Federated Accelerated Stochastic Gradient Descent

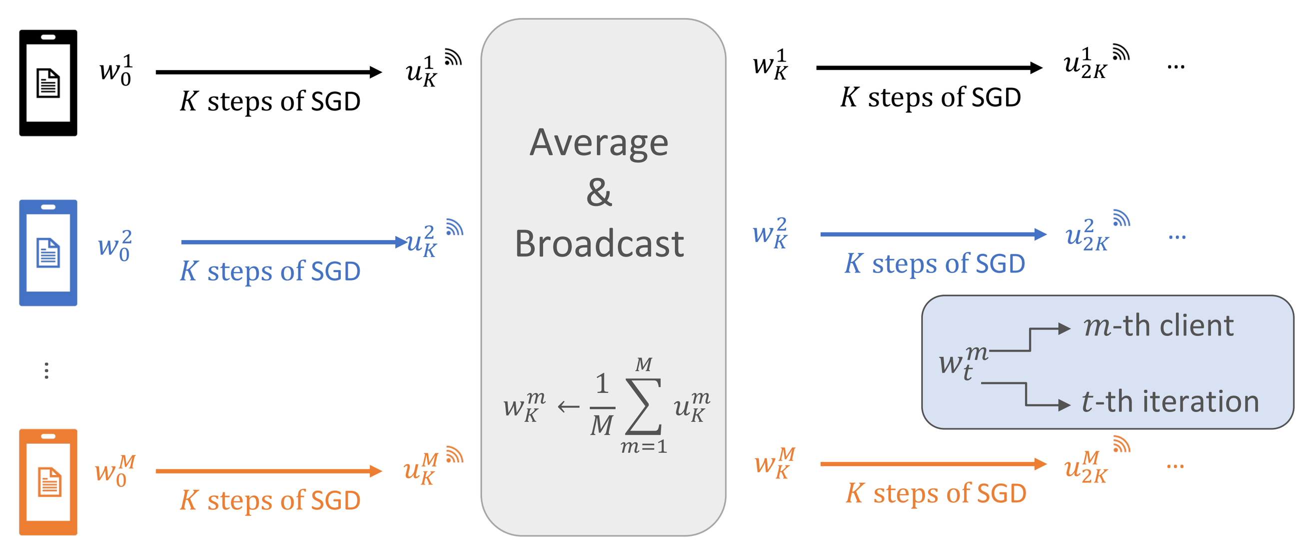
Honglin Yuan (Stanford University), Tengyu Ma (Stanford University)

Abstract

We propose Federated Accelerated Stochastic Gradient Descent (FedAc), the first principled acceleration of Federated Averaging (FedAvg, also known as Local SGD), which provably improves convergence speed and communication efficiency on various types of convex functions.
For example, for strongly convex and smooth functions, when using M workers, the previous state-of-the-art FedAvg analysis can achieve a linear speedup in M if given M rounds of synchronization, whereas FedAc only requires $M^{1/3}$ rounds. Moreover, we prove stronger guarantees for FedAc when the objectives are third-order smooth.

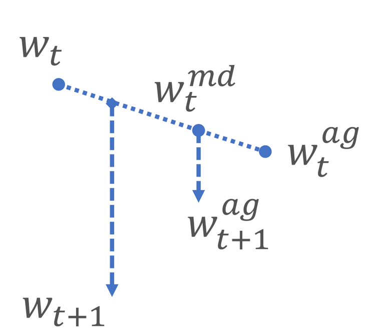
Algorithm: From FedAvg to FedAc

- FedAvg is the standard algorithm for Federated Optimization. Each client runs a local SGD and is periodically synchronized by averaging.

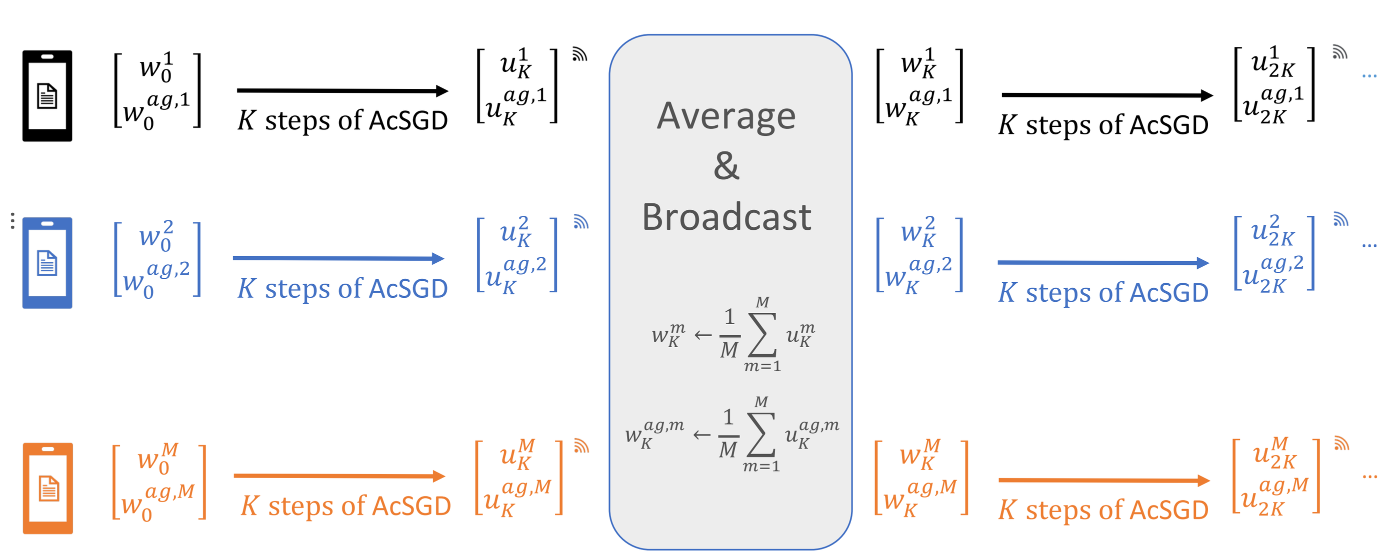


- We propose Federated Accelerated Stochastic Gradient Descent (FedAc).
- In FedAc, each client follows an accelerated SGD [Ghadimi et al., 2012] by maintaining a 3-tuple:
  - w_t as main state,
  - w_t^{ag} as aggregated state
  - w_t^{md} as an auxiliary "middle state"

- $w_t^{md} \leftarrow \beta^{-1} w_t + (1-\beta^{-1}) w_t^{ag}$
- $w_{t+1}^{ag} \leftarrow w_t^{md} - \eta \cdot \nabla f(w_t^{md}; \xi_t)$
- $w_{t+1} \leftarrow (1-\alpha^{-1}) w_t + \alpha^{-1} w_t^{md} - \gamma \cdot \nabla f(w_t^{md}; \xi_t)$



- During communication, both w_t and w_t^{ag} are averaged and broadcasted.



Theory: Setup

We consider the stochastic optimization $\min_{w \in \mathbb{R}^d} F(w) := \mathbb{E}_{\xi \sim \mathcal{D}}[f(w;\xi)]$, where

1. F is smooth and strongly convex
2. $\nabla f(w;\xi)$ has bounded variance
3. Each client can access $\nabla f(w;\xi)$ for independent sample ξ drawn from (the same) distribution $\mathcal{D}$

- Similar models have been studied by existing works on FedAvg e.g., [Stich et al., 2019], [Khaled et al., 2020], [Woodworth et al., 2020]
- Commonly known as i.i.d. settings, where FedAvg is also known as Local-SGD

Theory: Results

M: # of clients
R : # of sync. rounds
T : parallel runtime

FedAvg [Khaled et al., 2020], [Woodworth et al., 2020]:

$$\mathbb{E}[F(\cdot)] - F^* \le \tilde{\mathcal{O}}\left(\frac{1}{MT} + \frac{1}{TR}\right)$$

- Achieve linear speedup in M if the bound is dominated by $\tilde{\mathcal{O}}\left(\frac{1}{MT}\right)$
- Requires $R \sim M$ rounds to achieve linear speedup

FedAc **(Theorem 3.1)**

$$\mathbb{E}[F(\cdot)] - F^* \le \tilde{\mathcal{O}}\left(\frac{1}{MT} + \frac{1}{TR^3}\right)$$

- Requires only $R \sim M^{1/3}$ rounds to achieve linear speedup in M.
- **Acceleration saves communication!**

We establish stronger guarantee for both algorithms if $\nabla^{(3)}F$ is bounded

FedAvg with bounded $\nabla^{(3)}F$ **(Theorem 3.4)**

$$\mathbb{E}[F(\cdot)] - F^* \le \tilde{\mathcal{O}}\left(\frac{1}{MT} + \frac{1}{T^2R^2}\right)$$

FedAc with bounded $\nabla^{(3)}F$ **(Theorem 3.3)**

$$\mathbb{E}[F(\cdot)] - F^* \le \tilde{\mathcal{O}}\left(\frac{1}{MT} + \frac{1}{T^2R^6}\right)$$

We also study the convergence rates for general smooth convex objectives F. The results are summarized in this table.

Algorithms	Sync. Rounds (R) required for linear speedup		Reference
	Strongly Convex	General Convex	
FedAvg	$T^{1/2}M^{1/2}$	-	[Sti19]
	$T^{1/3}M^{1/3}$	-	[HKMC19]
	M	$T^{1/2}M^{3/2}$	[SK19][KMR20]
FedAc	$M^{1/3}$	$\min\{T^{1/4}M^{3/4}, T^{1/3}M^{2/3}\}$	This work
Stronger Guarantees when $\nabla^{(3)}F$ is bounded			
FedAvg	$\max\{T^{-1/2}M^{1/2}, 1\}$	$T^{1/2}M^{3/2}$	This work
FedAc	$\max\{T^{-1/6}M^{1/6}, 1\}$	$\max\{T^{1/4}M^{1/4}, T^{1/6}M^{1/2}\}$	This work

Theory: Proof Sketch

Most analysis framework of Federated Algorithms (at least implicitly) requires the stability of algorithms being parallelized

- For example, SGD is stable → FedAvg can work 🙂
- Unfortunately, standard Accelerated SGD is not stable enough
- In fact, we show that even deterministic standard Accelerated GD may not be initial-value stable (Theorem 4.2) 😞 may be of individual interest

Our solution: acceleration-stability trade-off 🤔

SGD → FedAvg
slow but stable

Standard AcSGD
fastest but unstable

Generalized AcSGD → Our FedAc
faster and stable 😁

Experiments: FedAc vs FedAvg & mini-batch

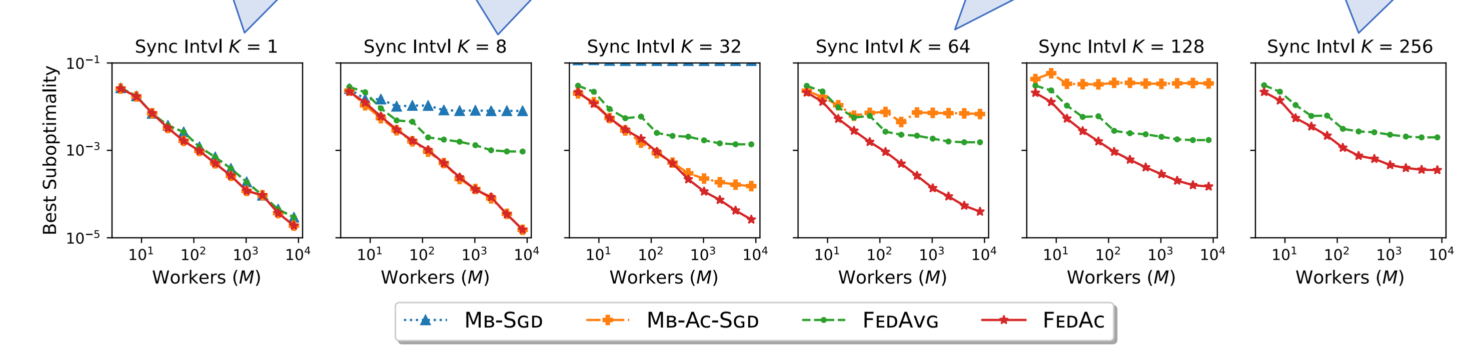
We compare FedAc with three baselines: FedAvg, the minibatch SGD with the same rounds of communication (Mb-Sgd), and minibatch accelerated SGD (Mb-Ac-Sgd).

All 4 algorithms attain linear speedup when K = 1

FedAvg and Minibatch-SGD loses linear speedup for K as low as 8

Minibatch-Accelerated-SGD drastically worsens for $K \geq 64$

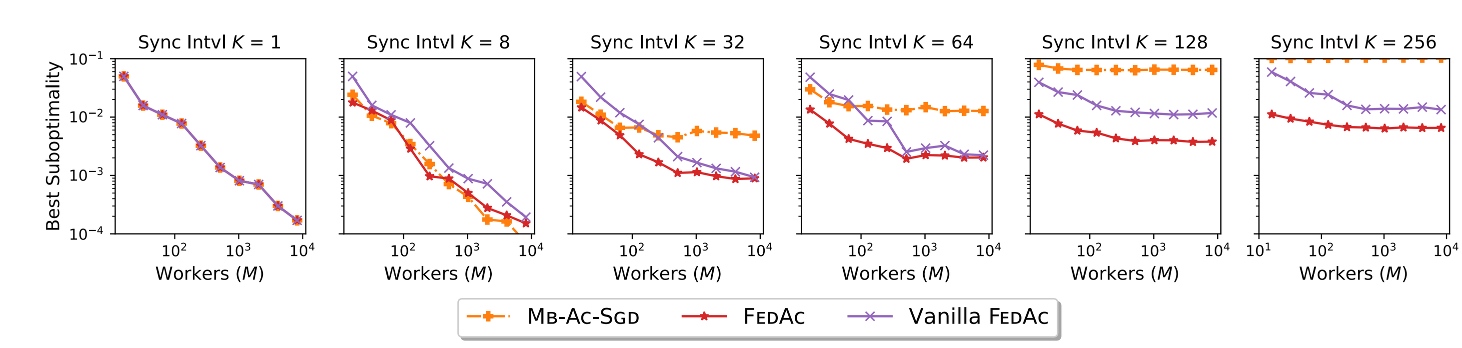
FedAc outperforms the baselines for all tested M and K.



Observed linear speedup with respect to clients M under various synchronization intervals K.

Experiments: Principled FedAc vs Vanilla FedAc

- We also compared our principled FedAc with the vanilla version of FedAc without the acceleration-stability trade-off.
- The result suggests direct Acceleration indeed suffers from instability, which complements our study on the instability of accelerated SGD.



References

Stich "Local SGD converges fast and communicates little" In: ICLR 2019

Stich et al. "The Error-Feedback Framework: Better Rates for SGD with Delayed Gradients and Compressed Communication" In: arXiv:1909.05350

Farzin Haddadpour et al. "Local SGD with periodic averaging: Tighter analysis and adaptive synchronization" In: NeurIPS 2019

Ahmed Khaled et al. "Tighter Theory for Local SGD on Identical and Heterogeneous Data" In: AISTATS 2020

Blake Woodworth et al. "Is Local SGD Better than Minibatch SGD?" In: ICML 2020