LAG: Lazily Aggregated Gradient for Communication-Efficient Distributed Learning

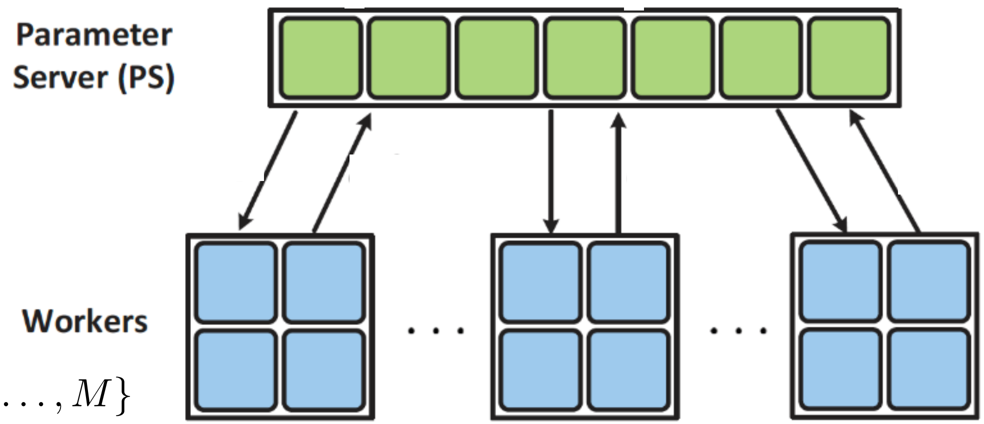
Tianyi Chen *Georgios Giannakis* *Tao Sun* *Wotao Yin*

UMN, ECE
UCLA, Math

NeurIPS 2018

Overview

$$\underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\text{minimize}} \ \mathcal{L}(\boldsymbol{\theta}) \quad \text{with} \quad \mathcal{L}(\boldsymbol{\theta}) := \sum_{m \in \mathcal{M}} \mathcal{L}_m(\boldsymbol{\theta})$$

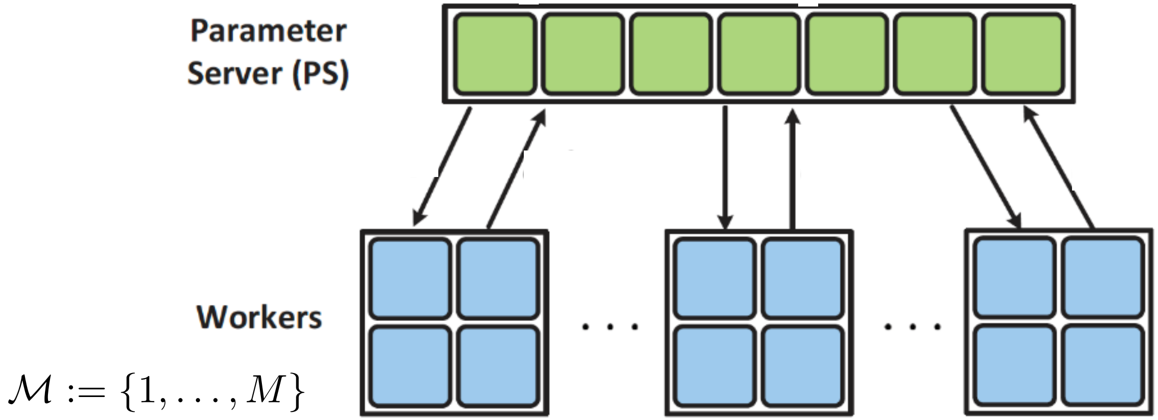


$$\mathcal{M} := \{1, \ldots, M\}$$

J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le et al., "Large-scale distributed deep networks," *Proc. NIPS*., Lake Tahoe, NV, pp. 1223–1231, 2012

Overview

$$\underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\text{minimize}}\ \mathcal{L}(\boldsymbol{\theta}) \quad \text{with} \quad \mathcal{L}(\boldsymbol{\theta}) := \sum_{m\in\mathcal{M}} \mathcal{L}_m(\boldsymbol{\theta})$$

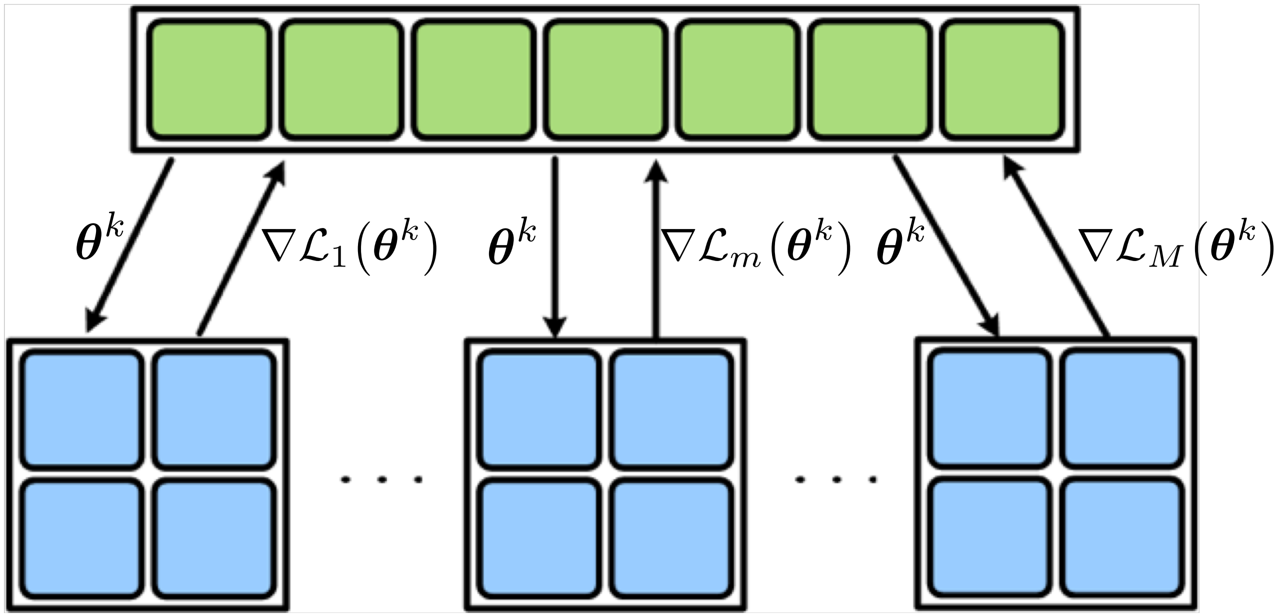


- ❑ Solvers: gradient descent (GD), momentum methods…
- ❑ Our method improves GD by
  - same convergence rate in theory
  - reduced communication in theory
  - more than *90%* communication saving in practice

J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le et al., "Large-scale distributed deep networks," *Proc. NIPS*., Lake Tahoe, NV, pp. 1223–1231, 2012

Vanilla GD implementation

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^{k} - \alpha \sum_{m \in \mathcal{M}} \nabla \mathcal{L}_m(\boldsymbol{\theta}^k)$$



- Per iteration communication overhead for *M* uploads (one per worker)

Prior art

- Communication-efficient distributed learning
  - Quantized gradient descent [Kashyap et al., 07], [Alistarh et al., 17], [Suresh et al., 17]…
  - Increasing computation before communication [Jaggi et al., 14], [Ma et al., 17], [Smith et al., 17]…
  - Sparse SGD with large entries [Aji-Heafield 17], [Sun et al., 17], [Lin et al., 18], [Stich et al., 18]…
    - number of communication rounds is not reduced

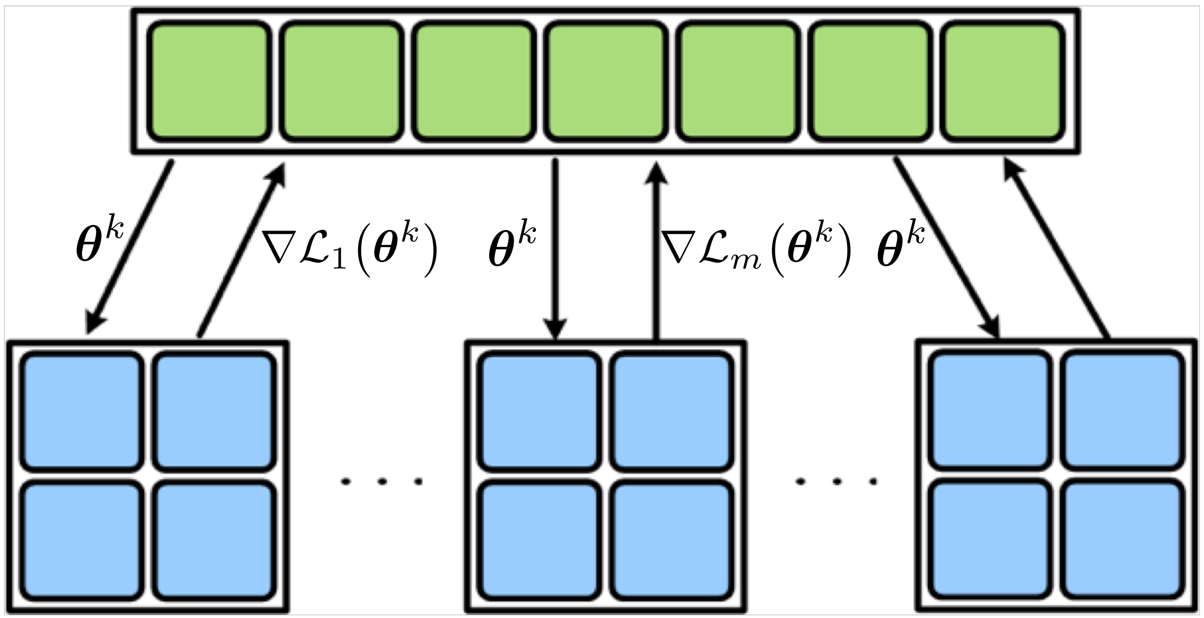
Our contribution

Adaptively skip communication, provable communication reduction

Our LAG implementation

Fresh gradient Old gradient

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \alpha \sum_{m \in \mathcal{M}^k} \nabla \mathcal{L}_m(\boldsymbol{\theta}^k) - \alpha \sum_{m \in \mathcal{M}/\mathcal{M}^k} \nabla \mathcal{L}_m(\hat{\boldsymbol{\theta}}_m^{k-1})$$



- Select a subset of workers $\mathcal{M}^k \subseteq \mathcal{M}$ to upload
- Remaining workers in $\mathcal{M}/\mathcal{M}^k$ do not upload

LAG: GD under two alternative communication rules

- Worker-side rule (LAG-WK): Include worker *m* in $\mathcal{M}^k$ if

Old gradient

$$\left\|\nabla\mathcal{L}_m(\boldsymbol{\theta}^k)-\nabla\mathcal{L}_m(\hat{\boldsymbol{\theta}}_m^{k-1})\right\|\geq\frac{1}{M}\left\|\frac{1}{\alpha}\left(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{k-1}\right)\right\|$$

Gradient innovation — Optimization progress

T. Chen, G. B. Giannakis, T. Sun, and W. Yin "LAG: Lazily Aggregated Gradient for Communication-Efficient Distributed Learning," *Proc. of NIPS*, Montreal, Canada, December 3-8, 2018.

LAG: GD under two alternative communication rules

- Worker-side rule (LAG-WK): Include worker *m* in $\mathcal{M}^k$ if

Old gradient

$$\left\|\nabla\mathcal{L}_m(\boldsymbol{\theta}^k)-\nabla\mathcal{L}_m(\hat{\boldsymbol{\theta}}_m^{k-1})\right\|\geq\frac{1}{M}\left\|\frac{1}{\alpha}\left(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{k-1}\right)\right\|$$

Gradient innovation — Optimization progress

- Server-side rule (LAG-PS): Include worker *m* in $\mathcal{M}^k$ if

L_m: smoothness of $\mathcal{L}_m$

$$L_m\left\|\boldsymbol{\theta}^k-\hat{\boldsymbol{\theta}}_m^{k-1}\right\|\geq\frac{1}{M}\left\|\frac{1}{\alpha}\left(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{k-1}\right)\right\|$$

  - LAG-PS is a **sufficient condition** for LAG-WK.

T. Chen, G. B. Giannakis, T. Sun, and W. Yin "LAG: Lazily Aggregated Gradient for Communication-Efficient Distributed Learning," *Proc. of NIPS*, Montreal, Canada, December 3-8, 2018.

Iteration and communication complexity

(nonconvex) Local loss $\mathcal{L}_m(\boldsymbol{\theta})$ is smooth.

(convex) Loss $\mathcal{L}(\boldsymbol{\theta})$ is convex.

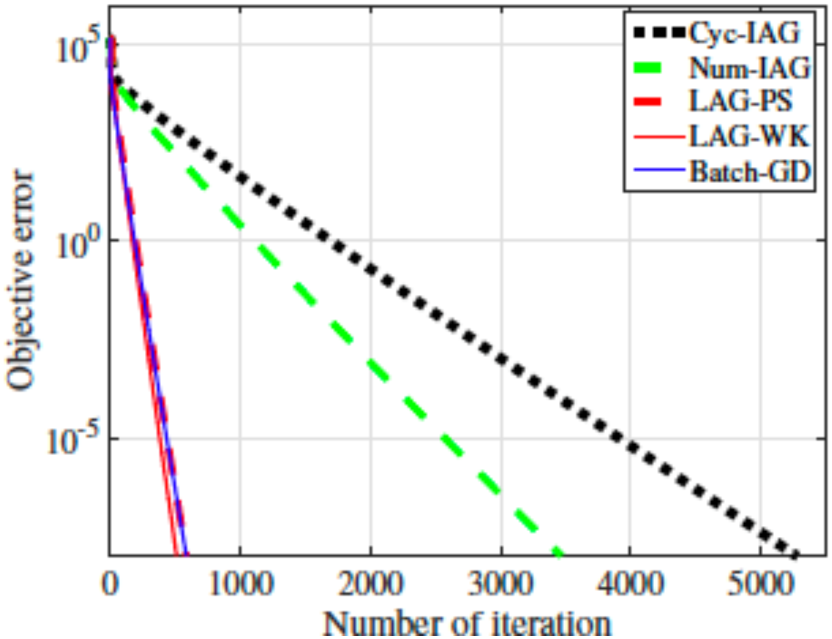
(strongly convex) Loss $\mathcal{L}(\boldsymbol{\theta})$ is (restricted) strongly convex.

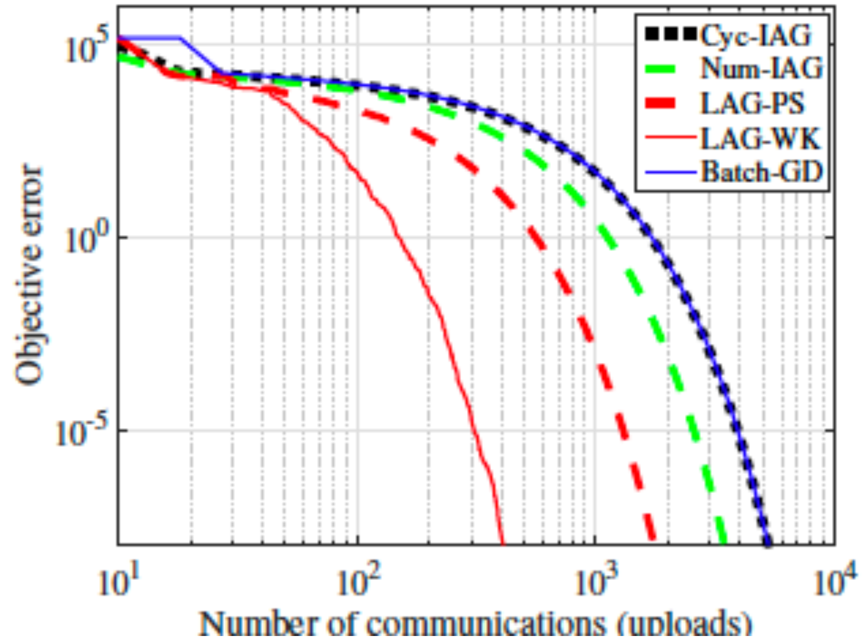
Theorem 1 In all cases, LAG enjoys the **same convergence rate** as GD.

Theorem 2 If local objectives are heterogeneous, LAG requires **smaller number of uploads** to a given accuracy than GD; e.g., as small as *1/M*.

Linear prediction

- Real datasets distributed on M = 9 workers



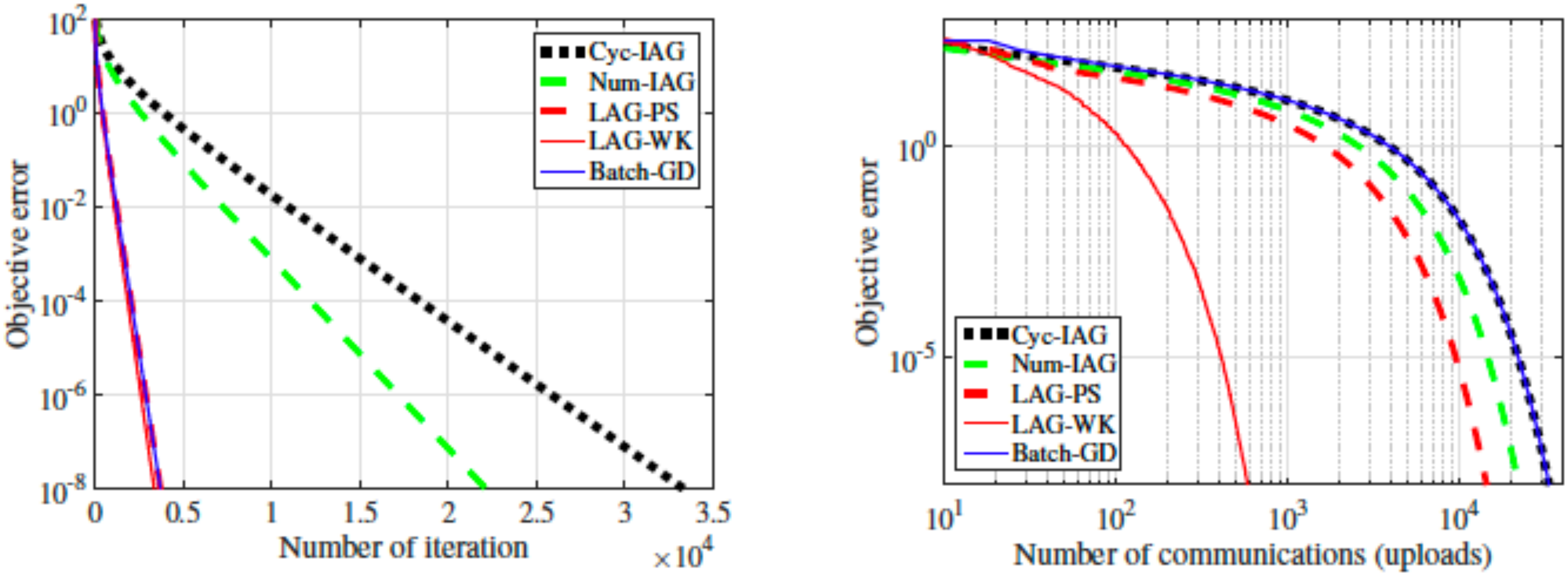


Cyc-/Num-IAG: cyclic/non-uniform update of incremental aggregated gradient

M. Lichman, "UCI machine learning repository," 2013. [Online]: http://archive.ics.uci.edu/ml

Logistic regression

❑ Real datasets distributed on M = 9 workers



➢ LAG needs same number of iterations but fewer uploads

Conclusions

- Adaptive communication rules for distributed learning
- Not degrade convergence but reduce communication

Thank You!

Thu Dec 6th 05:00 -- 07:00 PM @ Room 210 & 230 AB #8