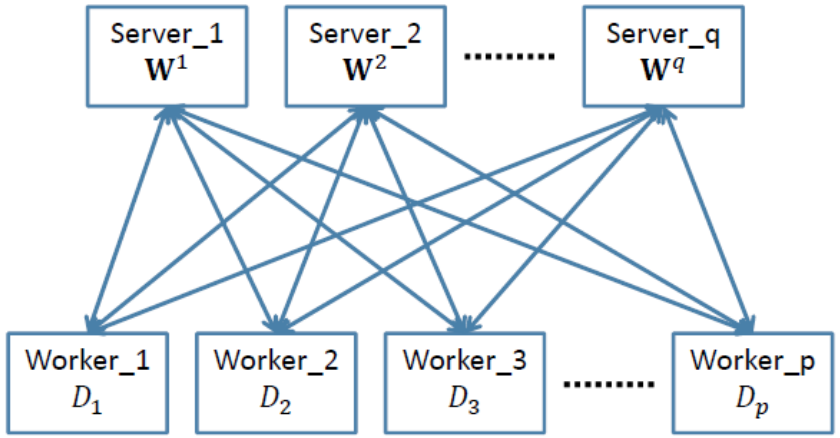
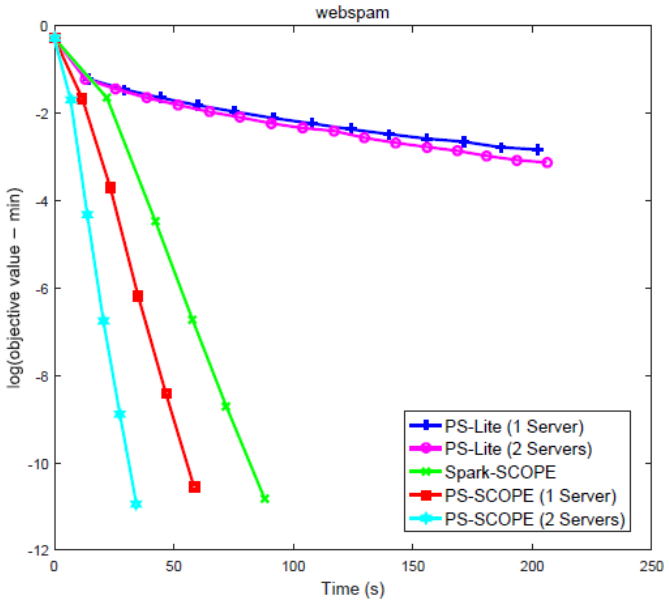


Figure: Distributed framework of PS-SCOPE.



- PS-Lite：基于SSP/ASP实现的参数服务器 [Mu Li, et al., OSDI 2014]
- 比PS-Lite快数十倍到上百倍

Shen-Yi Zhao, …, Wu-Jun Li. "SCOPE: Scalable composite optimization for learning on Spark." Proceedings of the AAAI Conference on Artificial Intelligence. 2017.

Proximal SCOPE (pSCOPE)

Task of the kth worker:
for $t = 0, 1, 2, ...T-1$ **do**
  Wait until receiving $\mathbf{w}_t$ from master;
  Let $\mathbf{u}_{k,0} = \mathbf{w}_t$, calculate $\mathbf{z}_k = \sum_{i \in D_k} f_i(\mathbf{w}_t)$ and send $\mathbf{z}_k$ to master;
  Wait until receiving $\mathbf{z}$ from master;
  for $m = 0, 1, 2, ...M-1$ **do**
    Randomly choose an instance $\mathbf{x}_{i_{k,m}} \in D_k$;
    Calculate $\mathbf{v}_{k,m} = \nabla f_{i_{k,m}}(\mathbf{u}_{k,m}) - \nabla f_{i_{k,m}}(\mathbf{w}_t) + \mathbf{z}$;
    Update $\mathbf{u}_{k,m+1} = \text{prox}_{R,\eta}(\mathbf{u}_{k,m} - \eta \mathbf{v}_{k,m})$;
  end for
  Send $\mathbf{u}_{k,M}$ to master
end for

$$\text{prox}_{R,\eta}(\mathbf{u}) = \arg\min_{\mathbf{v}} (R(\mathbf{v}) + \frac{1}{2\eta}\|\mathbf{v} - \mathbf{u}\|^2)$$

Shen-Yi Zhao, …, Wu-Jun Li. "Proximal SCOPE for distributed sparse learning." Advances in Neural Information Processing Systems. 2018.

通信频次优化算法

南京大学 NANJING UNIVERSITY

理论分析

➢线性收敛率（强凸）

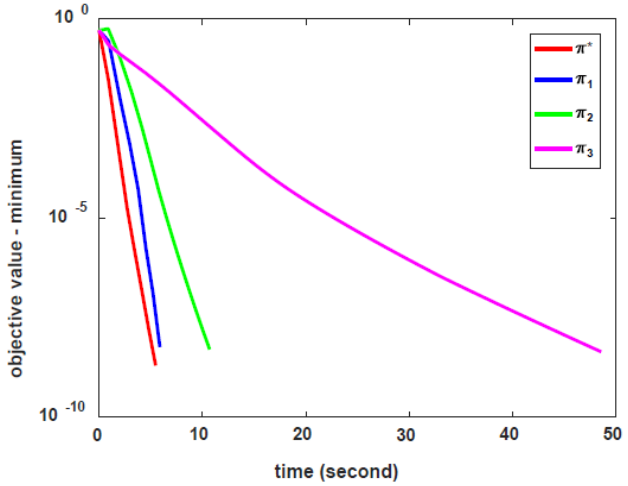
Assume $\pi = [F_1(\cdot), \ldots, F_p(\cdot)]$ *is a* (ϵ, ξ)*-good partition w.r.t.* $P(\cdot)$. *For convenience, we set* $\mu_k = \mu, L_k = L, k = 1, 2 \ldots, p$. *If* $\|\mathbf{w}_t - \mathbf{w}^*\|^2 \geq \epsilon$, *then*

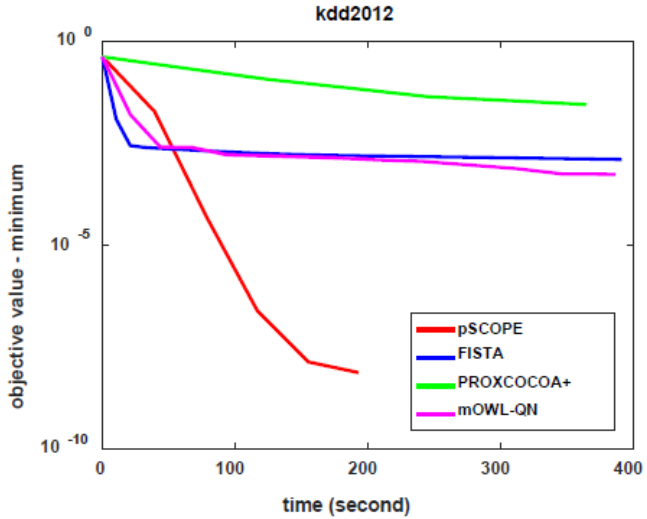
$$\mathbb{E}\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq [(1 - \mu\eta + 2L^2\eta^2)^M + \frac{2L^2\eta + 2\xi}{\mu - 2L^2\eta}]\|\mathbf{w}_t - \mathbf{w}^*\|^2$$

实验效果

kdd2012:
119,705,032样本，
54,686,452特征

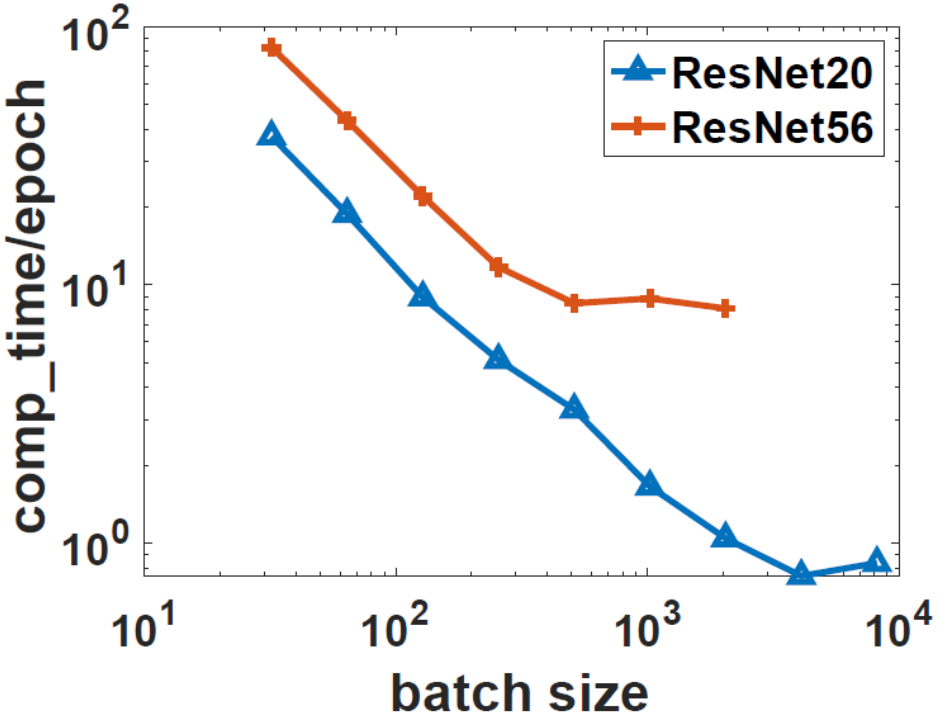
8 Workers





Shen-Yi Zhao, …, Wu-Jun Li. "Proximal SCOPE for distributed sparse learning." Advances in Neural Information Processing Systems. 2018.

- 大批量学习
  - 更有效地利用GPU等硬件
  - 随着机器数的增加，分布式机器学习必须大批量



通信频次优化算法

南京大学 NANJING UNIVERSITY

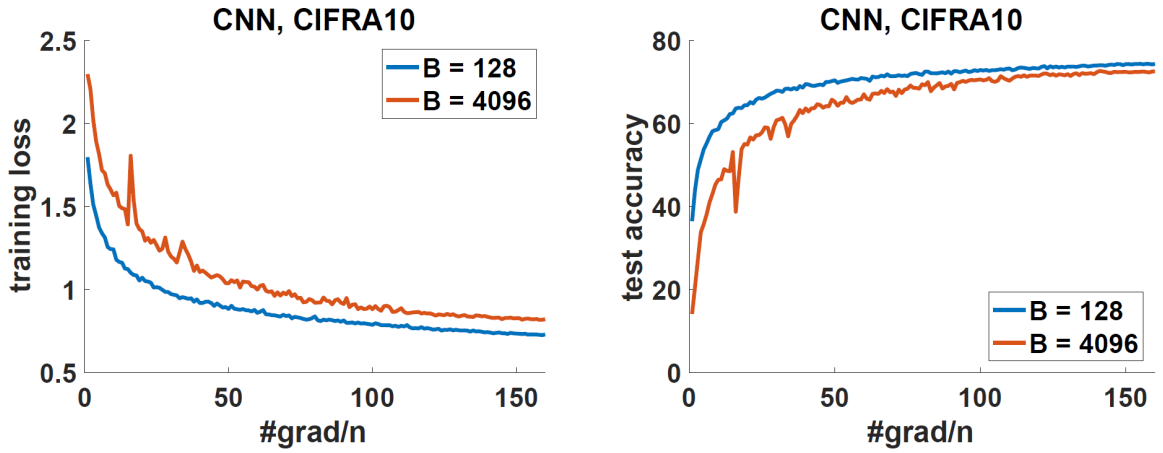
大批量学习

$$w_{t+1} = w_t - \frac{\eta}{Kb}\sum_{k=1}^{K}\sum_{i\epsilon\mathcal{D}_{t,k}}\nabla f_i(w_t)$$

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F(w_t)\|^2 \le \mathcal{O}(\frac{1}{\sqrt{TKb}})$$

K：机器数
b：批量大小
T：通信次数

批量越大
通信次数越少



问题：盲目增大批量会导致模型泛化性能下降
目标：保证模型泛化性能（或梯度有效率）的前提下增大训练批量

通信频次优化算法

Stochastic Normalized Gradient Descent (SNGD) [Hazan, 2015]

$$\boldsymbol{g}_t = \eta \sum_{k=1}^{K} \sum_{i \in \mathcal{D}_{t,k}} \nabla f_i(\boldsymbol{w}_t)/(Kb)$$

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \frac{\boldsymbol{g}_t}{\|\boldsymbol{g}_t\|}$$

归一化梯度可以避免批量大小对算法的影响

- 我们首先证明在non-convex条件下，SNGD收敛并且适用于大批量优化，然后提出SNGD with Momentum (SNGM)。

SNGD with Momentum (SNGM)

$$\boldsymbol{m}_{t+1} = \beta \boldsymbol{m}_t + \frac{\boldsymbol{g}_t}{\|\boldsymbol{g}_t\|} \quad \beta \in [0, 1)$$

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{m}_{t+1}$$

[1] Shen-Yi Zhao, Yin-Peng Xie, and Wu-Jun Li. "On the convergence and improvement of stochastic normalized gradient descent." Science China Information Sciences (SCIS). 2021.

[2] Shen-Yi Zhao, Chang-Wei Shi, Yin-Peng Xie, and Wu-Jun Li. "Stochastic normalized gradient descent with momentum for large batch training." SCIS 2024.

通信频次优化算法

SNGD with Momentum (SNGM)

- 在平滑非凸问题上，SNGM与动量SGD (MSGD)的对比（$B = Kb, \mathcal{C} = TB$）

	SNGM	MSGD [Yu, 2019]
动量计算	$\boldsymbol{m}_{t+1} = \beta \boldsymbol{m}_t + \frac{\boldsymbol{g}_t}{\|\boldsymbol{g}_t\|}$	$\boldsymbol{m}_{t+1} = \beta \boldsymbol{m}_t + \boldsymbol{g}_t$
学习速率	大于0	$\eta \leq (1-\beta)^2/((1+\beta)L)$
收敛速度	$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla P(\boldsymbol{w}_t)\| \leq \mathcal{O}(\frac{1}{\mathcal{C}^{1/4}})$	$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla P(\boldsymbol{w}_t)\| \leq \sqrt{\mathcal{O}(\frac{1}{\sqrt{\mathcal{C}}}) + \mathcal{O}(\frac{B^2}{\mathcal{C}})}$
批量大小	$\sqrt{\mathcal{C}}$	$\mathcal{O}(\min\{\frac{\sqrt{\mathcal{C}}}{L}, \mathcal{C}^{1/4}\})$

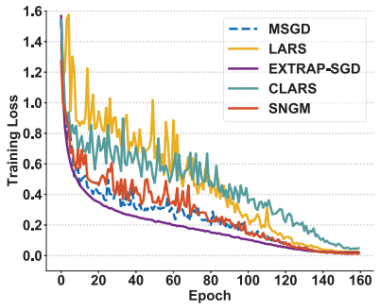
- 通信开销降低两个数量级以上
- 证明Layer-wise normalization（LARS/LAMB采用）将降低收敛速率

Shen-Yi Zhao, Chang-Wei Shi, Yin-Peng Xie, and Wu-Jun Li. "Stochastic normalized gradient descent with momentum for large batch training." SCIS 2024.

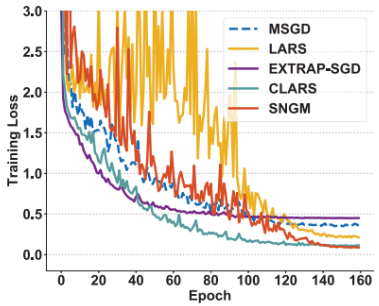
南京大学 NANJING UNIVERSITY

SNGD with Momentum (SNGM)

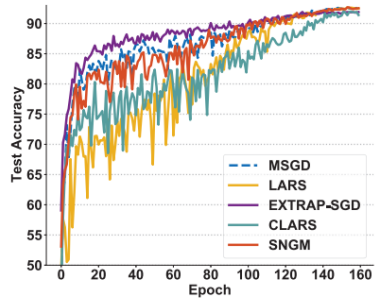
➢图像分类 （左：ResNet20/Cifar10；右上：ViT微调）



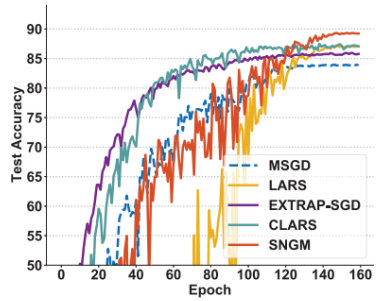
(a) $B = 128$



(b) $B = 8192$



(c) $B = 128$



(d) $B = 8192$

		$B = 128$	$B = 1024$	$B = 4096$	$B = 8192$	$B = 16384$
CIFAR-10	MSGD	98.96%	99.02%	98.93%	98.81%	98.40%
	LARS [34]	**99.08%**	98.97%	98.94%	98.85%	98.36%
	SNGM	99.00%	**99.12%**	**98.98%**	**99.00%**	**98.82%**
CIFAR-100	MSGD	92.91%	**92.99%**	92.86%	91.57%	85.84%
	LARS [34]	92.59%	92.57%	91.80%	92.17%	90.36%
	SNGM	**93.06%**	**92.99%**	**92.93%**	**92.36%**	**90.48%**

Table 3 Training time (second) per epoch of SNGM on the ResNet20/CIFAR-10 training task.

B	128	1024	2048	4096	8192
time/epoch	13.95	2.15	1.78	1.6	1.49

Shen-Yi Zhao, Chang-Wei Shi, Yin-Peng Xie, and Wu-Jun Li. "Stochastic normalized gradient descent with momentum for large batch training." SCIS 2024.

SNGD with Momentum (SNGM)

NLP

Table 6 Test perplexity results on WikiTex-2.

	$B = 20$	$B = 1000$	$B = 2000$
MSGD	**113.26**	114.34	118.50
LARS	115.71	116.29	119.35
SNGM	113.74	**112.90**	**115.65**

Table 7 Training time (second) per epoch of SNGM on the LSTM/Wikitext-2 training task.

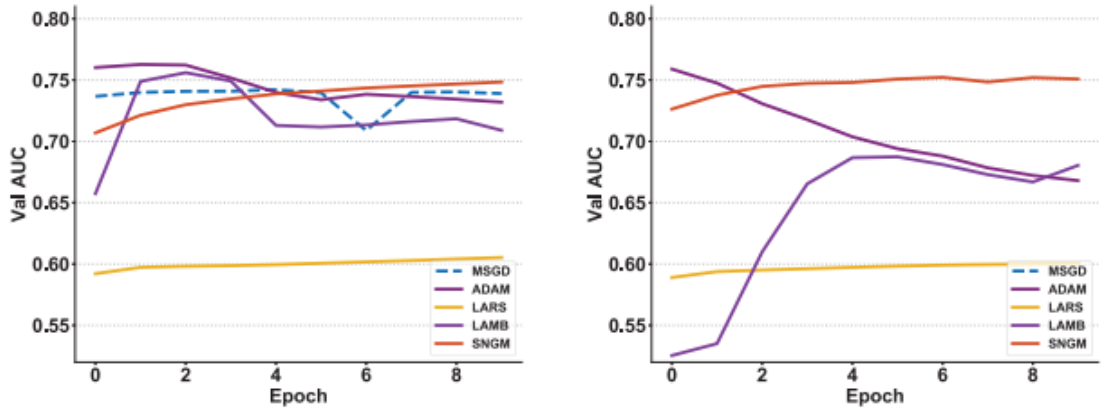
B	20	1000	2000
time/epoch	69.8	24.44	15.22

Shen-Yi Zhao, Chang-Wei Shi, Yin-Peng Xie, and Wu-Jun Li. "Stochastic normalized gradient descent with momentum for large batch training." SCIS 2024.

SNGD with Momentum (SNGM)

➢点击率预测

	MSGD	ADAM	LARS	LAMB	SNGM
$B = 1024$	0.7397	0.7311	0.6065	0.7066	**0.7489**
$B = 8192$	0.5	0.6666	0.6006	0.6811	**0.7514**



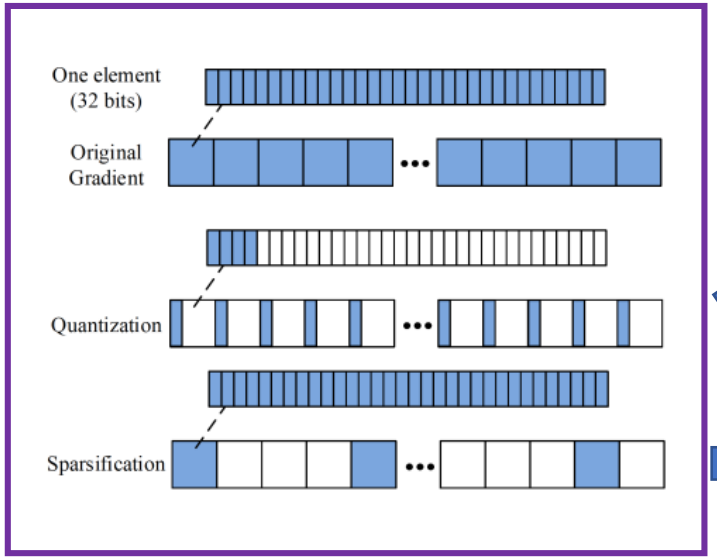
(a) $B = 1024$ (b) $B = 8192$

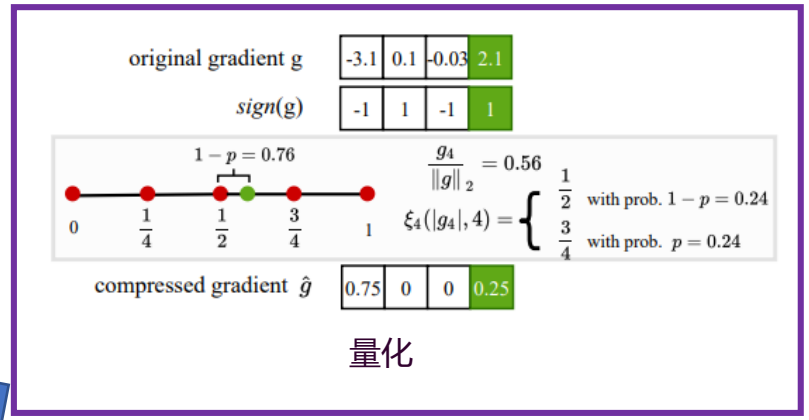
Shen-Yi Zhao, Chang-Wei Shi, Yin-Peng Xie, and Wu-Jun Li. "Stochastic normalized gradient descent with momentum for large batch training." SCIS 2024.

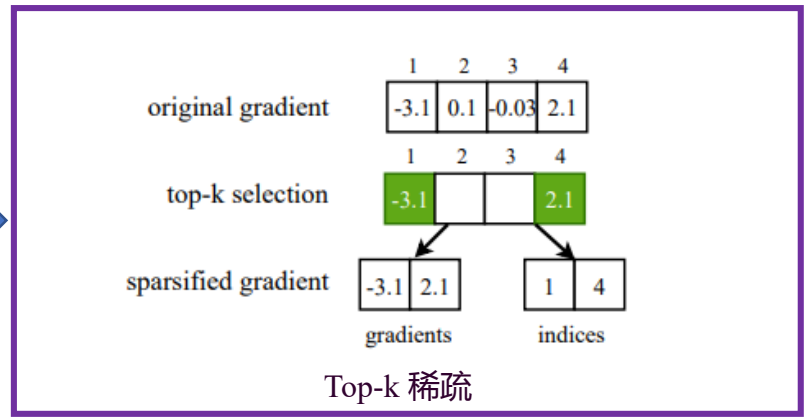
通信数量优化算法

量化/稀疏

- 量化：将浮点数转化成低比特表示
- 稀疏：仅通信部分维度







Zhenheng Tang, et al. "Communication-efficient distributed deep learning: A comprehensive survey." arXiv preprint arXiv:2003.06307 (2020).

Global Momentum Compression (GMC)

Algorithm 1 GMC

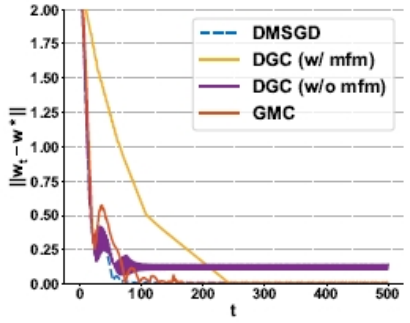
Input: sparsification compressor $\mathcal{C}(\cdot)$, number of workers K, number of iterations T, model parameters $\mathbf{w}_0$, learning rate η, momentum coefficient $\beta \in [0,1)$, training dataset $\mathcal{D}_k, \forall k \in [K]$;
Set $\mathbf{w}_{-1} = \mathbf{w}_0, \mathbf{e}_{0,k} = \mathbf{0}, \forall k \in [K]$;
for iteration $t \in [T]$ **do**
  Workers:
  for worker $k \in [K]$ parallelly **do**
    Randomly pick a mini-batch of training data $\mathcal{I}_{t,k} \subseteq \mathcal{D}_k$ with $|\mathcal{I}_{t,k}| = b$ and compute $\nabla f(\mathbf{w}_t; \mathcal{I}_{t,k}) = \frac{1}{b} \sum_{\xi \in \mathcal{I}_{t,k}} \nabla f(\mathbf{w}_t; \xi)$;
    $\mathbf{e}_{t+\frac{1}{2},k} = \mathbf{e}_{t,k} + \nabla f(\mathbf{w}_t; \mathcal{I}_{t,k}) - \frac{\beta}{\eta}(\mathbf{w}_t - \mathbf{w}_{t-1})$;
    Generate a sparse vector $\mathcal{C}(\mathbf{e}_{t+\frac{1}{2},k})$ and send $\mathcal{C}(\mathbf{e}_{t+\frac{1}{2},k})$ to the server;
    Update the error residual $\mathbf{e}_{t+1,k} = \mathbf{e}_{t+\frac{1}{2},k} - \mathcal{C}(\mathbf{e}_{t+\frac{1}{2},k})$;
    Receive $\mathbf{w}_{t+1} - \mathbf{w}_t$ from server;
    Get $\mathbf{w}_{t+1}$ by $\mathbf{w}_{t+1} = \mathbf{w}_t + (\mathbf{w}_{t+1} - \mathbf{w}_t)$;
  end for
  Server:
  Receive $\mathcal{C}(\mathbf{e}_{t+\frac{1}{2},k})$ from all the workers;
  $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \frac{1}{K} \sum_{k \in [K]} \mathcal{C}(\mathbf{e}_{t+\frac{1}{2},k})$;
  Send $\mathbf{w}_{t+1} - \mathbf{w}_t$ to workers;
end for

Chang-Wei Shi, Shen-Yi Zhao, …, Wu-Jun Li. "Global momentum compression for sparse communication in distributed SGD." arXiv 2019, arXiv 2024.

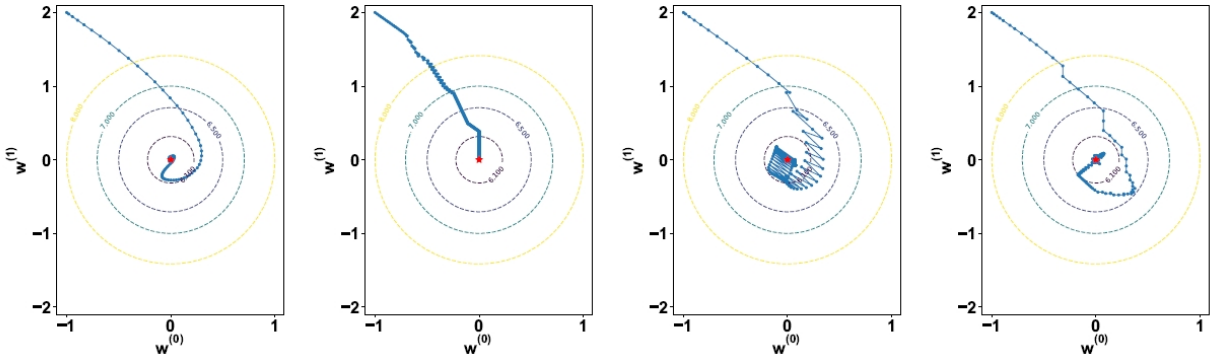
南京大学 NANJING UNIVERSITY

Global Momentum Compression (GMC)

- 本地目标函数（两个worker）
  - Worker 1: $F_0(\mathbf{w}) = f(\mathbf{w};\xi_1) = \sum_{i\in[d]}(d-i)*[w^{(i)}-(i+1)]^2$ ；极小值点$x = [1,2,\dots,d]^T$
  - Worker 2: $F_1(\mathbf{w}) = f(\mathbf{w};\xi_2) = \sum_{i\in[d]}(d-i)*[w^{(i)}+(i+1)]^2$ ；极小值点$x = [-1,-2,\dots,-d]^T$
- 全局目标函数：$F = \frac{F_0+F_1}{2}$；全局极小值点$x = [\underbrace{0,0,\dots,0}_{d}]^T$
- $(\mathrm{d}=2)$



(a) d=2, s=0.5d

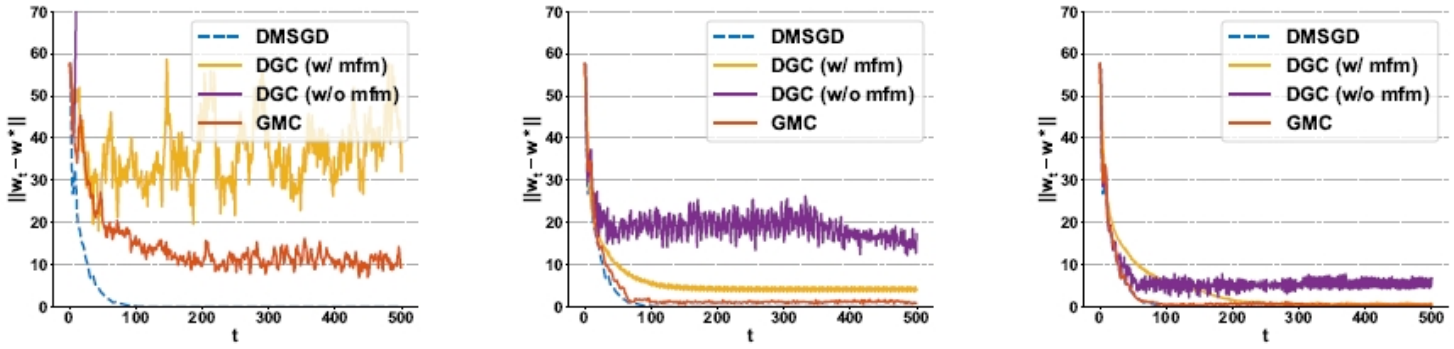


(a) DMSGD (b) DGC (w/ mfm) (c) DGC (w/o mfm) (d) GMC

Chang-Wei Shi, Shen-Yi Zhao, …, Wu-Jun Li. "Global momentum compression for sparse communication in distributed SGD." arXiv 2019, arXiv 2024.

Global Momentum Compression (GMC)

- 本地目标函数（两个worker）
  - Worker 1: $F_0(\mathbf{w}) = f(\mathbf{w};\xi_1) = \sum_{i\in[d]}(d-i)*[w^{(i)}-(i+1)]^2$ ；极小值点$x = [1,2,\dots,d]^T$
  - Worker 2: $F_1(\mathbf{w}) = f(\mathbf{w};\xi_2) = \sum_{i\in[d]}(d-i)*[w^{(i)}+(i+1)]^2$ ；极小值点$x = [-1,-2,\dots,-d]^T$
- 全局目标函数：$F = \frac{F_0+F_1}{2}$；全局极小值点$x = [\underbrace{0,0,\dots,0}_{d}]^T$
- $(\mathrm{d} = 20)$



(b) d=20, s=0.1d　(c) d=20, s=0.5d　(d) d=20, s=0.8d

Chang-Wei Shi, Shen-Yi Zhao, …, Wu-Jun Li. "Global momentum compression for sparse communication in distributed SGD." arXiv 2019, arXiv 2024.

南京大学 NANJING UNIVERSITY

Global Momentum Compression (GMC)

➢理论分析

Lemma 3 *Let* $\mathbf{z}_t \triangleq \mathbf{w}_t + \frac{\beta}{1-\beta}(\mathbf{w}_t - \mathbf{w}_{t-1}) - \frac{\eta}{1-\beta}\bar{\mathbf{e}}_t$, *then we have:*

$$\mathbf{z}_{t+1} = \mathbf{z}_t - \frac{\eta}{1-\beta}\nabla f(\mathbf{w}_t; \mathcal{I}_t).$$

Lemma 4 *With Assumption 1, 2, if* $\beta \leq \frac{\delta}{4\sqrt{2+\delta}}$, *the error residual can be bounded:*

$$\frac{1}{K}\sum_{k\in[K]} \mathbb{E}\|\mathbf{e}_{t,k}\|^2 \leq E^2,$$

where $E^2 = \frac{(1-\delta)(1+\frac{4}{\delta})G^2}{1-[(1-\frac{\delta}{4})(1-\frac{\beta}{K})^2+(1+\frac{\delta}{4})(1+\frac{2}{\delta})\beta^2]}$.

Lemma 5 *The gap between* $\mathbf{z}_t$ *and* $\mathbf{w}_t$ *can be bounded:* $\mathbb{E}\|\mathbf{z}_t - \mathbf{w}_t\|^2 \leq C_1\eta^2$, *where* $C_1 = \frac{6(2E^2+G^2)\beta^2}{(1-\beta)^4} + \frac{2E^2}{(1-\beta)^2}$.

Theorem 6 *With Assumptions 1, 2, 3 and 4, if* $\beta \leq \frac{\delta}{4\sqrt{2+\delta}}$ *and* $\eta \leq \frac{1-\beta}{2L}$, *Algorithm 1 has the following convergence rate:*

$$\frac{1}{T}\sum_{t\in[T]} \mathbb{E}\|\nabla F(\mathbf{w}_t)\|^2 \leq \frac{4(1-\beta)(F(\mathbf{z}_0)-F^*)}{T\eta} + \frac{2L\sigma^2}{(1-\beta)b}\frac{\eta}{K} + 2C_1L^2\eta^2,$$

where $C_1 = \frac{6(2E^2+G^2)\beta^2}{(1-\beta)^4} + \frac{2E^2}{(1-\beta)^2}$.

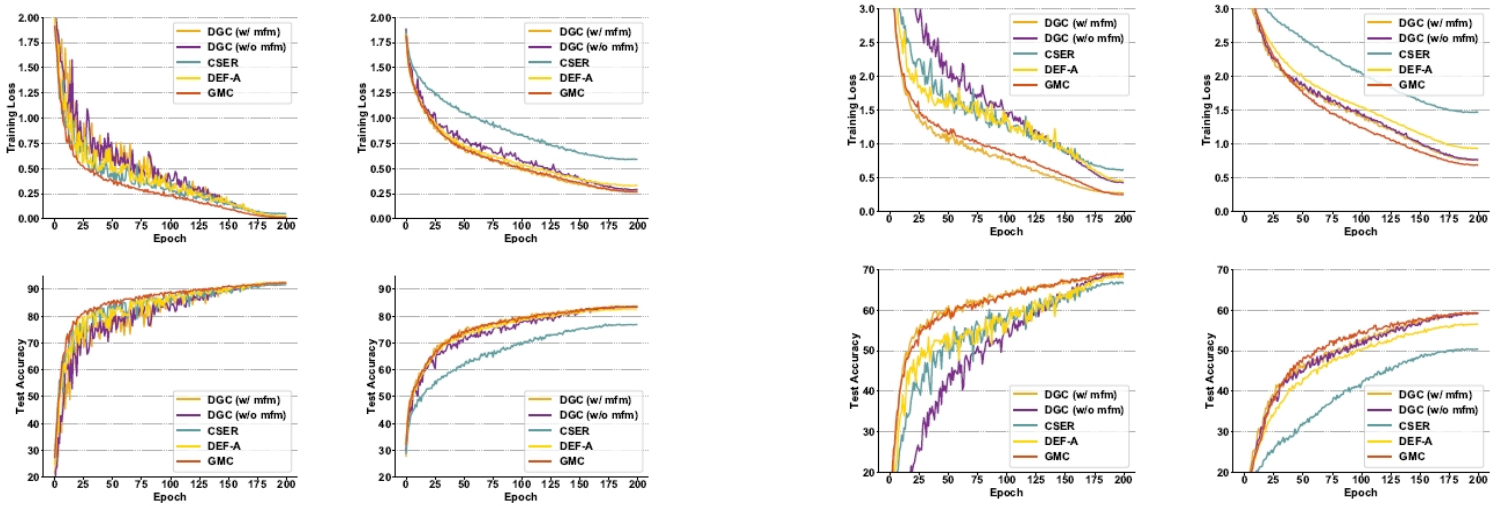
Chang-Wei Shi, Shen-Yi Zhao, …, Wu-Jun Li. "Global momentum compression for sparse communication in distributed SGD." arXiv 2019, arXiv 2024.

Global Momentum Compression (GMC)

- ResNet20/CIFAR10训练
- 通信0.1%, top-k
- IID data distribution

Table 1: Empirical results of different methods under IID data distribution.

Dataset	Model	Method	DGC (w/ mfm)	DGC (w/o mfm)	CSER	DEF-A	GMC
CIFAR10	ResNet20 (BN)	Accuracy	**92.44%**	92.30%	91.70%	92.29%	92.30%
		RCC	0.66%	0.66%	0.62%	0.65%	0.64%
	ViT	Accuracy	**83.67%**	83.35%	76.81%	82.67%	83.46%
		RCC	0.82%	0.81%	0.62%	0.81%	0.80%
CIFAR100	ResNet20 (BN)	Accuracy	68.05%	68.66%	66.62%	68.72%	**68.89%**
		RCC	0.64%	0.64%	0.65%	0.64%	0.64%
	ViT	Accuracy	**59.28%**	59.15%	50.35%	56.56%	**59.28%**
		RCC	0.80%	0.79%	0.65%	0.79%	0.78%



(a) ResNet20, CIFAR10 (b) ViT, CIFAR10 (c) ResNet20, CIFAR100 (d) ViT, CIFAR100

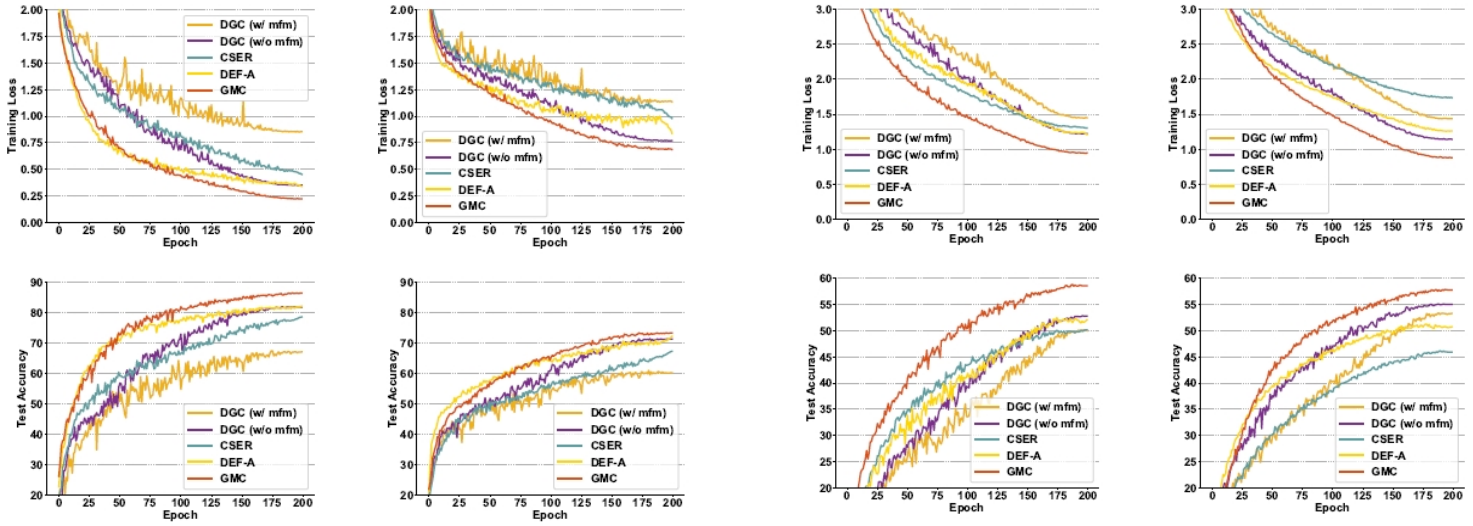
Chang-Wei Shi, Shen-Yi Zhao, …, Wu-Jun Li. "Global momentum compression for sparse communication in distributed SGD." arXiv 2019, arXiv 2024.

Global Momentum Compression (GMC)

- ResNet20/CIFAR10训练
- 通信0.1%, top-k
- Non-IID data distribution

Table 2: Empirical results of different methods under non-IID data distribution.

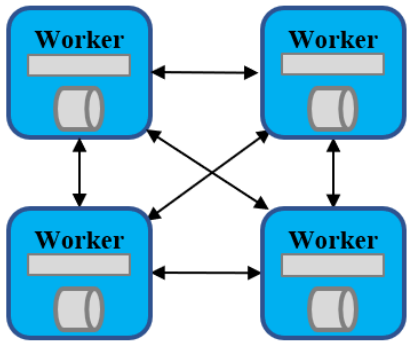
Dataset	Model	Method	DGC (w/ mfm)	DGC (w/o mfm)	CSER	DEF-A	GMC
CIFAR10	ResNet20 (GN)	Accuracy	67.13%	81.93%	78.65%	82.15%	**86.51%**
		RCC	0.65%	0.65%	0.63%	0.65%	0.64%
	ViT	Accuracy	60.24%	71.35%	67.35%	72.25%	**73.34%**
		RCC	0.81%	0.81%	0.67%	0.81%	0.81%
CIFAR100	ResNet20 (GN)	Accuracy	50.02%	52.79%	50.17%	52.07%	**58.59%**
		RCC	0.64%	0.64%	0.65%	0.64%	0.63%
	ViT	Accuracy	53.29%	55.00%	45.91%	50.74%	**57.77%**
		RCC	0.79%	0.80%	0.70%	0.79%	0.79%



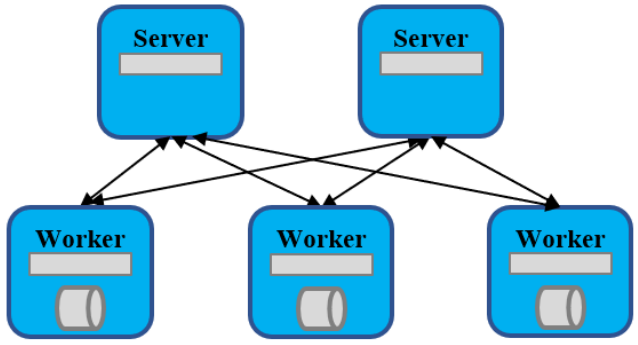
(a) ResNet20, CIFAR10 (b) ViT, CIFAR10 (c) ResNet20, CIFAR100 (d) ViT, CIFAR100

Chang-Wei Shi, Shen-Yi Zhao, …, Wu-Jun Li. "Global momentum compression for sparse communication in distributed SGD." arXiv 2019, arXiv 2024.

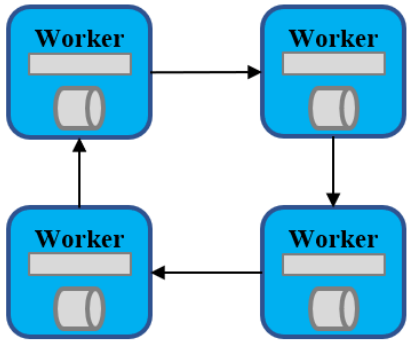
通信模式优化算法



AllReduce



Parameter Server (PS)



RingAllReduce (RA)

去中心化通信架构

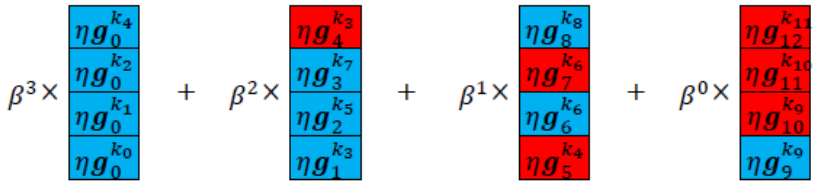
- 目前主流的机器学习平台主要采用Parameter Server (PS)或AllReduce实现分布式机器学习中不同机器之间的通信
- AllReduce通信开销大；PS采用中心化通信模式，中心Server存在通信拥塞问题
- 去中心化通信模式和算法
  - 越来越受到关注，但存在很多挑战
- 同步 vs 异步

通信模式优化算法

南京大学 NANJING UNIVERSITY

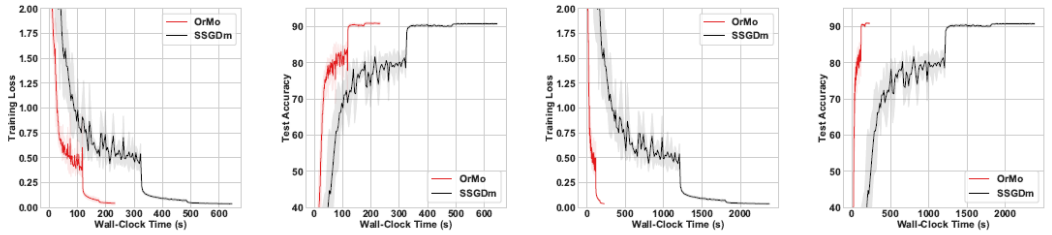
OrMo: 带动量的异步分布式SGD算法

- 动量被证明能提升机器学习算法的速度和精度，但在异步分布式算法中简单加入动量会损失精度，目前还没有好的解决方案；
- 提出了异步分布式算法中有序动量的概念，并设计了基于有序动量的异步分布式算法OrMo。理论证明了收敛性，并首次建立了带动量的异步分布式算法收敛率不依赖于延迟上界的理论。实验验证了有效性。论文发表于NeurIPS 2024。



$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla F(\mathbf{w}_t)\|^2 \leq \mathcal{O}\left(\frac{\sigma}{\sqrt{T}}+\frac{K^{\frac{2}{3}}}{T^{\frac{2}{3}}}+\frac{K}{T}\right)$$

Number of Workers	16 (homogeneous)		64 (homogeneous)		16 (heterogeneous)		64 (heterogeneous)	
Methods	Training Loss	Test Accuracy	Training Loss	Test Accuracy	Training Loss	Test Accuracy	Training Loss	Test Accuracy
ASGD	0.51 ± 0.01	66.16 ± 0.36	0.96 ± 0.03	61.61 ± 0.59	0.51 ± 0.01	65.94 ± 0.39	0.95 ± 0.03	61.74 ± 0.30
naive ASGDm	0.54 ± 0.01	65.46 ± 0.20	1.03 ± 0.05	59.96 ± 0.90	0.53 ± 0.00	65.69 ± 0.42	0.97 ± 0.06	61.13 ± 1.02
shifted momentum	0.47 ± 0.01	66.37 ± 0.14	0.82 ± 0.01	63.55 ± 0.32	0.47 ± 0.00	66.28 ± 0.14	0.82 ± 0.04	63.28 ± 0.66
SMEGA^2	$\mathbf{0.41 \pm 0.00}$	67.32 ± 0.22	0.69 ± 0.00	64.16 ± 0.12	$\mathbf{0.40 \pm 0.01}$	67.29 ± 0.16	0.68 ± 0.02	64.12 ± 0.53
OrMo	$\mathbf{0.41 \pm 0.01}$	$\mathbf{67.56 \pm 0.34}$	$\mathbf{0.56 \pm 0.00}$	$\mathbf{65.48 \pm 0.17}$	$\mathbf{0.40 \pm 0.01}$	$\mathbf{67.71 \pm 0.33}$	$\mathbf{0.58 \pm 0.02}$	$\mathbf{65.43 \pm 0.35}$



(a) homogeneous (b) heterogeneous

Chang-Wei Shi, Yi-Rui Yang, Wu-Jun Li. Ordered Momentum for Asynchronous SGD. NeurIPS 2024.

通信模式优化算法

南京大学 NANJING UNIVERSITY

OrLoMo: 带动量的本地迭代异步分布式SGD算法

- 本地迭代是分布式训练中常用的通过减少通信频次来提升训练效率的方法。但还没有工作研究带动量的本地迭代异步分布式SGD算法。
- 设计了首个带动量的本地迭代异步分布式SGD算法OrLoMo，在OrMo的基础上，进一步减少了通信频次。建立了非凸问题上收敛性不依赖于延迟上界的理论。实验表明，OrLoMo在异构集群中的训练速度可以比同步算法提升2倍以上，即使是在同构集群中，其训练速度也快于相应的同步算法。论文发表于AAAI 2026。

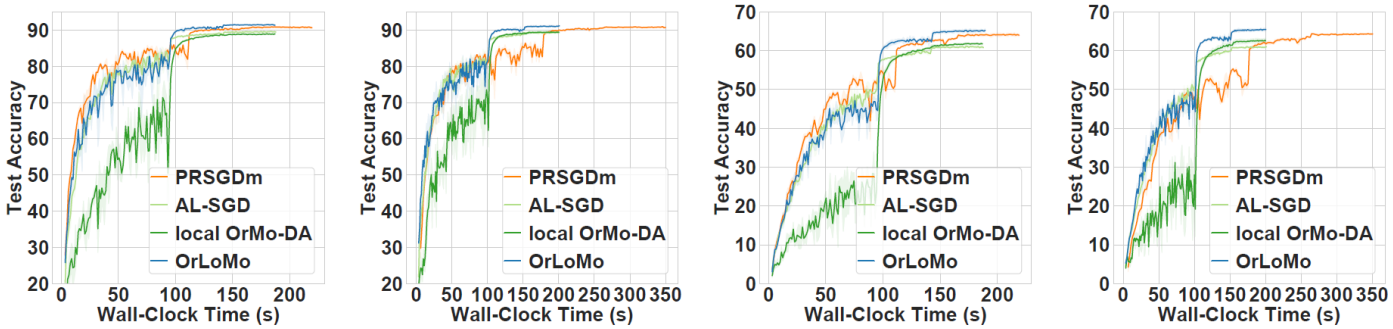
Theorem 1. *With Assumptions 1, 2 and 3, letting* $16LS\gamma \leq (1-\beta)^2$ *and*

$$\eta_t = \begin{cases} \frac{1}{K} & \tau_t \leq 2K, \\ \frac{1}{\tau_t} & \tau_t > 2K, \end{cases} \quad (6)$$

OrLoMo in Algorithm 2 has the following convergence rate:

$$\mathbb{E}\left\|\nabla F(\bar{\mathbf{w}}_T)\right\|^2 \leq \frac{4K(1-\beta)\left(F(\mathbf{w}_0)-F^*\right)}{\gamma ST} + \frac{4\gamma L\sigma^2}{K(1-\beta)^2} + \frac{\gamma^2 L^2(S-1)\sigma^2}{(1-\beta)^2},$$

where $\mathbb{E}\left\|\nabla F(\bar{\mathbf{w}}_T)\right\|^2 = \frac{\sum_{k\in[K]}\hat{\eta}_{0,k}\|\nabla F(\mathbf{w}_0)\|^2}{\sum_{k\in[K]}\hat{\eta}_{0,k}+\sum_{t=1}^{T-1}\hat{\eta}_{t,k_{t-1}}} + \frac{\sum_{t=1}^{T-1}\hat{\eta}_{t,k_{t-1}}\mathbb{E}\|\nabla F(\mathbf{w}_t)\|^2}{\sum_{k\in[K]}\hat{\eta}_{0,k}+\sum_{t=1}^{T-1}\hat{\eta}_{t,k_{t-1}}}.$



(a) homogeneous (CIFAR10) (b) heterogeneous (CIFAR10) (c) homogeneous (CIFAR100) (d) heterogeneous (CIFAR100)

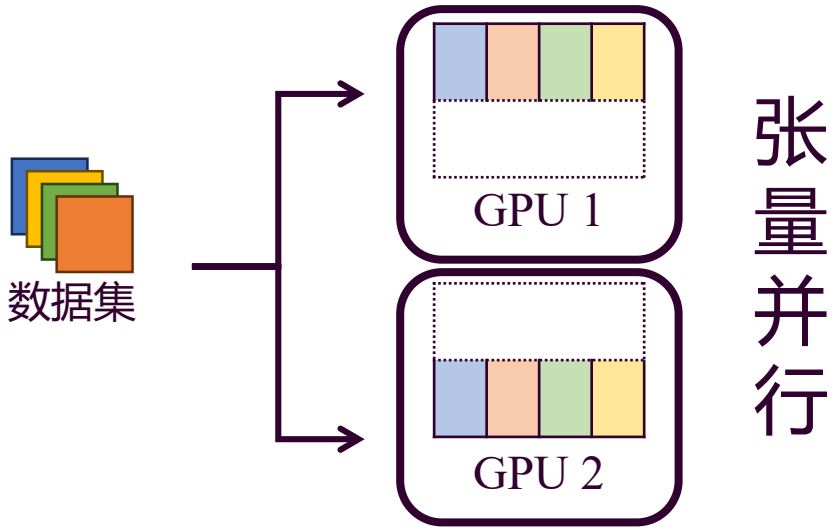
Chang-Wei Shi, Shi-Shang Wang, Wu-Jun Li. Ordered Local Momentum for Asynchronous Distributed Learning under Arbitrary Delays. AAAI 2026.

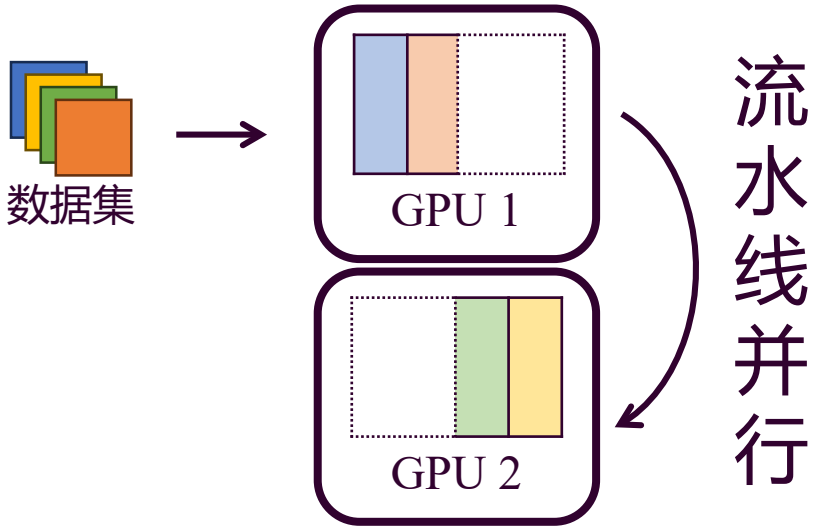
分布式优化

- 数据并行
- 模型并行
- 混合并行
- 自动并行

模型并行

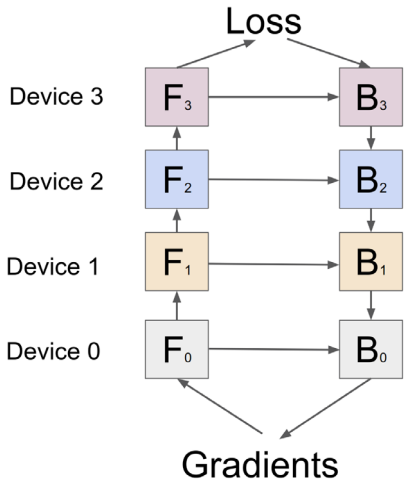
- 切分模型，分配给参与训练的各设备
- 分类
  - 张量并行：以权重张量为单位切分模型，训练数据的粒度为mini-batch
  - 流水线并行：以层为单位切分模型，训练数据的粒度为micro-batch

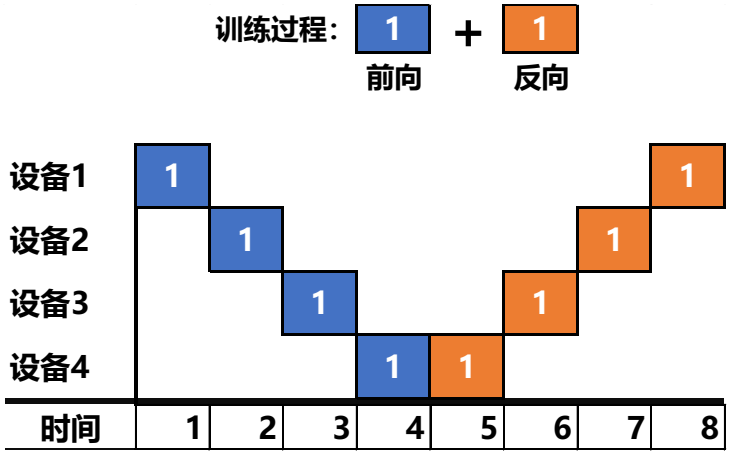




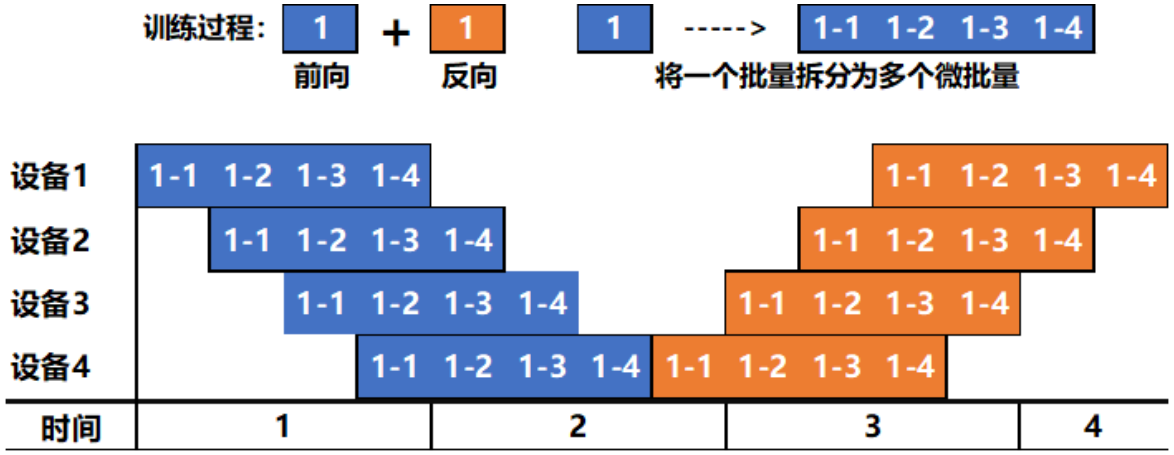
流水线并行

- 以层为单位切分模型（Inter-layer model parallel），将每个mini-batch的样本分为更小的micro-batch，从而组装成流水线进行训练，提升训练效率
- 分类：
  - 同步流水线并行：一个mini-batch的权重同步更新
  - 异步流水线并行：一个mini-batch的权重异步更新





原始模型并行



流水线模型并行

分布式优化

- 数据并行
- 模型并行
- 混合并行
- 自动并行

混合并行

DeepSpeed ZeRO

- P_{os}：设数据并行度为N_d，将优化器状态(optimizer states)切分为N_d个分片，第i个节点仅更新第i个分片的优化器状态和参数，每次更新后使用all-gather通信更新所有参数
- P_g：每个节点就对自己所需那部分参数分片的gradient做reduce，释放其它gradient占用的显存，等价于一次reduce-scatter通信
- P_p：每个节点只存储自己所需的参数分片，需要全量weight时用all-gather通信获取

- Sharded Dada Parallel (SDP)： P_{os+g}
- Fully Sharded Dada Parallel (FSDP)： P_{os+g+p}

[1] Samyam Rajbhandari et. al., ZeRO: Memory Optimizations toward Training Trillion Parameter Models, In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2020.
[2] Samyam Rajbhandari, et. al., . ZeRO-Infinity: Breaking the GPU Memory Wall for Extreme Scale Deep Learning. arXiv preprint arXiv:2104.07857, 2021.

DeepSpeed ZeRO

DP	7.5B Model (GB)			128B Model (GB)			1T Model (GB)		
	P_{os}	P_{os+g}	P_{os+g+p}	P_{os}	P_{os+g}	P_{os+g+p}	P_{os}	P_{os+g}	P_{os+g+p}
1	120	120	120	2048	2048	2048	16000	16000	16000
4	52.5	41.3	**30**	896	704	512	7000	5500	4000
16	35.6	**21.6**	7.5	608	368	128	4750	2875	1000
64	**31.4**	16.6	1.88	536	284	**32**	4187	2218	250
256	30.4	15.4	0.47	518	263	8	4046	2054	62.5
1024	30.1	15.1	0.12	513	257	2	4011	2013	**15.6**

Table 1: Per-device memory consumption of different optimizations in *ZeRO*-DP as a function of DP degree . Bold-faced text are the combinations for which the model can fit into a cluster of 32GB V100 GPUs.

[1] Samyam Rajbhandari et. al., ZeRO: Memory Optimizations toward Training Trillion Parameter Models, In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2020.

混合并行

DeepSpeed ZeRO

- 优点：可以训练非常大规模的模型
- 缺点：通信开销大，训练效率不高

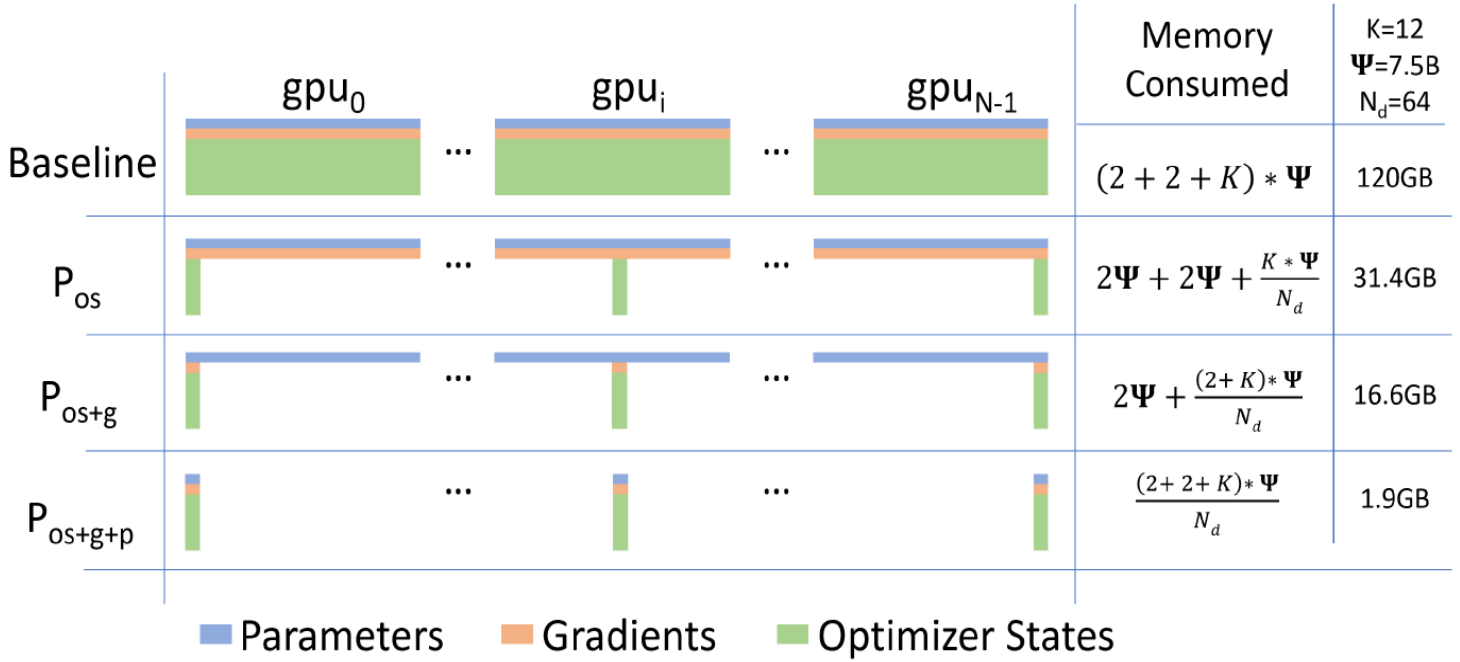
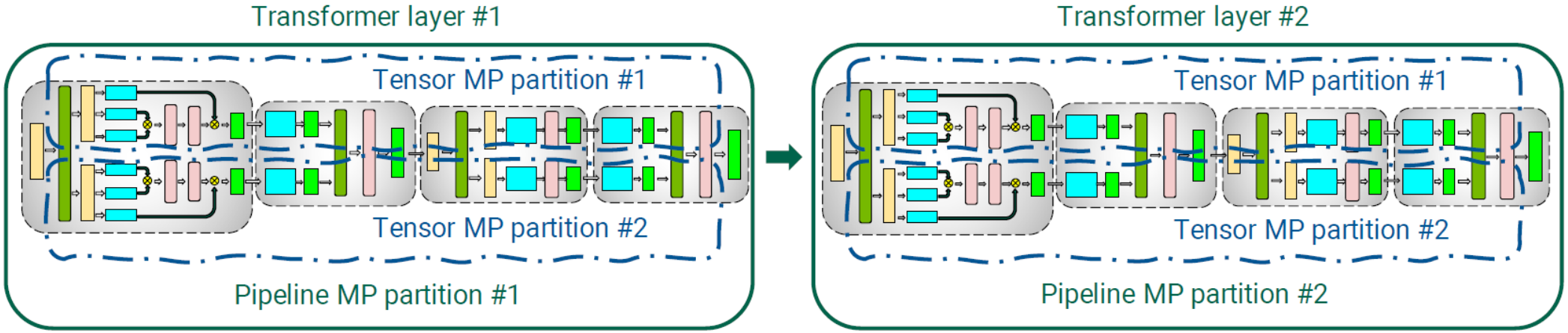
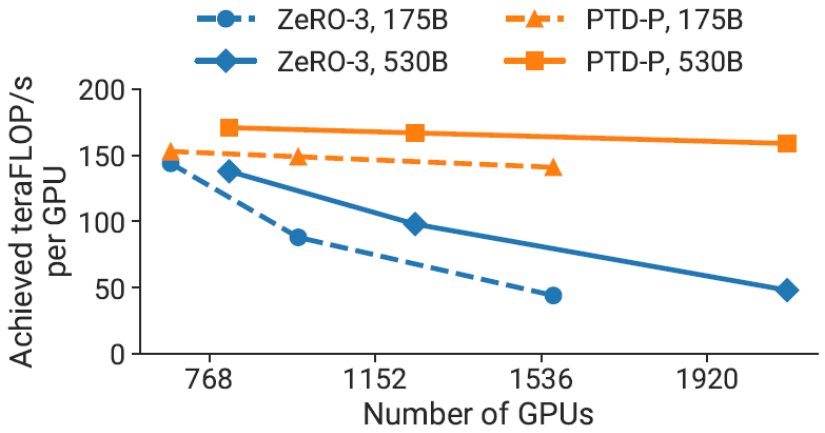


Figure 1: Comparing the per-device memory consumption of model states, with three stages of $ZeRO$-DP optimizations. Ψ denotes model size (number of parameters), K denotes the memory multiplier of optimizer states, and N_d denotes DP degree. In the example, we assume a model size of $\Psi = 7.5B$ and DP of $N_d = 64$ with $K = 12$ based on mixed-precision training with Adam optimizer.

[1] Samyam Rajbhandari et. al., ZeRO: Memory Optimizations toward Training Trillion Parameter Models, In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2020.
[2] Samyam Rajbhandari, et. al., . ZeRO-Infinity: Breaking the GPU Memory Wall for Extreme Scale Deep Learning. arXiv preprint arXiv:2104.07857, 2021.

Megatron-LM： PTD-P (pipeline, tensor, and data parallelism)





Deepak Narayanan 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, 2021.

分布式优化

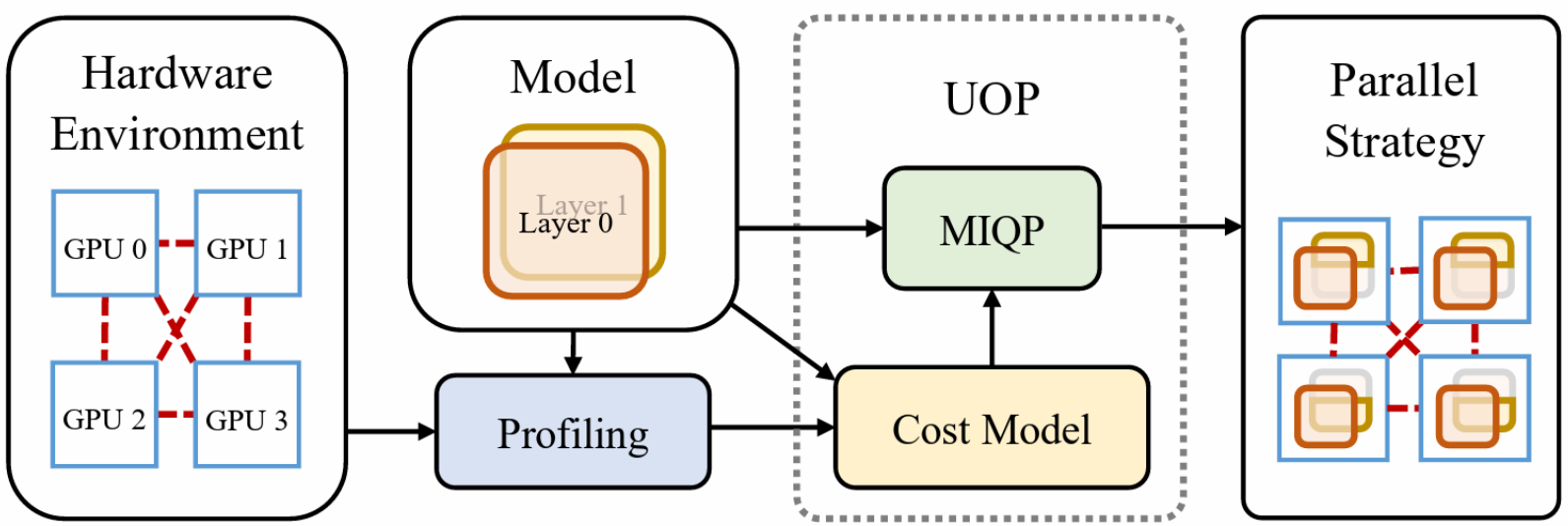
- 数据并行
- 模型并行
- 混合并行
- 自动并行

自动并行

- 已有自动并行方法
  - 启发式搜索：FlexFlow, …
  - 整数规划：DNN-Partitioning, Colossal-Auto, …
  - 动态规划：Galvatron, Piper, DAPPLE, Pipedream, …
  - 混合方法：Alpa, …

UniAP: Unifying Inter- and Intra-Layer Automatic Parallelism by MIQP

- 现有方法用分层的方法搜索自动并行策略，错失最优的并行策略
- 利用混合整数二次规划统一解决DP+TP+PP+FSDP的策略空间下的自动并行策略搜索问题



最佳论文奖候选

Hao Lin, Ke Wu, Jie Li, Jun Li, Wu-Jun Li. UniAP: Unifying Inter- and Intra-Layer Automatic Parallelism by Mixed Integer Quadratic Programming. CVPR 2025.

UniAP的代价模型 (Cost Model)

- 时间模型
  - 根据每个sample的前向时间估计每一层的执行时间
  - 根据张量的重排布通信量和all-reduce通信效率估计节点内的通信时间
  - 根据张量的重排布通信量和P2P通信效率估计跨节点的通信时间
  - 引入计算-通信重叠因子，减少估计误差
- 显存模型
  - 根据张量的形状、数据类型和批量大小估计显存开销
  - 额外考虑上下文显存等其它类型的显存开销，减少估计误差

Hao Lin, Ke Wu, Jie Li, Jun Li, Wu-Jun Li. UniAP: Unifying Inter- and Intra-Layer Automatic Parallelism by Mixed Integer Quadratic Programming. CVPR 2025.

UniAP的混合整数二次规划 (MIQP)

$$\min \quad tpi = \sum_{i=1}^{ps} p_i + \sum_{j=1}^{os} o_j + \max\{p_1, \ldots, p_{ps}\}(c-1)$$

最大化吞吐量，等价于最小化Time-Per-Sample

$$\text{s.t.} \quad \sum_{u \in V} P_{ui} S_u^\top A_u + \sum_{\langle u,v \rangle \in E} P_{ui} P_{vi} (S_u^\top R_{uv} S_v) = p_i, \quad \forall i \in \{1, \ldots, ps\}$$

节点内部重排布代价约束

$$\sum_{\langle u,v \rangle \in E} P_{uj} P_{v(j+1)} (S_u^\top R'_{uv} S_v) = o_j, \quad \forall j \in \{1, \ldots, os\}$$

流水线阶段间通信代价约束

$$\sum_{u \in V} P_{ui} S_u^\top M_u \leqslant m, \quad \forall i \in \{1, \ldots, ps\}$$

显存代价约束

$$V_i = \{\forall u \in V : P_{ui} = 1\} \text{ is contiguous}, \quad \forall i \in \{1, \ldots, ps\}$$

流水线的保序约束：

$$\begin{aligned} Z_{vi} &\geqslant P_{vi}, && \forall v \in V, \forall i \in \{1, 2, \ldots, ps\} \\ Z_{vi} &\leqslant Z_{ui}, && \forall \langle u, v \rangle \in E, \forall i \in \{1, 2, \ldots, ps\} \\ Z_{vi} &\leqslant P_{vi} - P_{ui} + 1, && \forall \langle u, v \rangle \in E, \forall i \in \{1, 2, \ldots, ps\} \end{aligned}$$

$$\sum_{i=1}^{ps} P_{ui} = 1, \quad \forall u \in V$$

$$\sum_{u \in V} P_{ui} \geqslant 1, \quad \forall i \in \{1, \ldots, ps\}$$

放置策略约束

$$\sum_{k=1}^{|g_u|} S_{uk} = 1, \quad \forall u \in V$$

并行策略约束

$$P_{ui} \in \{0, 1\} \quad \forall u \in V, i \in \{1, \ldots, ps\}$$

$$S_{uk} \in \{0, 1\} \quad \forall u \in V, k \in \{1, \ldots, |g_u|\}$$

0-1约束

Hao Lin, Ke Wu, Jie Li, Jun Li, Wu-Jun Li. UniAP: Unifying Inter- and Intra-Layer Automatic Parallelism by Mixed Integer Quadratic Programming. CVPR 2025.

自动并行

- UniAP的统一优化过程 (UOP)
  - 枚举流水线并行的维度
  - 枚举微批量大小
  - 无流水线并行时仅求解层间并行
  - 取全局最优值

Algorithm 1 Unified Optimization Process

Input: Profiling results PR, strategy dictionary SD, mini-batch size B, computation graph $\mathcal{G}$, and the number of GPUs n.
Output: Optimal cost $cost_{min}$, pipeline degree deg_{min}, the number of micro-batches c_{min}, layer placement P_{min}, and intra-layer strategy S_{min}

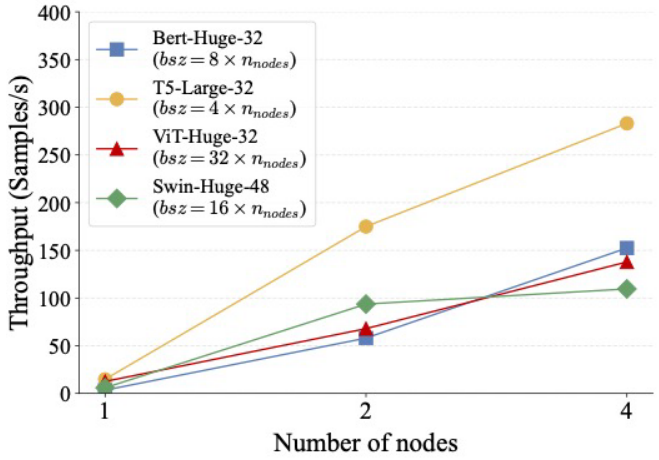
$deg_{min} = 1$;
$c_{min} = B$;
$A, R, \_, M$ = `CalculateCost`$(PR, SD[1], \mathcal{G}, B)$;
$cost_{min}, P_{min}, S_{min}$ = `QIP`(A, R, M);
for deg **in** $\{2, 4, \ldots, n\}$ **do**
  for $c = 2$ **to** B **and** $c \mid B$ **do**
    Micro-batch size $b = B/c$;
    A, R, R', M = `CalculateCost`$(PR, SD[deg], \mathcal{G}, b)$;
    $cost, P, S$ = `MIQP`(A, R, R', M, deg, c);
    if $cost < cost_{min}$ **then**
      $cost_{min}, deg_{min}, c_{min}, P_{min}, S_{min} = cost, deg, c, P, S$;
    end if
  end for
end for

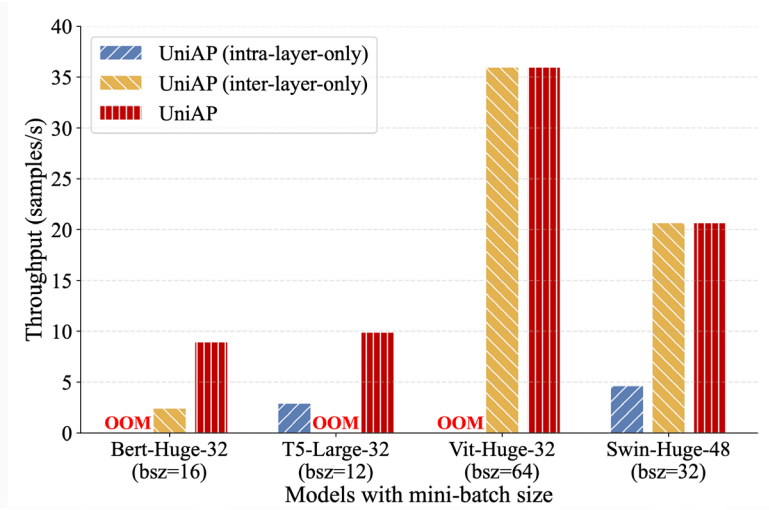
Hao Lin, Ke Wu, Jie Li, Jun Li, Wu-Jun Li. UniAP: Unifying Inter- and Intra-Layer Automatic Parallelism by Mixed Integer Quadratic Programming. CVPR 2025.

UniAP的实验结果

Env.	Model	Training throughput (samples/s)			Minimum speedup	Maximum speedup
		Galvatron	Alpa	UniAP		
EnvA	BERT-Huge	**33.46 ± 0.28**	31.56 ± 0.04	**33.46 ± 0.28**	1.00	1.06
	T5-Large	**23.29 ± 0.04**	MEM×[2]	**23.29 ± 0.04**	1.00	1.00
	ViT-Huge	**109.51 ± 0.07**	97.66 ± 1.42	**109.51 ± 0.07**	1.00	1.12
	Swin-Huge	CUDA×[3]	N/A[4]	**67.96 ± 0.12**	N/A[4]	N/A[4]
EnvB	BERT-Huge	6.27 ± 0.17	8.95 ± 0.06	**10.77 ± 0.13**	1.20	1.71
	T5-Large[1]	**8.06 ± 0.06**	MEM×[2]	7.98 ± 0.05	0.99	0.99
	ViT-Huge	32.20 ± 0.17	38.74 ± 0.20	**45.58 ± 0.54**	1.18	1.41
	Swin-Huge	13.90 ± 0.17	N/A[4]	**19.08 ± 0.10**	1.37	1.37
EnvC	Llama-7B	1.22 ± 0.01	N/A[4]	**4.63 ± 0.007**	3.80	3.80

Env.	Model	Strategy optimization time (min.)			Minimum speedup	Maximum speedup
		Galvatron	Alpa	UniAP		
EnvA	BERT-Huge	6.44 ± 0.588	> 40	**0.37 ± 0.002**	17.29	> 107.41
	T5-Large	12.41 ± 0.122	MEM×[2]	**0.89 ± 0.007**	13.98	13.98
	ViT-Huge	6.29 ± 0.464	> 40	**0.57 ± 0.009**	10.95	> 69.60
	Swin-Huge	11.88 ± 0.666	N/A[4]	**2.16 ± 0.004**	5.49	5.49
EnvB	BERT-Huge	2.04 ± 0.010	> 40	**1.51 ± 0.005**	1.34	> 26.32
	T5-Large[1]	2.64 ± 0.110	MEM×[2]	**0.91 ± 0.005**	2.90	2.90
	ViT-Huge	2.37 ± 0.180	> 40	**1.11 ± 0.011**	2.14	> 36.01
	Swin-Huge	4.29 ± 0.320	N/A[4]	**2.29 ± 0.010**	1.87	1.87
EnvC	Llama-7B	6.84 ± 0.055	N/A[4]	**0.58 ± 0.006**	11.83	11.83





Hao Lin, Ke Wu, Jie Li, Jun Li, Wu-Jun Li. UniAP: Unifying Inter- and Intra-Layer Automatic Parallelism by Mixed Integer Quadratic Programming. CVPR 2025.

UniAP：国产卡 8节点 × 4 DCU/节点

Model	Training throughput (samples/s)			Strategy optimization time (min.)		
	Megatron	DeepSpeed	UniAP	Megatron	DeepSpeed	UniAP
Llama-7B	**2.01 ± 0.005**	SOL×[1)]	**2.01 ± 0.005**	> 8.0 hours	SOL×[1)]	**3.07 ± 0.121**
Llama-13B	**0.82 ± 0.001**	SOL×[1)]	**0.82 ± 0.001**	> 2.5 hours	SOL×[1)]	**1.95 ± 0.076**

Model	Batch size	Training throughput (samples/s)				#infeasible[5)]	#candidate[6)]
		Top-1[1)]	Top-2[2)]	Slowest[3)]	Median[4)]		
Llama-7B	8	2.01	1.92	0.22	0.82	41	64
Llama-13B	4	0.82	0.58	0.27	0.42	42	48

Hao Lin, Ke Wu, Jie Li, Jun Li, Wu-Jun Li. UniAP: Unifying Inter- and Intra-Layer Automatic Parallelism by Mixed Integer Quadratic Programming. CVPR 2025.

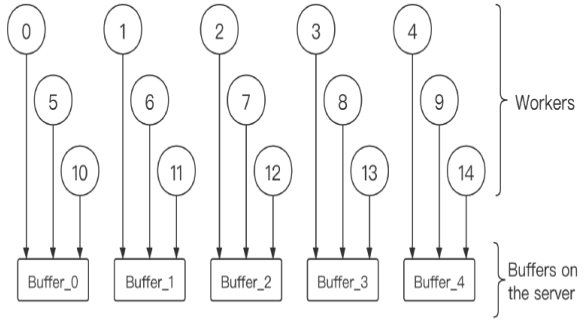
拜占庭分布式学习

- Meta 官方最近披露了在 16384 块 H100 80GB 集群上进行 Llama3 405B 模型训练的故障率
  - 短短 54 天，发生 419 次中断，平均每三小时崩溃一次
  - 10万卡集群，平均每20分钟崩溃一次
- 国产卡
- 解决方案
  - 异步分布式训练算法
  - 鲁棒分布式训练算法

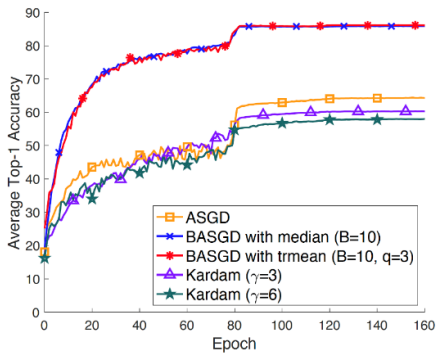
拜占庭分布式学习

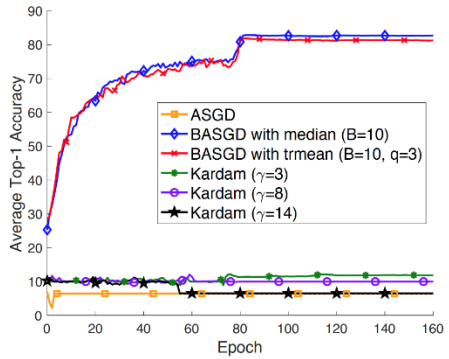
南京大学 NANJING UNIVERSITY

- BASGD: 基于异步通信架构的拜占庭分布式学习算法
  - 拜占庭问题：故障；故障、恶意攻击（联邦学习场景）
  - 基于同步通信架构的方法存在中心服务器通信瓶颈问题；
  - 设计了基于异步通信架构的拜占庭分布式学习算法BASGD/BASGDm，解决了中心服务器通信瓶颈问题，且Server不需存储样本，能处理non-iid数据。首个Server不需存储样本且能抗恶意攻击的异步算法。相关论文发表于ICML 2021/JMLR 2023。



$$\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{w}^t)\|^2]}{T} \leq O\left(\frac{L[F(\mathbf{w}^0)-F^*]}{T^{\frac{1}{2}}}\right) + O\left(\frac{rd\tilde{D}}{T^{\frac{1}{2}}(q-r+1)^{\frac{1}{2}}}\right) + O\left(\frac{rDd\sigma}{(q-r+1)^{\frac{1}{2}}}\right) + O\left(\frac{rDd\kappa}{(q-r+1)^{\frac{1}{2}}}\right) + O\left(\frac{r^{\frac{3}{2}}LD\tilde{D}d^{\frac{3}{2}}\tau_{max}}{(q-r+1)^{\frac{3}{4}}}\right)$$





Yi-Rui Yang, Wu-Jun Li. BASGD: Buffered Asynchronous SGD for Byzantine Learning. ICML 2021/JMLR 2023.

拜占庭分布式学习

南京大学 NANJING UNIVERSITY

ByzSGDnm: 基于规范化动量的拜占庭分布式学习算法

- 首次从理论和实验上证明了随着故障节点或者恶意节点数的增加，最优的批量大小应当增大，但直接增大批量会造成泛化精度下降；
- 设计了基于规范化动量的算法ByzSGDnm，可以容许更大的批量，得到比已有方法更高的精度。额外的好处：减小通信开销，提升学习速度。相关论文发表于ICLR 2024。

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(\mathbf{w}_t)\|^2 \le \mathcal{U}(B)$$

$$B^* = \left(\frac{3}{16L^2(F_0)^2 m}\right)^{\frac{1}{3}}\left(\frac{c\delta(1+c\delta m)}{m(1-\delta)}\right)^{\frac{1}{3}}\sigma^{\frac{4}{3}}\mathcal{C}^{\frac{1}{3}}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \cdot \frac{\mathbf{Agg}(\mathbf{u}_t^{(1)},\ldots,\mathbf{u}_t^{(m)})}{\|\mathbf{Agg}(\mathbf{u}_t^{(1)},\ldots,\mathbf{u}_t^{(m)})\|}$$

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(\mathbf{w}_t)\| \le \frac{6\left[\sqrt{2cm\delta(1-\delta)}+1\right]^{\frac{1}{2}}(5LF_0\sigma^2)^{\frac{1}{4}}}{\mathcal{C}^{\frac{1}{4}}} + \frac{18\left[\sqrt{2cm\delta(1-\delta)}+1\right]^{\frac{3}{4}}c}{\mathcal{C}^{\frac{1}{2}}}$$

Batch size	ByzSGDm with KR		
	$\delta=0$	$\delta=\frac{1}{8}$	$\delta=\frac{3}{8}$
32×8	**91.08%**	55.84%	38.55%
64×8	89.98%	63.22%	54.15%
128×8	89.71%	75.06%	55.98%
256×8	89.15%	84.47%	59.28%
512×8	86.15%	**85.68%**	83.42%
1024×8	84.97%	83.48%	**83.45%**

Batch size	Best	32×8	64×8	128×8	256×8	512×8	1024×8
ByzSGDm + KR	**83.45%**	38.55%	54.15%	55.98%	59.28%	83.42%	**83.45%**
ByzSGDnm + KR	**85.93%**	43.47%	70.88%	80.20%	82.83%	85.12%	**85.93%**
ByzSGDm + GM	**87.62%**	63.11%	70.88%	82.08%	**87.62%**	86.95%	84.75%
ByzSGDnm + GM	**89.13%**	69.45%	83.23%	86.63%	88.66%	**89.13%**	88.16%
ByzSGDm + CM	**83.25%**	33.11%	55.66%	66.38%	82.47%	**83.25%**	80.94%
ByzSGDnm + CM	**86.03%**	61.28%	71.46%	80.24%	83.55%	**86.03%**	85.74%
ByzSGDm + CC	**87.46%**	72.83%	79.45%	84.94%	87.25%	**87.46%**	83.70%
ByzSGDnm + CC	**88.53%**	78.50%	83.91%	86.56%	88.32%	**88.53%**	87.89%

Batch size	32×8	64×8	128×8	256×8	512×8
ByzSGDm	2007.39s	985.52s (×2.04 faster)	522.27s (×3.84 faster)	366.98s (×5.47 faster)	314.80s (×6.38 faster)
ByzSGDnm	1985.78s	978.50s (×2.03 faster)	515.46s (×3.85 faster)	376.70s (×5.27 faster)	327.62s (×6.06 faster)

Yi-Rui Yang, Chang-Wei Shi, Wu-Jun Li. On the Effect of Batch Size in Byzantine-Robust Distributed Learning. ICLR 2024.

拜占庭分布式学习

南京大学 NANJING UNIVERSITY

- 拜占庭鲁棒性与无拜占庭准确度之间的tradeoff理论
  - 首次在理论上研究了基于鲁棒聚合的分布式机器学习方法中，拜占庭鲁棒性与无拜占庭准确度之间的矛盾关系。
  - 理论结果表明，对更多拜占庭工作节点鲁棒的方法，在无拜占庭工作节点情形下的准确度更低。因此，在实际应用中设计拜占庭鲁棒的分布式机器学习算法时，需要注意拜占庭鲁棒性和无拜占庭准确度之间的权衡。实验数据验证并进一步支持了本文中的理论结果。论文发表于ICML 2025。

Theorem 3.1 (Lower Bound). *If an (f,κ)-robust aggregator is ϵ-accurate, we have $\epsilon \geq \frac{f}{n-f}$.*

Table 2. Values of ϵ for different robust aggregators

Aggregator	GM	$\mathrm{TM}_{f/n}$	CM	Lower bound
ϵ	1	$\frac{f}{n-f}$	$\frac{\lfloor \frac{n-1}{2} \rfloor}{n-\lfloor \frac{n-1}{2} \rfloor}$	$\frac{f}{n-f}$

f	Multi-Krum (Blanchard et al., 2017)		
	$\alpha = 0.1$	$\alpha = 1.0$	$\alpha = 10.0$
0 (=mean)	89.42%	89.36%	89.55%
1	88.05% (-1.37%)	87.50% (-1.86%)	88.36% (-1.19 %)
3	83.50% (-5.92%)	84.49% (-4.87%)	87.02% (-2.53%)
5	69.86% (-19.56%)	80.40% (-8.96%)	84.64% (-4.91%)
7	40.31% (-49.11%)	68.54% (-20.82%)	73.69% (-15.86%)

Yi-Rui Yang, Chang-Wei Shi, Wu-Jun Li. On the Tension between Byzantine Robustness and No-Attack Accuracy in Distributed Learning. ICML 2025.

UniAP：高质效分布式机器学习平台/框架

南京大学 NANJING UNIVERSITY

UniAP + SNGM

模型	训练环境	批量大小	梯度下降算法	准确率
ViT-Base	2×4 NVIDIA Titan XP	1024	MSGD	71.63%
			SNGM	**76.23%**
		8192	MSGD	63.78%
			SNGM	**74.20%**
ViT-Huge	1×8 NVIDIA Tesla V100	1024	MSGD	70.22%
			SNGM	**76.85%**
		8192	MSGD	56.59%
			SNGM	**70.03%**

模型	训练环境	批量大小	使用算法	平均遍历用时（秒）
ViT-Base	2×4 NVIDIA Titan XP	1024	SNGM	165.89
			SNGM-3D	**54.49**
		8192	SNGM	17.64
			SNGM-3D	**11.78**

UniAP + ByzSGDnm

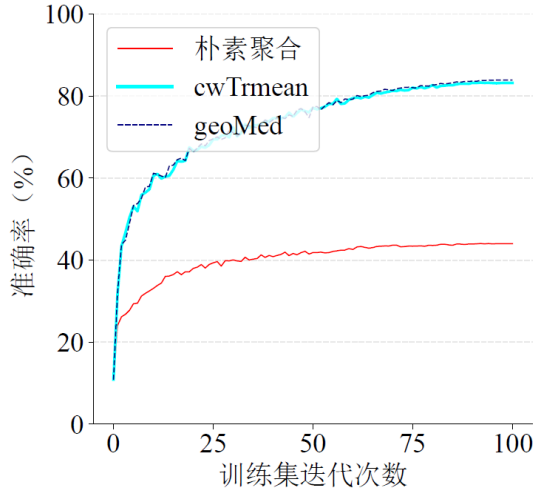


表 5-4　训练过程中参数服务器的内存负载对比

GPU 数量	原始 ByzSGDnm 算法	ATP
2	7.50GB	**6.25GB**
4	9.08GB	**6.68GB**
8	12.17GB	**7.73GB**

单卡多核：CUDA
多机多卡：UniAP

- **英伟达GPU卡流行的重要原因是CUDA解决了多核编程的易用性和高效性问题；**
- **UniAP可以看成是大模型多机多卡分布式训练的“CUDA++”**

国产适配

- 因为软件生态问题，国产卡训练大模型还有很多问题，导致国产卡闲置和浪费
- 首要问题是解决易用性问题
- 已将UniAP移植到华为昇腾和海光DCU上，并运行成功
- UniAP自动规避无效策略，解决了国产卡的易用性问题；自动搜索最优策略，提升了国产卡的效率

国产适配

华为昇腾910A

多机 (2节点，16卡)，global_bsz=8				
方法	UniAP	Megatron	手动	手动
策略	4TP 2DP 2PP	8TP 2PP	4TP 4PP	2TP 2DP 4PP
	micro-bsz=1	micro-bsz=1	micro-bsz=1	micro-bsz=1
	#mic=4	#mic=8	#mic=8	#mic=4
throughput (sample/s)	2.2154 (+65%)	1.3381	2.1817	OOM

国产适配

华为昇腾910A

多机 (4节点，32卡)， global_bsz=16				
方法	UniAP	Megatron	手动	手动
策略	4TP 4DP 2PP	8TP 4PP	8TP 2DP 2PP	2TP 4DP 4PP
	micro-bsz=1	micro-bsz=1	micro-bsz=1	micro-bsz=1
	#mic=4	#mic=16	#mic=8	#mic=4
throughput (sample/s)	4.1170 (+68%)	2.4548	2.5513	OOM

集群越大，UniAP提升越大

海光DCU BW1000

- 硬件参数
  - 显存64G，算力BF16 480 TFLOPS，机内通信带宽100GB/s左右，机间通信带宽10GB/s左右，对标有NVLink的A100
- 场景
  - 训练阶段：预训练，后训练（SFT, 强化学习）
  - 模型：Qwen3-8B
- 系统实现
  - 预训练使用Megatron-LM，推理使用vLLM，强化学习使用veRL（训练后端为Megatron-LM，推理后端为vLLM）。均基于海光提供的适配版。

国产适配

- 在DCU集群上测试了Megatron框架多节点分布式训练
  - 集群大小：2个8卡节点，模型：Qwen3-8B，数据集：llama2_dataset
    分布式框架：MPI
  - global-batch-size: 256, micro-batch-size: 1, 开启sequence-parallel

并行策略	平均每卡吞吐量 (tokens/s)
8tp-1pp-2dp-1cp	70.83
8tp-2pp-1dp-1cp	65.73
2tp-2pp-4dp-1cp	158.05
4tp-2pp-1dp-2cp	51.60
2tp-2pp-2dp-2cp	86.85

国产适配

- 在DCU集群上测试了veRL框架多节点分布式训练
  - 集群大小：2个8卡节点，模型：Qwen3-8B，数据集：gsm8k，分布式框架：Ray
  - rollout_n: 5, max_prompt_length: 512, max_response_length: 1024

并行策略	平均每卡吞吐量 (tokens/s)
8tp-1pp-2dp-1cp	13.8368
8tp-2pp-1dp-1cp	12.9950
8tp-1pp-1dp-2cp	5.2994
4tp-2pp-1dp-2cp	11.2994
2tp-2pp-2dp-2cp	OOM

智能医疗应用


南京鼓楼医院大数据平台
专病库列表
专病库 病例数 字段数
17 脂肪肝专病库 10670 313
18 间质肺专病库 9970 277
19 慢阻肺专病库 9737 165
20 软组织肉瘤... 8895 303
21 前列腺癌专... 8497 235
22 骨科微创专... 8024 62
23 导管血流感... 7021 22
24 颈椎手术专... 5234 184
25 黑色素瘤专... 3519 379
26 骨肿瘤专病库 2338 75
专病库数据类型
人口学 16%
自定义 18%
诊断 4%
影像文件 4%
医嘱 4%
AI自动化 9%
病历 11%
随访 14%
检验 11%
病理 4%
检查 7%
总数据量 113.10亿条
总表数 11719张
总字段数 178459个
总存储量 2.0PB
影像数据存储量 1.9PB
非影像数据存储量 14.9TB
数据源
No.5 电子病历 333770477条
No.6 PACS 94409163条
No.7 病案 17435345条
No.1 护理 1716698952条
No.2 HIS 1709779748条
大数据中心算力
8P
生产数据占比 68.53%
专病数据占比 31.2%
支撑项目信息
国家自然科学基金-TREM2/SCD4在蛛网膜下腔出血后调控小胶质...
国家自然科学基金-面向临床实施的早期胃癌智能诊断方法研究
国家自然科学基金-TIM-4调控小胶质细胞向吞噬型转化促进蛛网膜...
国家重点研发计划-肺血栓栓塞症多维度数据平台建立及全流程管...
国家科技重大专项-重症感染患者代谢率和宏量营养素精准监测及...
非影像数据量/存储量增长情况
存储量（TB）
数据量（亿条）
10.00
80.00
2025-04
总存储量（TB）：10
数据量（亿条）：87
2023-07 2024-01 2024-07 2025-04 2025-07 2026-01
大模型应用场景
科室 场景
29 甲状腺外科 甲状腺结节辅助诊断与手术规划...
30 纪律监督室 采购与招投标风险筛查场景
31 精准医学中心 多组学数据分析解读场景
32 急诊医学科 急性胸痛辅助诊断与分析场景
33 康复科 康复效果定量评价与方案调整场景
34 口腔科 口腔影像辅助诊断场景
35 老年医学科 老年综合征风险评估场景
36 临床营养科 个性化营养方案生成场景
37 门诊部 智能预问诊与分诊场景
38 生物样本库 样本质量辅助评估场景
大模型访问量
11月: 229,314
1月 2月 3月 4月 5月 6月 7月 8月 9月 10月 11月 12月

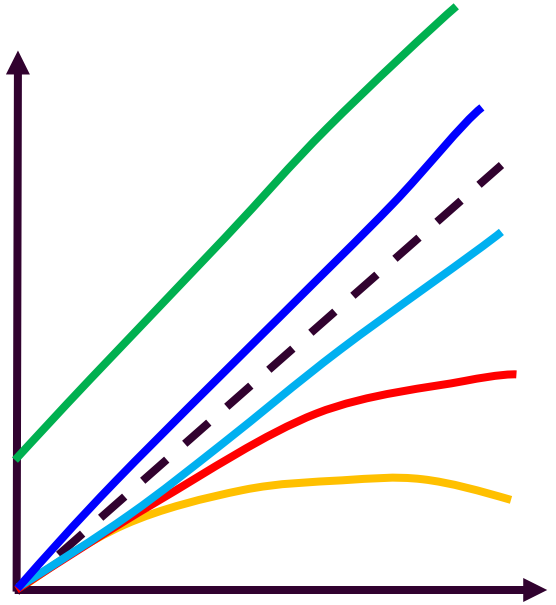
目录

总结

- 大模型的训练面临极大的算力（成本）挑战

- 分布式机器学习算法极大地影响算力利用率、梯度有效率、系统容错性和易用性

- 高质效分布式机器学习算法能提升算力利用率、梯度有效率、系统容错性和易用性，从而提升大模型的训练效率并降低能耗和成本，算力有限和算力充足场景都需要



实际算力增大倍数 = 理论算力增大倍数 * W

W为算法加权因子，可以大于1，也可以小于1

展望

- 多类型芯片（跨芯片）混合异构系统中的自动并行
- 跨数据中心/跨地域训练
- 多算法无缝整合
- 系统与平台实现
  - PCL算力网项目

谢谢

liwujun@nju.edu.cn