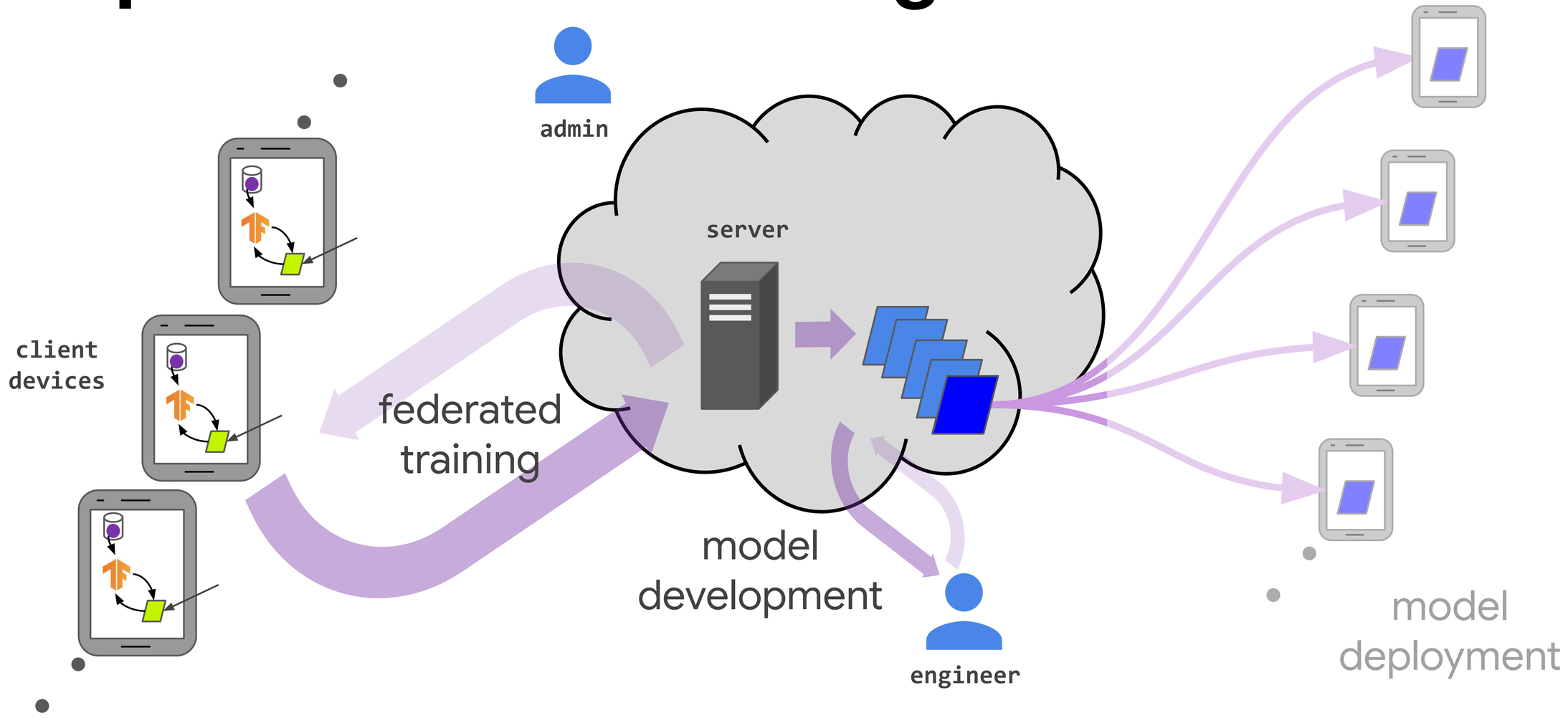
CSCI 699: Privacy Preserving Machine Learning - Week 11

Privacy in Federated Learning

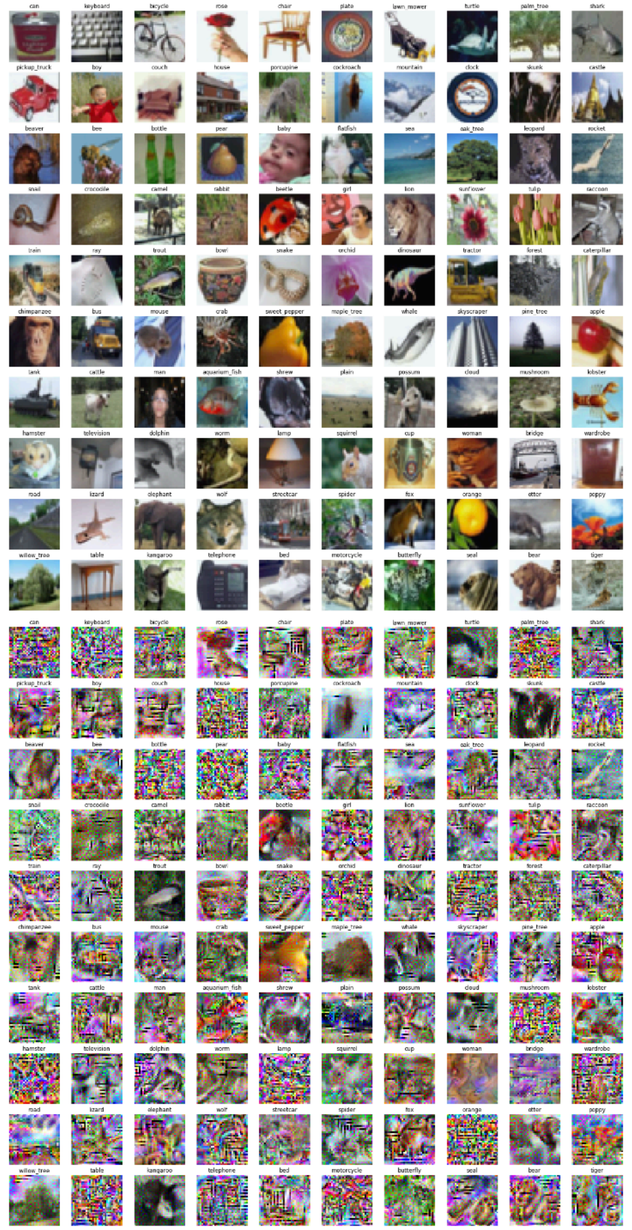
Sai Praneeth Karimireddy, Nov 15 2024.

Recap: Federated Learning



Recap: Federated Learning

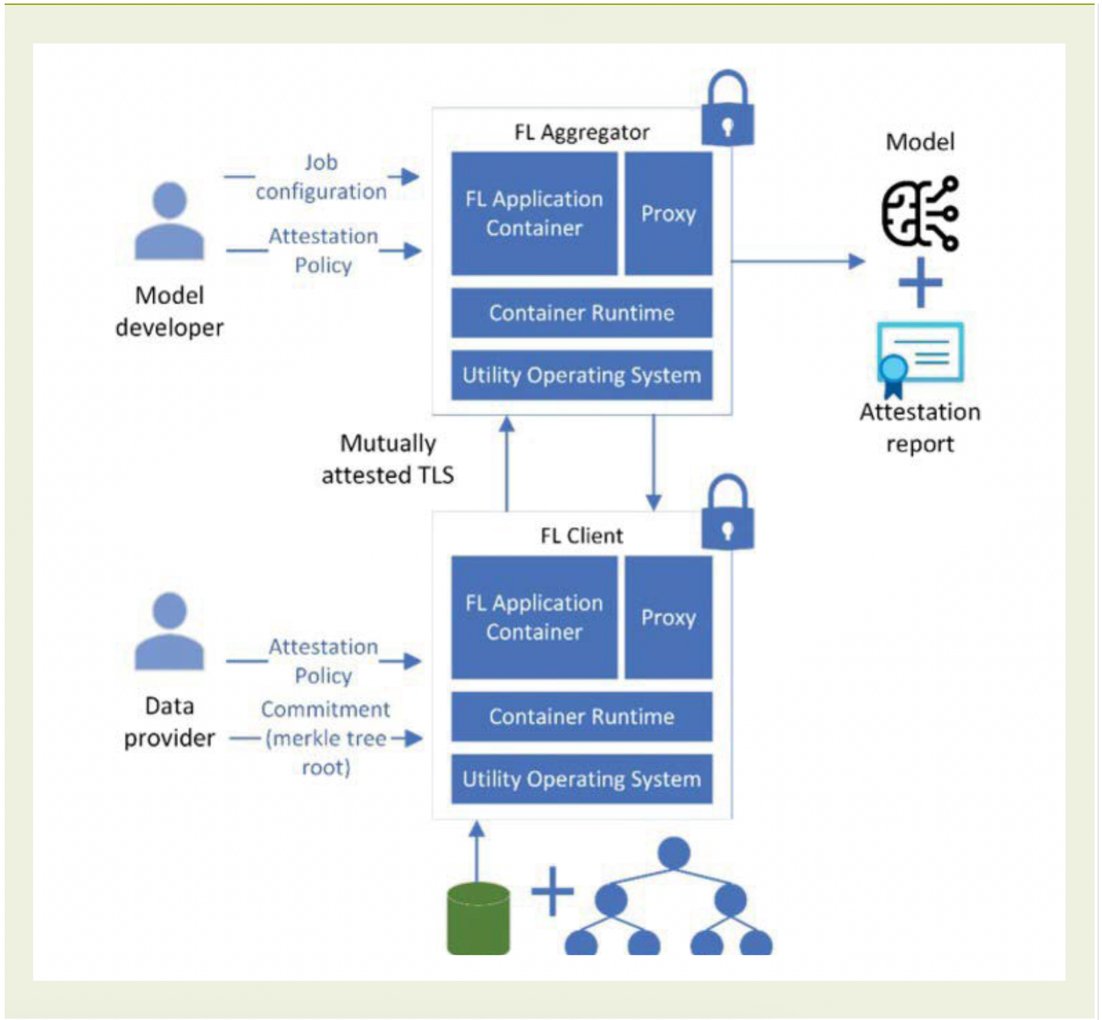
- We can invert gradients to recover the training data.
- Recovering is easier with smaller batch size vs. larger batch sizes.
- If the server is adversarial, can also insert “traps” into the model sent out to the clients. But, this is less likely - may be easy to detect.



How to Make Federated Learning Private?

Confidential Computing

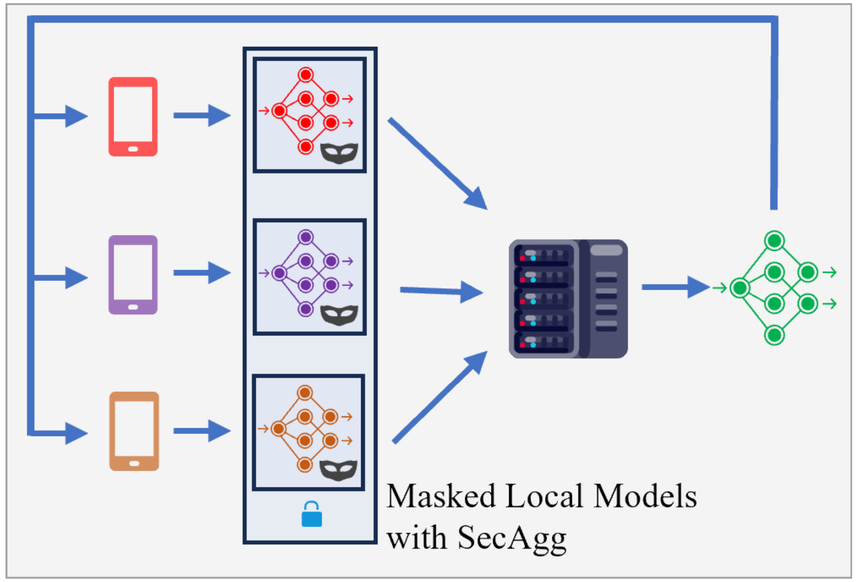
- **Idea 1:** Use confidential computing at server.
- Run the FL aggregation and add DP noise within TEE.
- Pro: Gives local-DP like privacy/ security guarantees at the cost of central-DP. **Server can be adversarial.**
- Con: Need a trusted TEE & verify attestation every round.



How to Make Federated Learning Private?

Secure Aggregation

- **Idea 2:** Just ensure server never sees “individual client” updates
    - larger batch
    - hides who sent what
- Use Multi-Party-Copmutation (MPC) to perform secure aggregation.
- Server only sees the aggregate, never individual updates.
- Typically assume server is **honest but curious.**



FL with SecAgg

Scalable Training communication efficiency

Collaborating without Sharing: Setup

[McMahan et al. 2016]

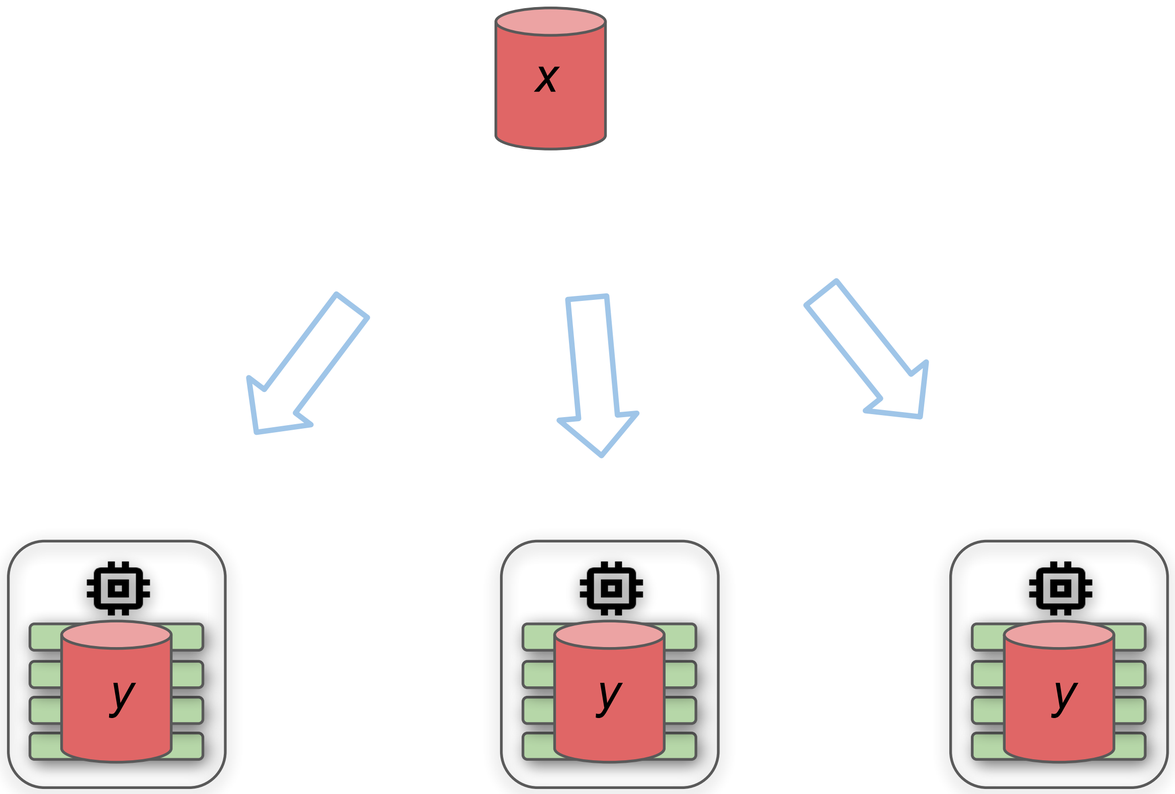
Clients
(MSF mobiles or
cancer registries)

Collaborating without Sharing: Setup

[McMahan et al. 2016]

Iteratively in each round,

- Copy latest model

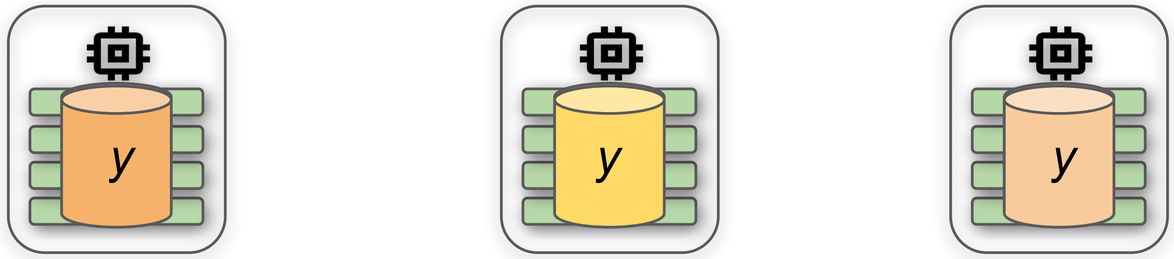


Collaborating without Sharing: Setup

[McMahan et al. 2016]

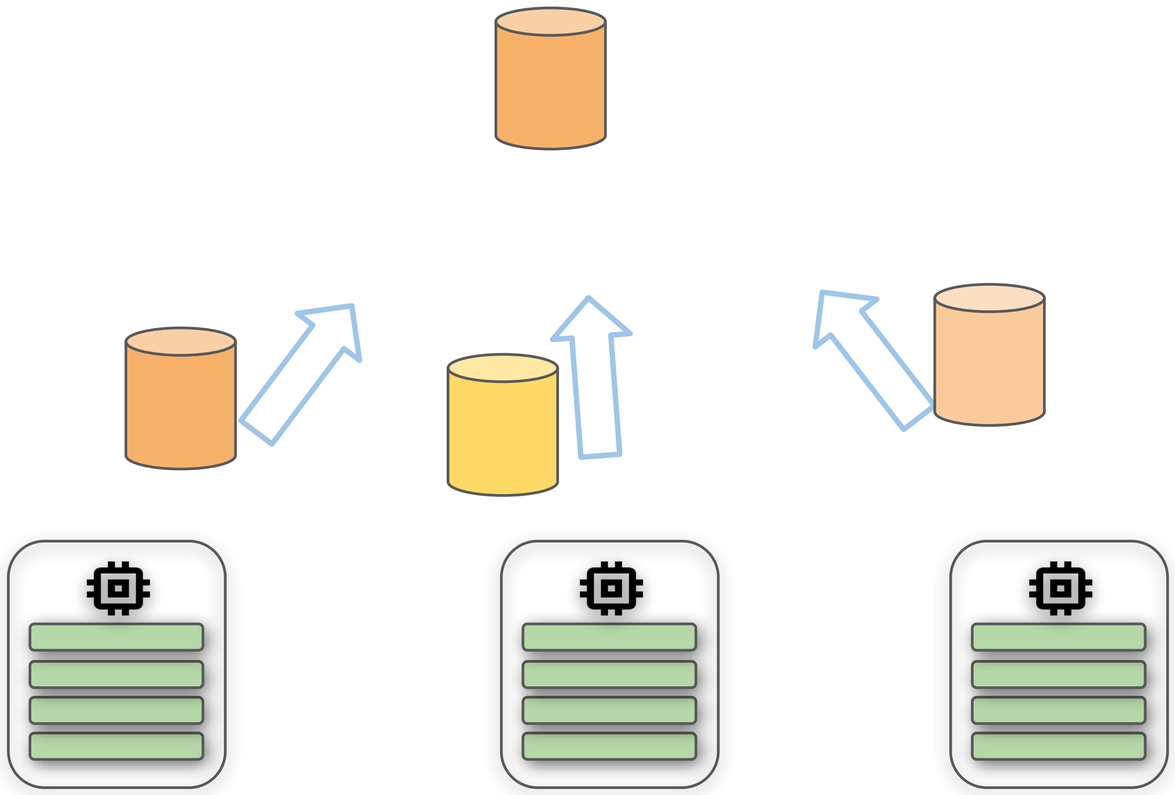
In each round,

- Update model using local data and compute



Collaborating without Sharing: Setup

[McMahan et al. 2016]

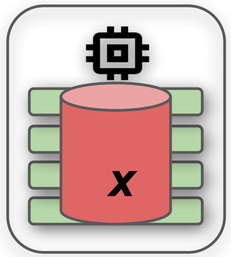


In each round,

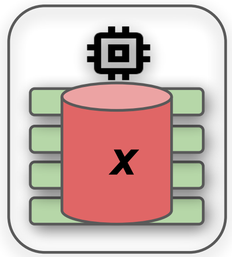
Aggregate models to get new model.

Repeat.

Distributed training



$L_1(x)$

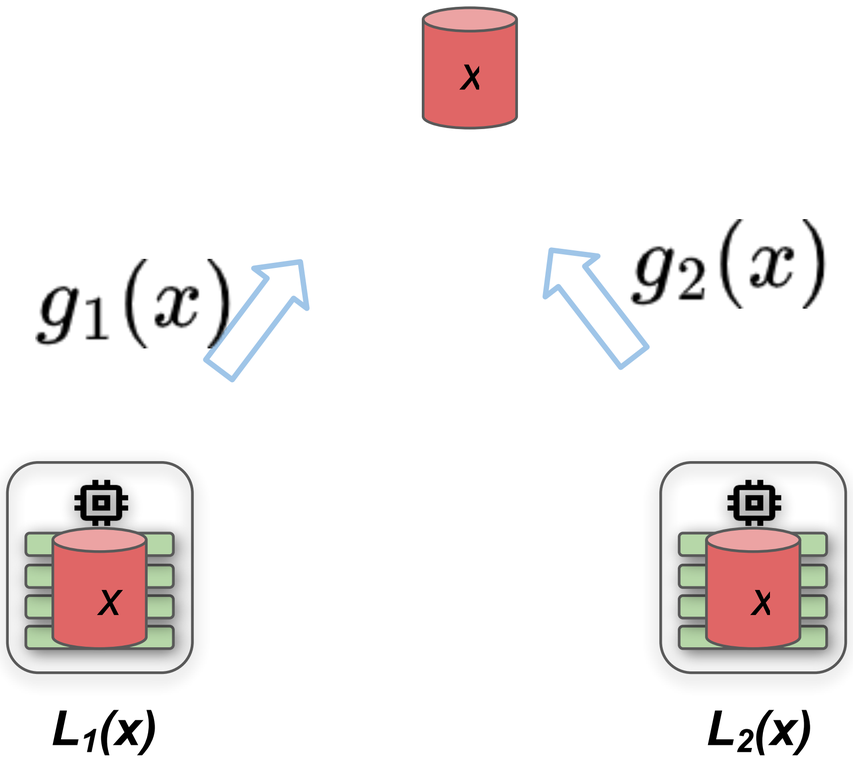


$L_2(x)$

$\min L(x) = 1/m \sum_i L_i(x)$

distributed stochastic optimization

Distributed training: SGD?



Compute local stochastic gradient and average them.

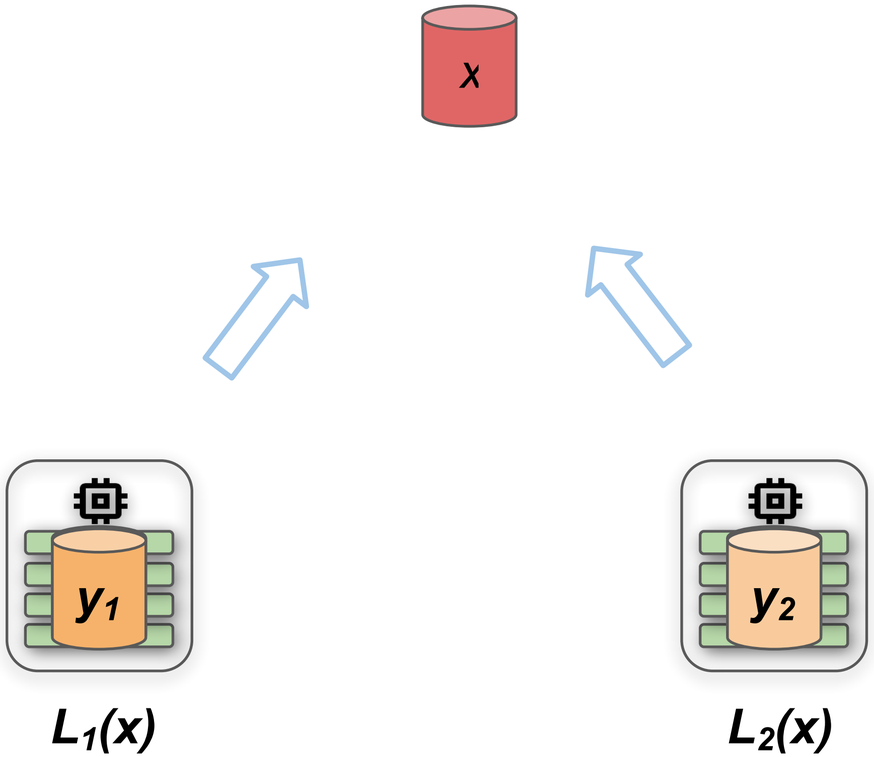
$$x \leftarrow x - \gamma \frac{1}{m} \sum_i g_i(x)$$

Excruciatingly slow! :(

Distributed training: SGD?

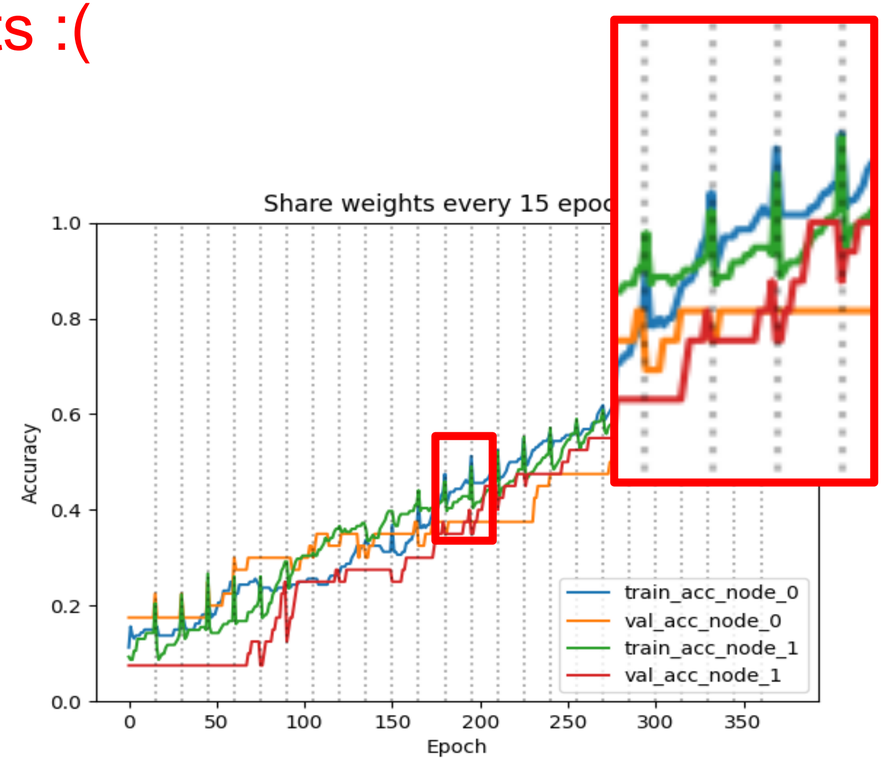
	Single GPU	vs.	Collaboration USA & Switzerland
computation per round	1ms		1ms
communication per round	**1ms**		**1s**
training time	**1 hr**		**5 days!**

Communication-efficiency: infrequent synchronization



Run **1k** SGD steps each round, before communicating.

Overfits :(



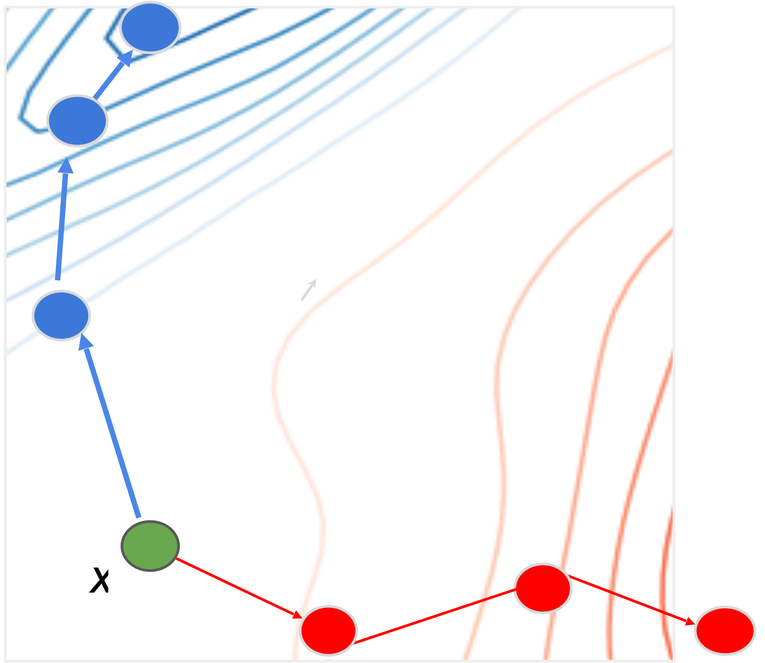
FedAvg [McMahan et al. 2016]

Communication-efficiency: infrequent synchronization

$$y_i = y_i - \eta g_i(y_i)$$

biased

Clients drift far from each other due to data heterogeneity.



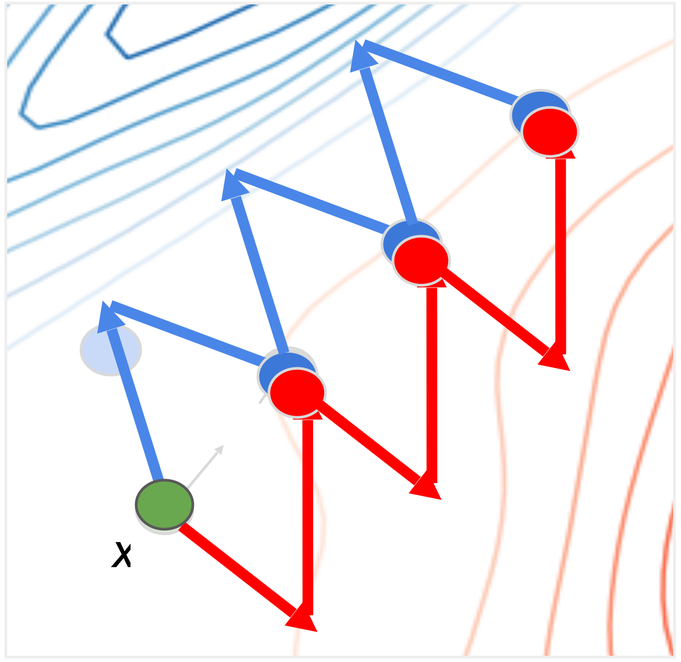
[SCAFFOLD - Karimireddy et al. ICML 2020]

Communication-efficiency: infrequent synchronization

$$y_i = y_i - \eta(g_i(y_i) + \boxed{c - c_i})$$

correction

Use history to compute correction.



[SCAFFOLD - Karimireddy et al. ICML 2020]

Infrequent synchronization

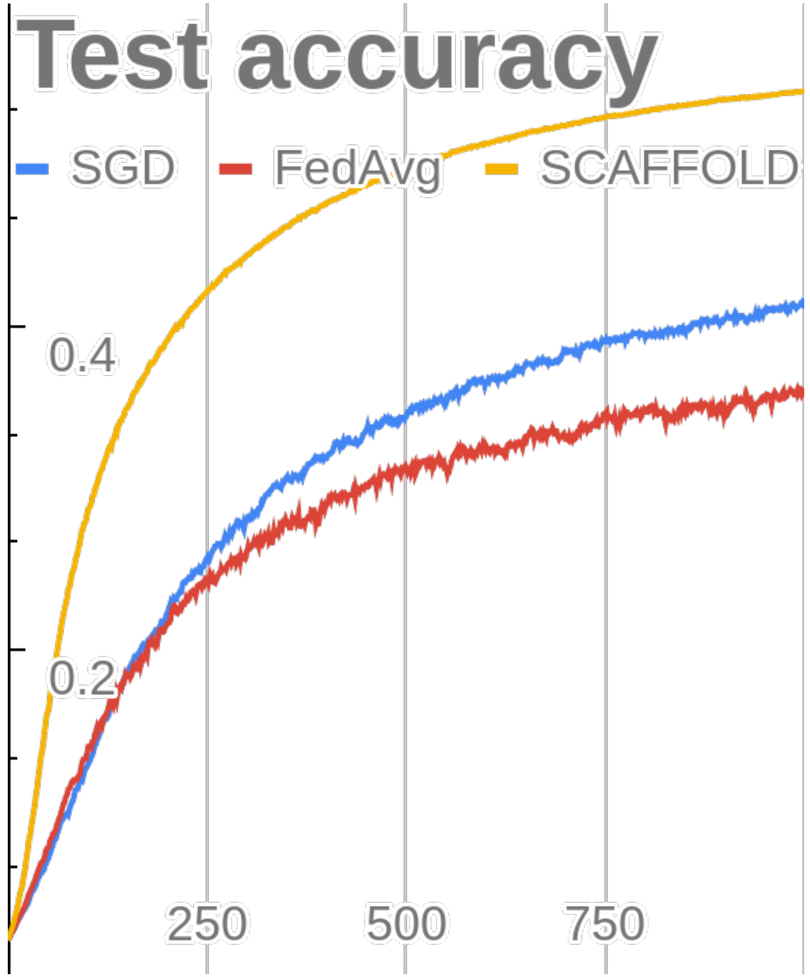
$$y_i = y_i - \eta(g_i(y_i) + \boxed{c - c_i})$$

correction

Theorem [Karimireddy et al. 2020]

SCAFFOLD converges with (nearly) optimal communication complexity.

[SCAFFOLD - Karimireddy et al. ICML 2020]



Communication rounds -->

Connections with variance reduction, operator splitting, …

Many many extensions…

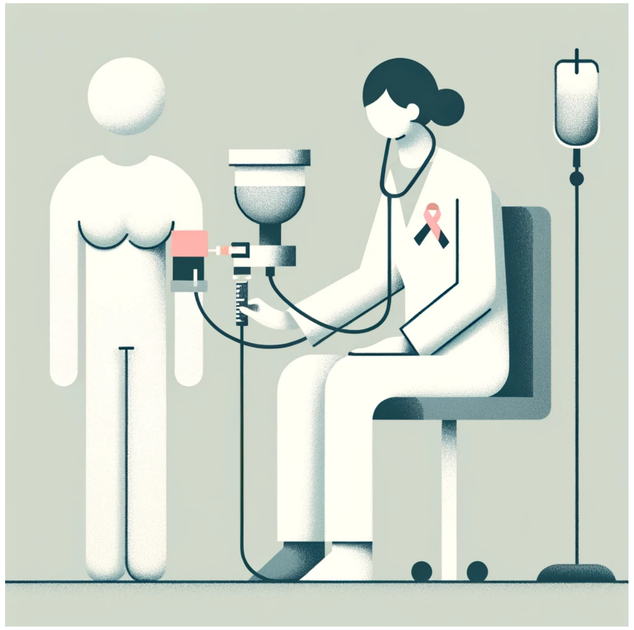
Mime: Extensions using Adam, etc.
Train-Convexify-Train:
Approximate DL using NTK

[Karimireddy et al. NeurIPS 2021]
[Yu, Wei, Karimireddy et al. NeurIPS 2022]

Google

google / fedjax Public

Neoadjuvant chemotherapy (NACT) for triple-negative breast cancer (TNBC)

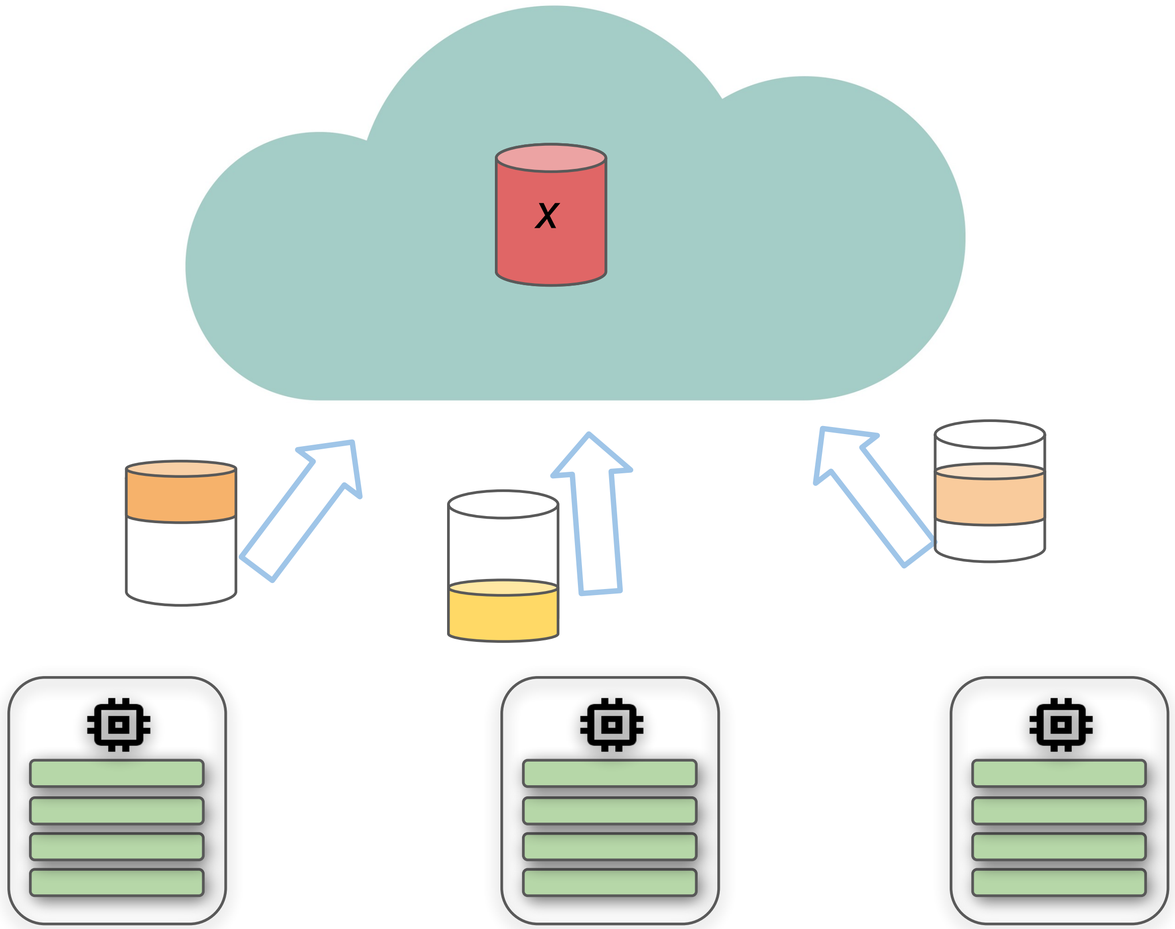


OWKIN

"Scaffold is the most promising collaborative strategies [sic]*"*

- Terrail et al. [Nature Medicine 2023]

Idea 2: Compressed Communication

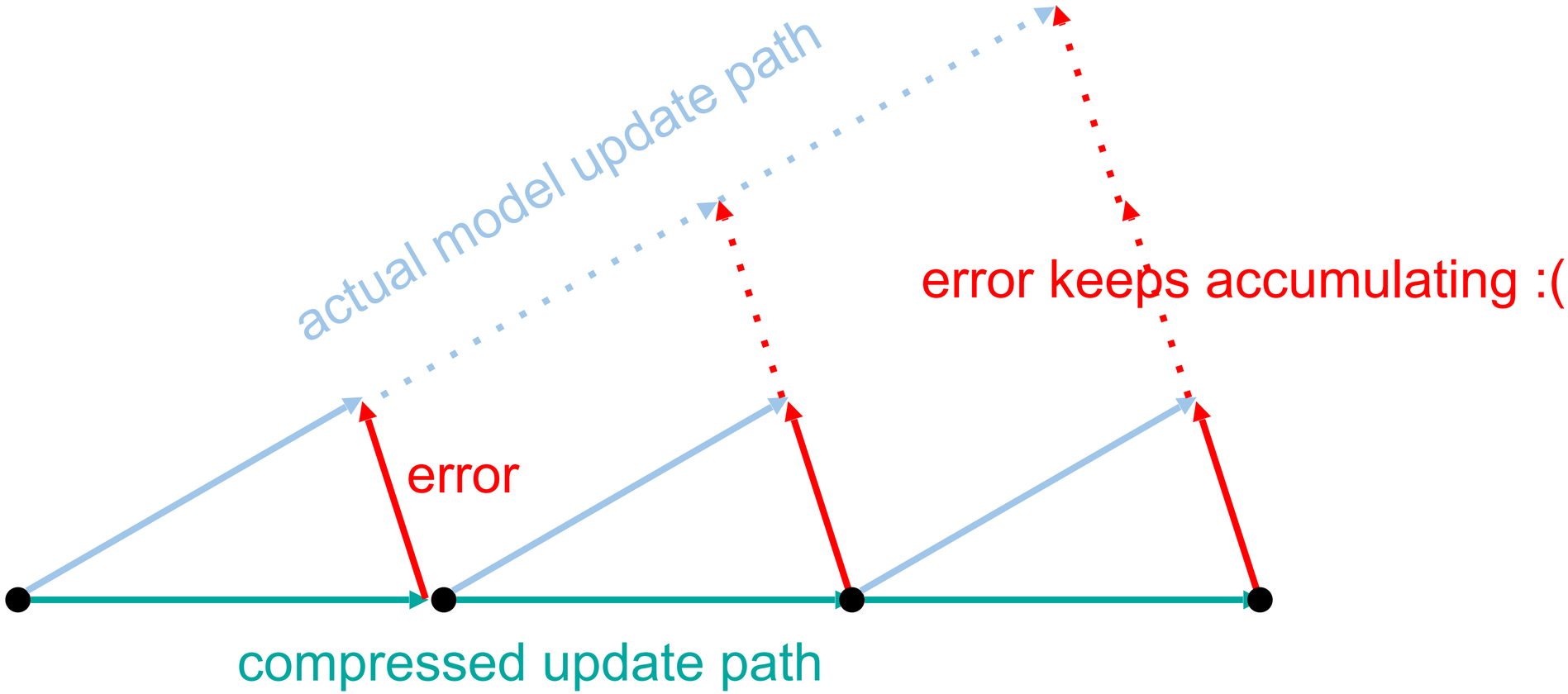


GPT-2 is 5GB!

Send only the *most important* parts.
E.g. sign, top-k, low-rank, …

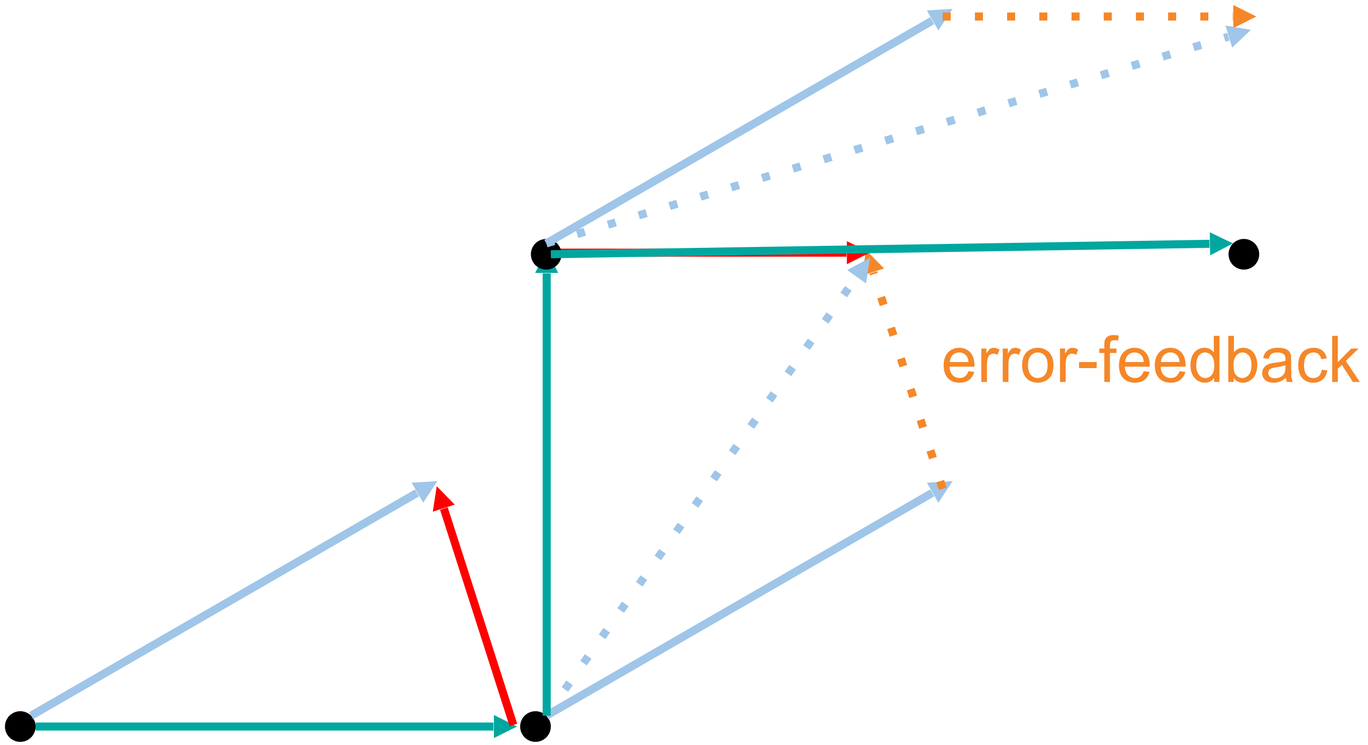
Use *error-feedback* to correct for bias.

Idea 2: Compressed Communication



[Karimireddy et al. ICML 2019],
[Stich & Karimireddy JMLR 2020]

Compression w/ **Error feedback**



Theorem (Informal):
The asymptotic rate of convergence of EF-SGD is the same as SGD.

add the error from previous round back into next round before compressing.

[Karimireddy et al. ICML 2019],
[Stich & Karimireddy JMLR 2020]

Compression w/ **Error feedback**

Theorem (Informal):

The asymptotic rate of convergence of EF-SGD is the same as SGD.

More concretely, for any δ-approximate compressor:

$$\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \leq \mathcal{O}\left(\frac{\sigma^2}{\sqrt{T}}\right) + \mathcal{O}\left(\frac{1}{\delta\, T}\right)$$

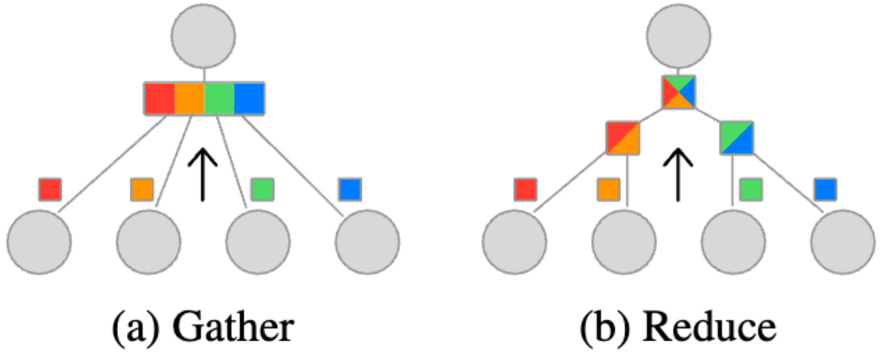
SGD rate

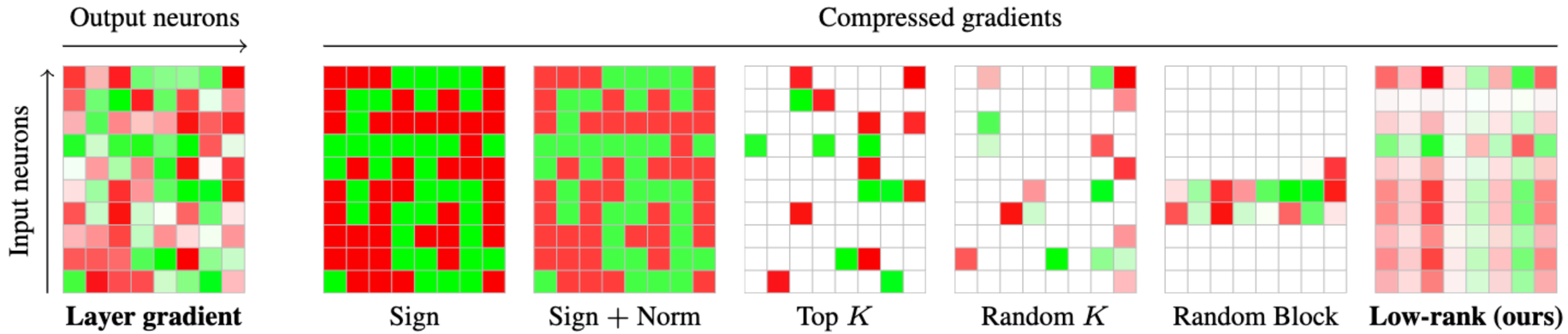
Lower order dependence

[Karimireddy et al. ICML 2019], [Stich & Karimireddy JMLR 2020]

Which compressor to use?

- Needs to satisfy some key properties:
  - Compatibility with all-reduce
  - Cheap to compute on GPU
  - Have a good compressor (large δ)





PowerSGD Results

		Test accuracy	Sent/epoch	All-reduce	Time/batch
No compression		94.3%	1023 MB	✓	312 ms
Medium	**Rank 7**	94.6%	24 MB	✓	285 ms
	Random Block	93.3%	24 MB	✓	243 ms
	Random K	94.0%	24 MB	✓	540 ms
	Sign+Norm	93.9%	32 MB	✗	429 ms
	Top K	94.4%	32 MB	✗	444 ms
High	**Rank 2**	94.4%	8 MB	✓	239 ms
	Random Block	87.8%	8 MB	✓	240 ms
	Random K	92.6%	8 MB	✓	534 ms
	Top K	93.6%	8 MB	✗	411 ms

- If not compatible with all-reduce, time/batch (throughput) is bad
- Random compression is compatible with all-reduce but affects convergence.

[Vogels, Karimireddy, et al. NeurIPS 2019]

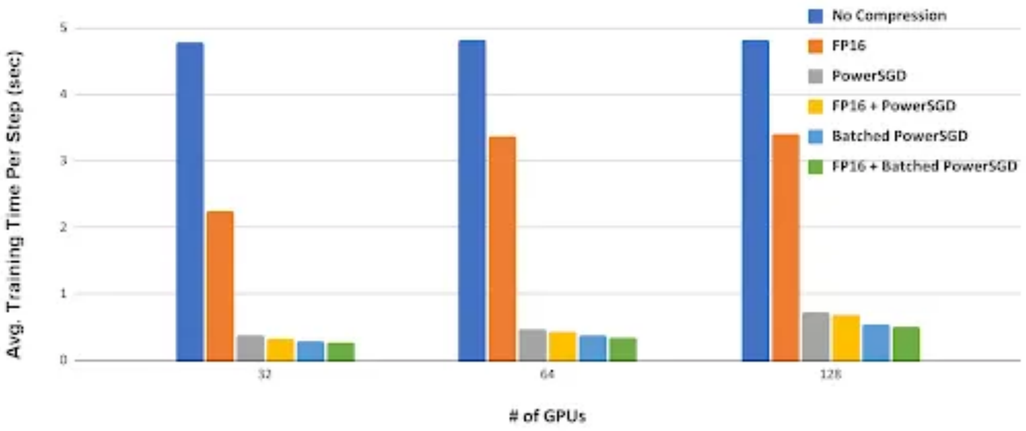
Meta

Accelerating PyTorch DDP by 10X With PowerSGD

PyTorch Follow

Nov 4, 2021 · 8 min read

Authors: *Yi Wang (Facebook AI), Alex Iankoulski (Amazon AWS), Pritam Damania (Facebook AI), Sundar Ranganathan (Amazon AWS)*

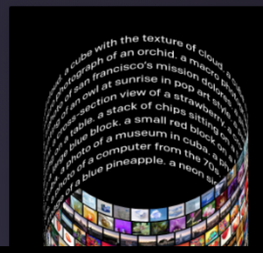


OpenAI

DALL·E: Creating Images from Text

We've trained a neural network called DALL·E that creates images from text captions for a wide range of concepts expressible in natural language.

January 5, 2021
27 minute read



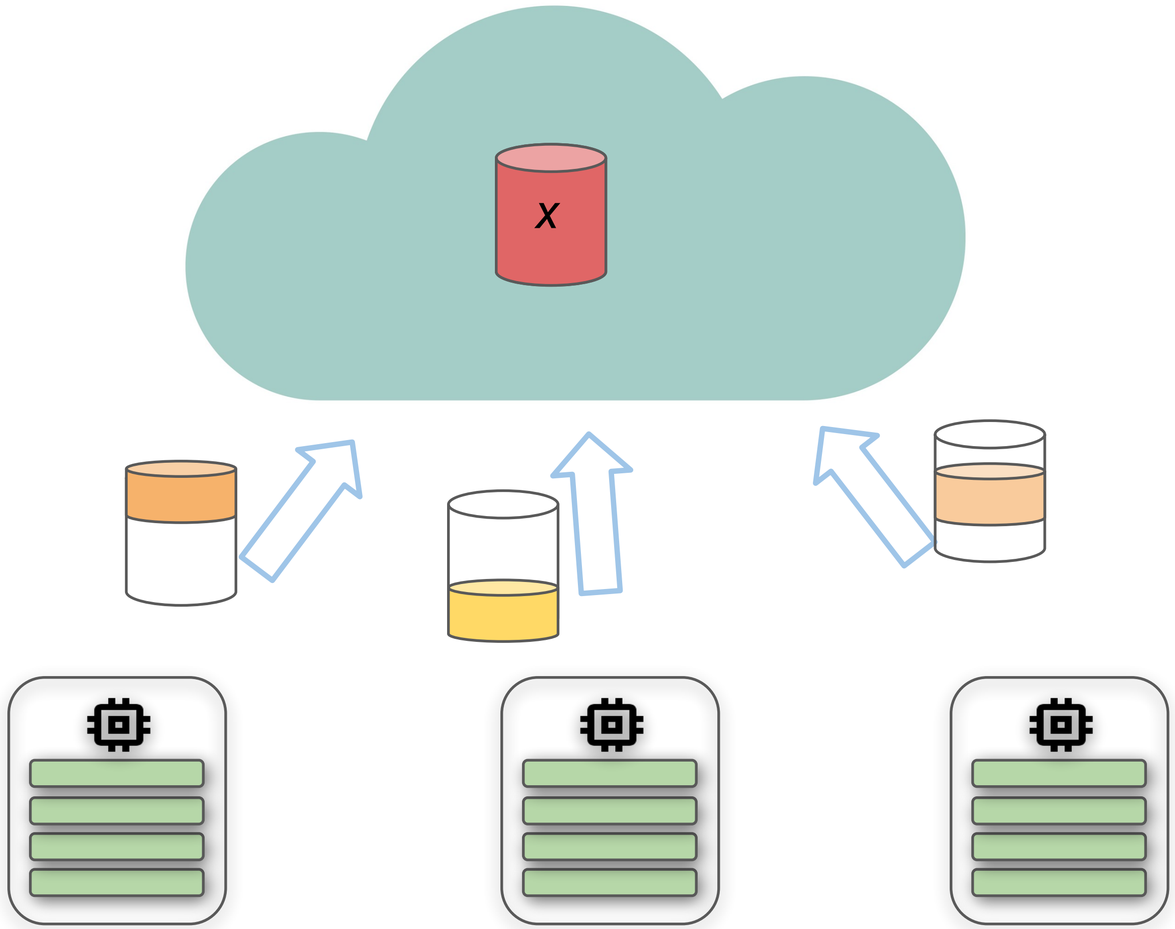
hayabusa
@shreyasgite

PowerSGD:
Every time you do all-reduce, the gradients are communicated from & to all GPUs. It is a considerable bottleneck. DALLE used nifty gradient compression (86%) from the PowerSGD paper to make this work. 15/n
arxiv.org/abs/1905.13727

Compression Scheme	Compression Rate
FP16	2
PowerSGD	984
FP16 + PowerSGD	1,969
Batched PowerSGD	1,946
FP16 + Batched PowerSGD	3,892

Compression for Privacy

Idea 2: Compressed Communication



We are communicating much less.

Intuitively, this must also mean more privacy.

Indeed true! We will see formal analysis soon.

Compression for privacy

	Communication (bits)	ℓ_2 error
Local DP [32, 42]	$\Theta\left(\lceil\varepsilon\rceil\right)$	$\Theta\left(\frac{d}{n\min(\varepsilon^2,\varepsilon)}\right)$
Distributed DP (with SecAgg) [33]	$\tilde{O}\left(n^2\min\left(\varepsilon,\varepsilon^2\right)\right)$	$\Theta\left(\frac{d}{n^2\min(\varepsilon^2,\varepsilon)}\right)$
Central DP (Theorem 4.4)	$\tilde{O}\left(n\min\left(\varepsilon,\varepsilon^2\right)\right)$	$O\left(\frac{d\log d}{n^2\min(\varepsilon^2,\varepsilon)}\right)$
Shuffle DP (Theorem 6.4)	$\tilde{O}\left(n\log(d)\min\left(\varepsilon,\varepsilon^2\right)\right)$	$O\left(\frac{d}{n^2\min(\varepsilon^2,\varepsilon)}\right)$

Table 1: Comparison of the communication costs of ℓ_2 mean estimation under local, distributed, central, and shuffle DP (with δ terms hidden). Compared to local DP, we see that error under central DP decays much faster (e.g., $1/n^2$ as opposed to $1/n$); compared to distributed DP with secure aggregation, our schemes achieve similar accuracy but saves the communication cost by a factor of n.

[Chen et al. 2023]