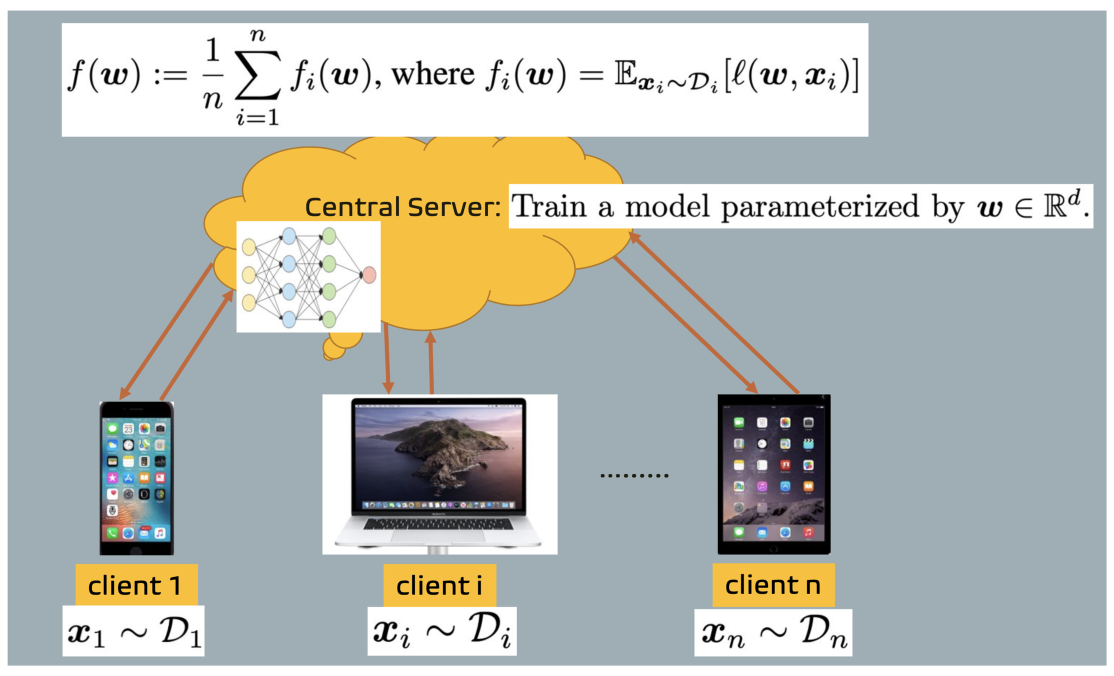
Faster Non-Convex Federated Learning via Global and Local Momentum

Rudrajit Das[1], Anish Acharya *[1], Abolfazl Hashemi *[2], Sujay Sanghavi[1], Inderjit S. Dhillon[1], and Ufuk Topcu[1]

[1]University of Texas at Austin , [2]Purdue University

Problem Setting and Motivation

Federated Learning (FL):



Federated Averaging, a.k.a. `FedAvg` [McM+17]: Clients perform **multiple** steps of local (S)GD updates with their own data, before communicating with the server. The server then updates the global model by simply **averaging** the received updates.

How good is `FedAvg`? On *smooth non-convex* functions ($g : \Theta \to \mathbb{R}$ is L-smooth if $\forall\, \boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$, $\|\nabla g(\boldsymbol{\theta}) - \nabla g(\boldsymbol{\theta}')\| \leq L\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$), `FedAvg` converges to an ϵ-stationary point ($\mathbb{E}[\|\nabla f(\boldsymbol{w})\|^2] \leq \epsilon$) in $\mathcal{O}(\epsilon^{-2})$ gradient updates [Kar+20]. Corresponding *lower bound* in the centralized setting is $\Omega(\epsilon^{-1.5})$ [Arj+19].

What hampers convergence in FL? High variance due to:
(i) plain averaging used in the **global** server aggregation step of `FedAvg`, exacerbated by heterogeneity of the clients; *this issue is specific to FL.*
(ii) noise of the **local** client-level stochastic gradients.

Key Idea: Reduce variance in the **global and local** updates to improve convergence.

`FedGLOMO`: Global and LOcal MOmentum-Based Variance Reduction

In order to achieve variance reduction, we propose to apply a *novel* **global momentum** term at the server in addition to **local momentum** at the clients. This is inspired by the variance-reducing momentum scheme of `STORM` [CO19].

Algorithm 1 `FedGLOMO`: Server Update

1. **Input:** Initial point $\boldsymbol{w}_0$, number of rounds of communication K, period E, learning rates $\{\eta_k\}_{k=0}^{K-1}$, global momentum parameters $\{\beta_k\}_{k=0}^{K-1}$, and number of participating clients r. Set $\boldsymbol{w}_{-1} = \boldsymbol{w}_0$.
2. **for** $k = 0, \ldots, K-1$ **do**
3. Server sends $\boldsymbol{w}_k$ to a set $\mathcal{S}_k$ of r clients chosen uniformly at random w/o replacement.
4. **for** client $i \in \mathcal{S}_k$ **do**
5. Run Algorithm 2 for client i with inputs $\boldsymbol{w}_0^{(i)} \leftarrow \boldsymbol{w}_k$, $\widehat{\boldsymbol{w}}_0^{(i)} \leftarrow \boldsymbol{w}_{k-1}$ and $\eta \leftarrow \eta_k$. Set $\boldsymbol{d} \to \boldsymbol{g}_k^{(i)}$ and $\widehat{\boldsymbol{d}} \to \widehat{\boldsymbol{g}}_{k-1}^{(i)}$.
6. **end for**
7. Set $\boldsymbol{u}_k = \frac{1}{r}\sum_{i\in\mathcal{S}_k} \boldsymbol{g}_k^{(i)} + \mathbb{1}(k>0)(1-\beta_k)\Big(\boldsymbol{u}_{k-1} - \frac{1}{r}\sum_{i\in\mathcal{S}_k}\widehat{\boldsymbol{g}}_{k-1}^{(i)}\Big)$. `// (Global Momentum)`
8. Update $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \boldsymbol{u}_k$.
9. **end for**

Algorithm 2 `FedGLOMO`: Client Update

1. **Input:** Initial points $\boldsymbol{w}_0^{(i)}$ and $\widehat{\boldsymbol{w}}_0^{(i)}$, period E, learning rate η.
2. **for** $\tau = 0, \ldots, E-1$ **do**
3. **if** $\tau = 0$ **then**
4. Set $\boldsymbol{v}_\tau^{(i)} = \nabla f_i(\boldsymbol{w}_\tau^{(i)})$ and $\widehat{\boldsymbol{v}}_\tau^{(i)} = \nabla f_i(\widehat{\boldsymbol{w}}_\tau^{(i)})$.
5. **else**
6. Pick a random batch of samples in client i, say $\mathcal{B}_\tau^{(i)}$.
7. Set $\boldsymbol{v}_\tau^{(i)} = \widetilde{\nabla} f_i(\boldsymbol{w}_\tau^{(i)}; \mathcal{B}_\tau^{(i)}) + (\boldsymbol{v}_{\tau-1}^{(i)} - \widetilde{\nabla} f_i(\boldsymbol{w}_{\tau-1}^{(i)}; \mathcal{B}_\tau^{(i)}))$ and $\widehat{\boldsymbol{v}}_\tau^{(i)} = \widetilde{\nabla} f_i(\widehat{\boldsymbol{w}}_\tau^{(i)}; \mathcal{B}_\tau^{(i)}) + (\widehat{\boldsymbol{v}}_{\tau-1}^{(i)} - \widetilde{\nabla} f_i(\widehat{\boldsymbol{w}}_{\tau-1}^{(i)}; \mathcal{B}_\tau^{(i)}))$. `// (Local Momentum)`
8. **end if**
9. Update $\boldsymbol{w}_{\tau+1}^{(i)} = \boldsymbol{w}_\tau^{(i)} - \eta \boldsymbol{v}_\tau^{(i)}$ and $\widehat{\boldsymbol{w}}_{\tau+1}^{(i)} = \widehat{\boldsymbol{w}}_\tau^{(i)} - \eta \widehat{\boldsymbol{v}}_\tau^{(i)}$.
10. **end for**
11. Send $\boldsymbol{d} := \boldsymbol{w}_0^{(i)} - \boldsymbol{w}_E^{(i)}$ and $\widehat{\boldsymbol{d}} := \widehat{\boldsymbol{w}}_0^{(i)} - \widehat{\boldsymbol{w}}_E^{(i)}$ to the server.

References

[Arj+19] Yossi Arjevani et al. "Lower bounds for non-convex stochastic optimization". In: *arXiv preprint arXiv:1912.02365* (2019).

[CO19] Ashok Cutkosky and Francesco Orabona. "Momentum-based variance reduction in non-convex SGD". In: *Advances in Neural Information Processing Systems*. 2019, pp. 15236–15245.

[Kar+20] Sai Praneeth Karimireddy et al. "Scaffold: Stochastic controlled averaging for federated learning". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5132–5143.

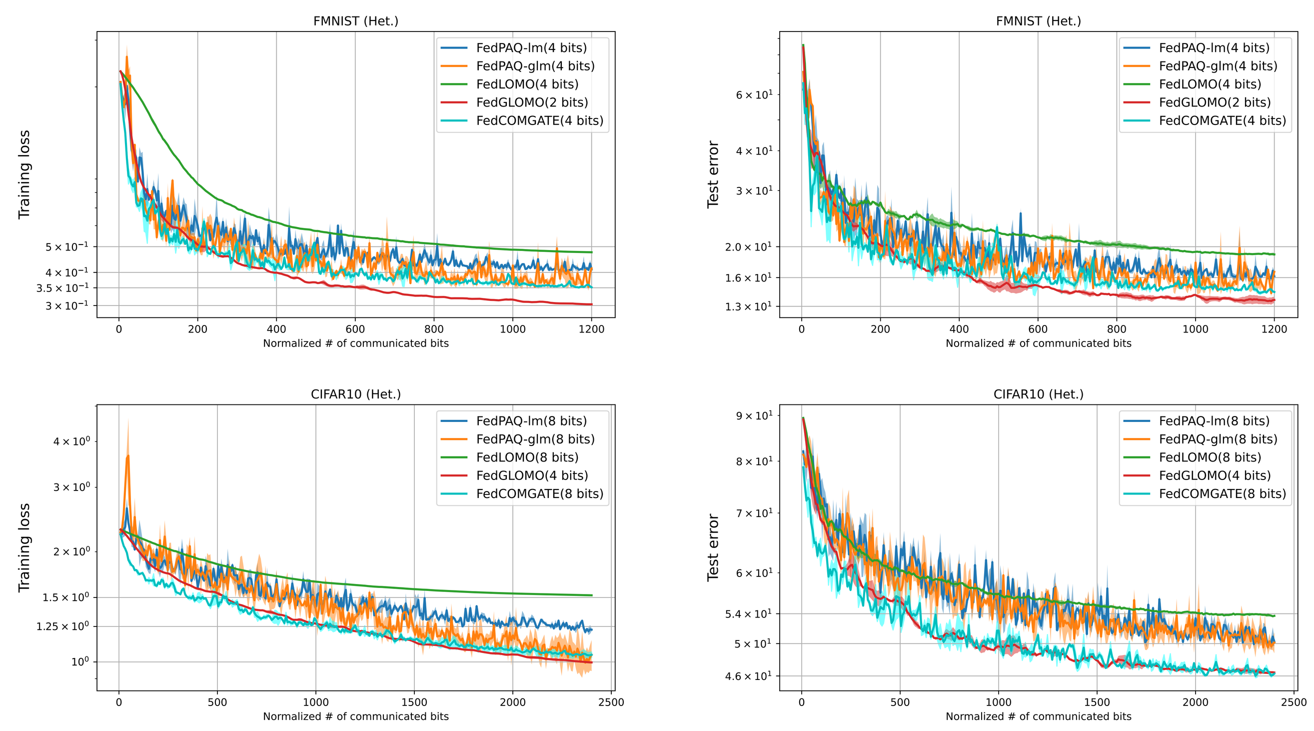
[McM+17] Brendan McMahan et al. "Communication-efficient learning of deep networks from decentralized data". In: *Artificial Intelligence and Statistics*. PMLR. 2017, pp. 1273–1282.

Convergence Guarantee

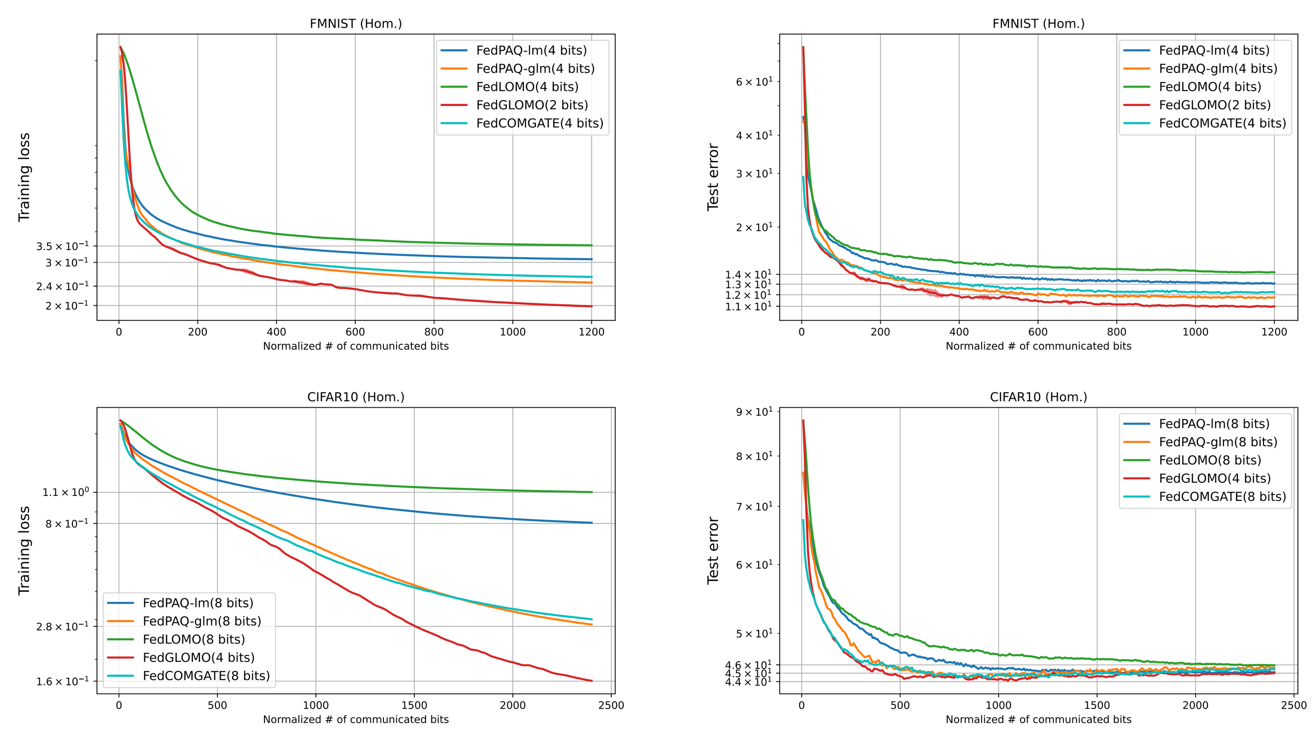
Theorem 1 (Informal)**.** *Suppose $\ell(., \boldsymbol{x})$ is smooth non-convex and non-negative $\forall\, \boldsymbol{x}$. Use full-device participation **only** for $k = 0$. Then, there exists $\eta_k = \eta$ and $\beta_k = \beta$ for all rounds k, such that `FedGLOMO` can achieve $\mathbb{E}_{k^*\sim \mathrm{Unif}[0,K-1]}[\|\nabla f(\boldsymbol{w}_{k^*})\|^2] \leq \epsilon$ in $K = \mathcal{O}(\epsilon^{-1.5})$ communication rounds and $E = \mathcal{O}(1)$ local steps.*

- `FedGLOMO` requires $T = KE = \mathcal{O}(\epsilon^{-1.5})$ gradient updates which matches the lower bound of [Arj+19]; **global momentum** is crucial to attain $\mathcal{O}(\epsilon^{-1.5})$ complexity. In contrast, `FedAvg` and most other FL algorithms require $\mathcal{O}(\epsilon^{-2})$ gradient updates.
- Our result also holds with compressed client-to-server communication (although the algorithm needs to be changed slightly); in fact, ours is the first such FL algorithm.

Empirical Results



Heterogeneous (Het.) Setting: 50 clients with each client having data of at most **2** classes. 50% device participation and 20 local steps per round.



Homogeneous (Hom.) Setting: 50 clients with each client having data from **all** classes in equal proportion. 50% device participation and 20 local steps per round.

Algo.	**CIFAR-10 Het.**	**FMNIST Het.**
`FedPAQ-lm`	50.26 ± 0.85	16.17 ± 0.53
`FedPAQ-glm`	49.88 ± 1.15	15.87 ± 1.10
`FedLOMO`	53.74 ± 0.17	18.95 ± 0.19
`FedGLOMO`	$\mathbf{46.42 \pm 0.05}$	$\mathbf{13.55 \pm 0.32}$
`FedCOMGATE`	$\mathbf{46.26 \pm 0.25}$	15.32 ± 0.09
Algo.	**CIFAR-10 Hom.**	**FMNIST Hom.**
`FedPAQ-lm`	$\mathbf{45.13 \pm 0.07}$	13.08 ± 0.05
`FedPAQ-glm`	45.70 ± 0.10	11.76 ± 0.06
`FedLOMO`	45.96 ± 0.01	14.22 ± 0.01
`FedGLOMO`	$\mathbf{44.97 \pm 0.05}$	$\mathbf{10.98 \pm 0.05}$
`FedCOMGATE`	45.46 ± 0.03	12.24 ± 0.01

Table 1: **Compressed Communication:** Average **test error** % ($\pm$ standard deviation) over the last five rounds.

Algo.	**CIFAR-10 Het.**	**FMNIST Het.**
`FedAvg-glm`	50.26 ± 0.74	16.17 ± 0.53
`MimeSGDm`	46.10 ± 0.13	$\mathbf{13.34 \pm 0.25}$
`FedGLOMO`	$\mathbf{45.41 \pm 0.15}$	$\mathbf{13.48 \pm 0.26}$

Table 2: **No Compression:** Average **test error** % ($\pm$ standard deviation) over the last five rounds.