On the Computation and Communication Complexity of Parallel SGD with Dynamic Batch Sizes for Stochastic Non-Convex Optimization

Poster @ Pacific Ballroom #103

Hao Yu, Rong Jin
Machine Intelligence Technology
Alibaba Group (US) Inc., Bellevue, WA

Stochastic Non-Convex Optimization

- Stochastic non-convex optimization

$$\min_{x \in \mathscr{R}^m} f(x) \triangleq \mathbb{E}_{\zeta \sim D}[F(x; \zeta)]$$

Stochastic Non-Convex Optimization

- Stochastic non-convex optimization

$$\min_{x \in \mathscr{R}^m} f(x) \triangleq \mathbb{E}_{\zeta \sim D}[F(x; \zeta)]$$

- SGD:

$$x_{t+1} = x_t - \gamma \boxed{\frac{1}{B} \sum_{i=1}^{B} \nabla F(x_t; \zeta_i)}$$

stochastic gradient averaged from a mini-batch of size B

Stochastic Non-Convex Optimization

- Stochastic non-convex optimization

$$\min_{x \in \mathscr{R}^m} f(x) \triangleq \mathbb{E}_{\zeta \sim D}[F(x; \zeta)]$$

- SGD: $$x_{t+1} = x_t - \gamma \boxed{\frac{1}{B} \sum_{i=1}^{B} \nabla F(x_t; \zeta_i)}$$

stochastic gradient averaged from a mini-batch of size B

- Singe node training:
  - Larger *B* can improve the utilization of computing hardware

Stochastic Non-Convex Optimization

- Stochastic non-convex optimization

$$\min_{x \in \mathscr{R}^m} f(x) \triangleq \mathbb{E}_{\zeta \sim D}[F(x;\zeta)]$$

- SGD:

$$x_{t+1} = x_t - \gamma \boxed{\frac{1}{B}\sum_{i=1}^{B} \nabla F(x_t;\zeta_i)}$$

stochastic gradient averaged from a mini-batch of size B

- Singe node training:
  - Larger *B* can improve the utilization of computing hardware
- Data-parallel training:
  - Multiple nodes form a bigger “mini-batch” by aggregating individual mini-batch gradients at each step.
  - Given a budget of gradient access, larger batch size yields fewer update/comm steps

Batch size for (parallel) SGD

- Question: Should always use a BS as large as possible in (parallel) SGD?

Batch size for (parallel) SGD

- Question: Should always use a BS as large as possible in (parallel) SGD?
    - You may tend to say “yes” because in strongly convex case, SGD with extremely large BS is close to GD?

Batch size for (parallel) SGD

- Question: Should always use a BS as large as possible in (parallel) SGD?
  - You may tend to say "yes" because in strongly convex case, SGD with extremely large BS is close to GD?
  - Theoretically, No! [Bottou&Bousquet'08] [Bottou et. al.'18] shows that with limited budgets of stochastic gradient (Stochastic First Order) access, GD (SGD with extremely large BS) has slower convergence than SGD with small batch sizes.

Batch size for (parallel) SGD

- Question: Should always use a BS as large as possible in (parallel) SGD?
  - You may tend to say "yes" because in strongly convex case, SGD with extremely large BS is close to GD?
  - Theoretically, No! [Bottou&Bousquet'08] [Bottou et. al.'18] shows that with limited budgets of stochastic gradient (Stochastic First Order) access, GD (SGD with extremely large BS) has slower convergence than SGD with small batch sizes.
- Under a finite SFO access budget, [Bottou et. al.'18] shows SGD with B=1 achieves better stochastic opt error than GD.

Batch size for (parallel) SGD

- Question: Should always use a BS as large as possible in (parallel) SGD?
  - You may tend to say "yes" because in strongly convex case, SGD with extremely large BS is close to GD?
  - Theoretically, No! [Bottou&Bousquet'08] [Bottou et. al.'18] shows that with limited budgets of stochastic gradient (Stochastic First Order) access, GD (SGD with extremely large BS) has slower convergence than SGD with small batch sizes.
- Under a finite SFO access budget, [Bottou et. al.'18] shows SGD with B=1 achieves better stochastic opt error than GD.
- Recall B=1 means poor hardware utilization and huge communication cost

Dynamic BS: reduce communication without sacrificing SFO convergence

- Motivating result:
  For strongly convex stochastic opt, [Friedlander&Schmidt'12] and [Bottou et.al.'18] show that SGD with exponentially increasing BS can achieve the same $O(1/T)$ SFO convergence as SGD with fixed small BS

Dynamic BS: reduce communication without sacrificing SFO convergence

- Motivating result:
  For strongly convex stochastic opt, [Friedlander&Schmidt'12] and [Bottou et.al.'18] show that SGD with exponentially increasing BS can achieve the same $O(1/T)$ SFO convergence as SGD with fixed small BS

- This paper explores how to use dynamic BS for non-convex opt such that:

Dynamic BS: reduce communication without sacrificing SFO convergence

- Motivating result:
  For strongly convex stochastic opt, [Friedlander&Schmidt'12] and [Bottou et.al.'18] show that SGD with exponentially increasing BS can achieve the same $O(1/T)$ SFO convergence as SGD with fixed small BS

- This paper explores how to use dynamic BS for non-convex opt such that:

  - do not sacrifice SFO convergence in (***parallel***) SGD. Recall (N node parallel) SGD with (B=1) has $O(1/\sqrt{\boxed{N}T})$ SFO convergence

T: SFO access budge at each node

Linear speedup w.r.t. # of nodes; computation power perfectly scaled out

Dynamic BS: reduce communication without sacrificing SFO convergence

- Motivating result:
  For strongly convex stochastic opt, [Friedlander&Schmidt'12] and [Bottou et.al.'18] show that SGD with exponentially increasing BS can achieve the same $O(1/T)$ SFO convergence as SGD with fixed small BS

- This paper explores how to use dynamic BS for non-convex opt such that:
  - do not sacrifice SFO convergence in (***parallel***) SGD. Recall (N node parallel) SGD with (B=1) has $O(1/\sqrt{\boxed{N}T})$ SFO convergence

T: SFO access budge at each node

Linear speedup w.r.t. # of nodes; computation power perfectly scaled out

Dynamic BS: reduce communication without sacrificing SFO convergence

- Motivating result:
  For strongly convex stochastic opt, [Friedlander&Schmidt'12] and [Bottou et.al.'18] show that SGD with exponentially increasing BS can achieve the same $O(1/T)$ SFO convergence as SGD with fixed small BS

- This paper explores how to use dynamic BS for non-convex opt such that:

  - do not sacrifice SFO convergence in (***parallel***) SGD. Recall (N node parallel) SGD with (B=1) has $O(1/\sqrt{\boxed{N}T})$ SFO convergence

    T: SFO access budge at each node

    Linear speedup w.r.t. # of nodes; computation power perfectly scaled out

  - reduce communication complexity (# of used batches) in ***parallel*** SGD

Non-Convex under PL condition

- PL condition: $\frac{1}{2}\|\nabla f(x)\|^2 \geq \mu(f(x) - f^*), \forall x$
  - Milder than strong convexity: strong convexity implies PL condition.
  - Non-convex fun under PL is typically as nice as strong convex fun.

Algorithm 1 CR-PSGD$(f, N, T, \mathbf{x}_1, B_1, \rho, \gamma)$

Input: N, T, $\mathbf{x}_1 \in \mathbb{R}^m$, γ , B_1 and $\rho > 1$.
Initialize $t = 1$
budge of SFO access at each worker
while $\sum_{\tau=1}^{t} B_\tau \leq \boxed{T}$ **do**
  Each worker calculates batch gradient average $\bar{\mathbf{g}}_{t,i} = \frac{1}{B_t}\sum_{j=1}^{B_t} F(\mathbf{x}_t; \zeta_{i,j})$.
  Each worker aggregates gradient average $\bar{\mathbf{g}}_t = \frac{1}{N}\sum_{i=1}^{N} \bar{\mathbf{g}}_{t,i}$.
  Each worker updates in parallel via: $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma\bar{\mathbf{g}}_t$.
  Set batch size $B_{t+1} = \lfloor \rho^t B_1 \rfloor$.
  Update $t \leftarrow t + 1$.
end while
Return: $\mathbf{x}_t$

Non-Convex under PL condition

- Under PL, we show using exponentially increasing batch sizes in PSGD with *N* workers has $O(\frac{1}{NT})$ SFO convergence with $O(\log T)$ comm rounds
    - SoA $O(\frac{1}{NT})$ SFO convergence with $O(\sqrt{NT})$ inter-worker comm rounds attained by local SGD in [Stich'18] for strongly convex opt only

Non-Convex under PL condition

- Under PL, we show using exponentially increasing batch sizes in PSGD with N workers has $O(\frac{1}{NT})$ SFO convergence with $O(\log T)$ comm rounds
  - SoA $O(\frac{1}{NT})$ SFO convergence with $O(\sqrt{NT})$ inter-worker comm rounds attained by local SGD in [Stich'18] for strongly convex opt only
- How about general non-convex without PL?

Non-Convex under PL condition

- Under PL, we show using exponentially increasing batch sizes in PSGD with *N* workers has $O(\frac{1}{NT})$ SFO convergence with $O(\log T)$ comm rounds
    - SoA $O(\frac{1}{NT})$ SFO convergence with $O(\sqrt{NT})$ inter-worker comm rounds attained by local SGD in [Stich'18] for strongly convex opt only
- How about general non-convex without PL?
- Inspiration from "**catalyst acceleration**" developed in [Lin et al.'15][Paquette et al.'18]
    - Instead of solving original problem directly, it repeatedly solves "strongly convex" proximal minimization

General Non-Convex Opt

- A new catalyst-like parallel SGD method

Algorithm 2 CR-PSGD-Catalyst$(f, N, T, \mathbf{y}_0, B_1, \rho, \gamma)$

Input: N, T, θ, $\mathbf{y}_0 \in \mathbb{R}^m$, γ , B_1 and $\rho > 1$.
Initialize $\mathbf{y}^{(0)} = \mathbf{y}_0$ and $k = 1$.
while $k \leq \lfloor \sqrt{NT} \rfloor$ **do**

strongly convex fun whose unbiased stochastic gradient is easily estimated

  Define $\boxed{h_\theta(\mathbf{x}; \mathbf{y}^{(k-1)}) \overset{\Delta}{=} f(\mathbf{x}) + \frac{\theta}{2}\|\mathbf{x} - \mathbf{y}^{(k-1)}\|^2}$.
  Update $\mathbf{y}^{(k)}$ via

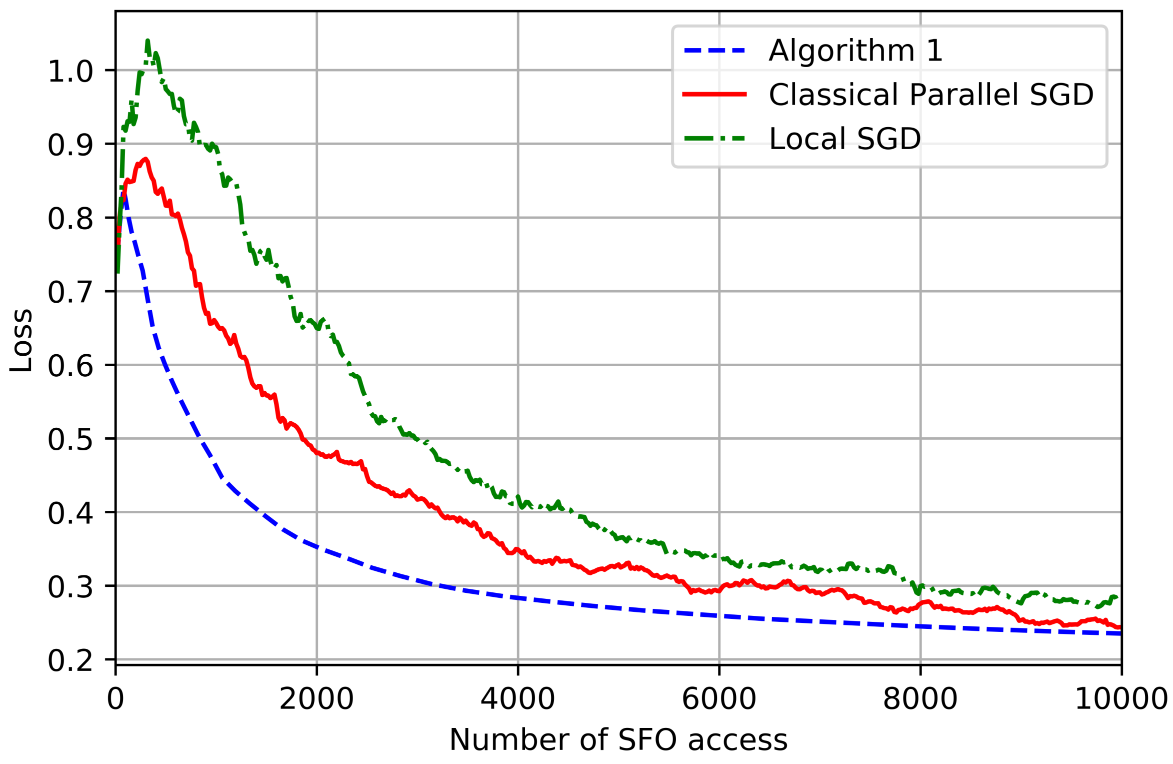
$$\mathbf{y}^{(k)} = \text{CR-PSGD}(h_\theta(\cdot; \mathbf{y}^{(k-1)}), N, \lfloor \sqrt{T/N} \rfloor, \mathbf{y}^{(k-1)}, B_1, \rho, \gamma)$$

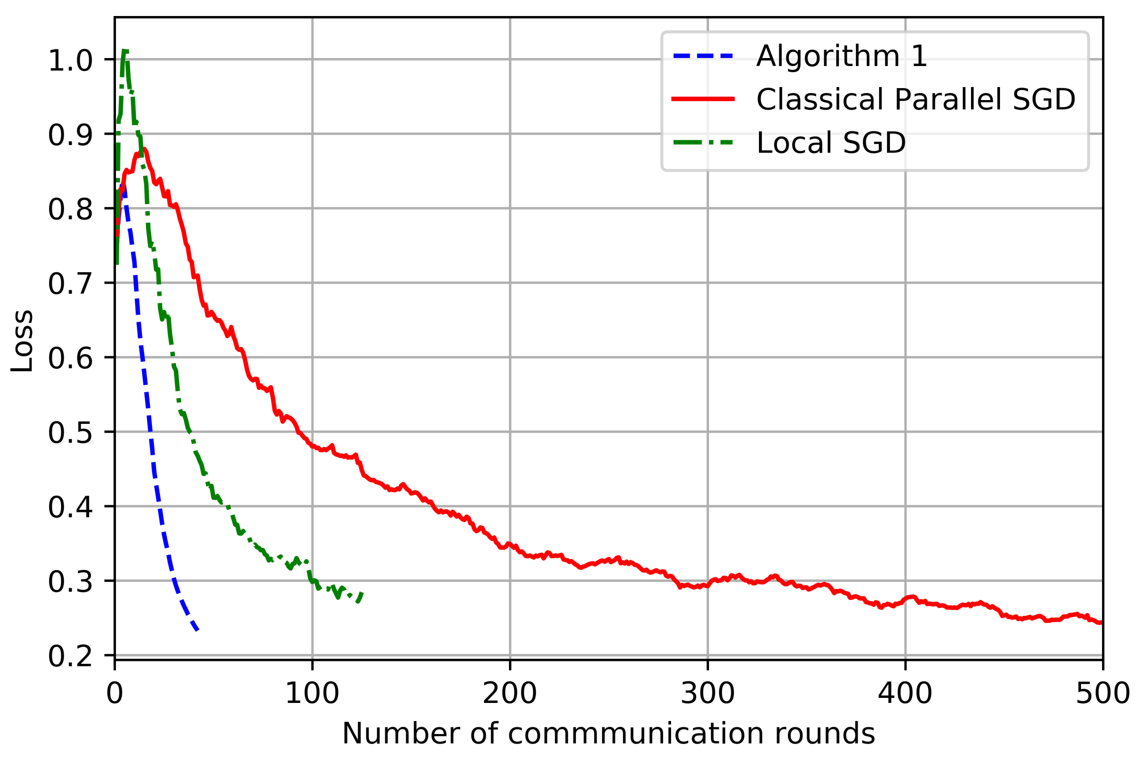
  Update $k \leftarrow k + 1$.
end while

- We show this catalyst-like parallel SGD (with dynamic BS) has $O(1/\sqrt{NT})$ SFO convergence with $O(\sqrt{NT}\log(\frac{T}{N}))$ comm rounds
  - SoA is $O(1/\sqrt{NT})$ SFO convergence with $O(N^{3/4}T^{3/4})$ inter-worker comm rounds

Experiments

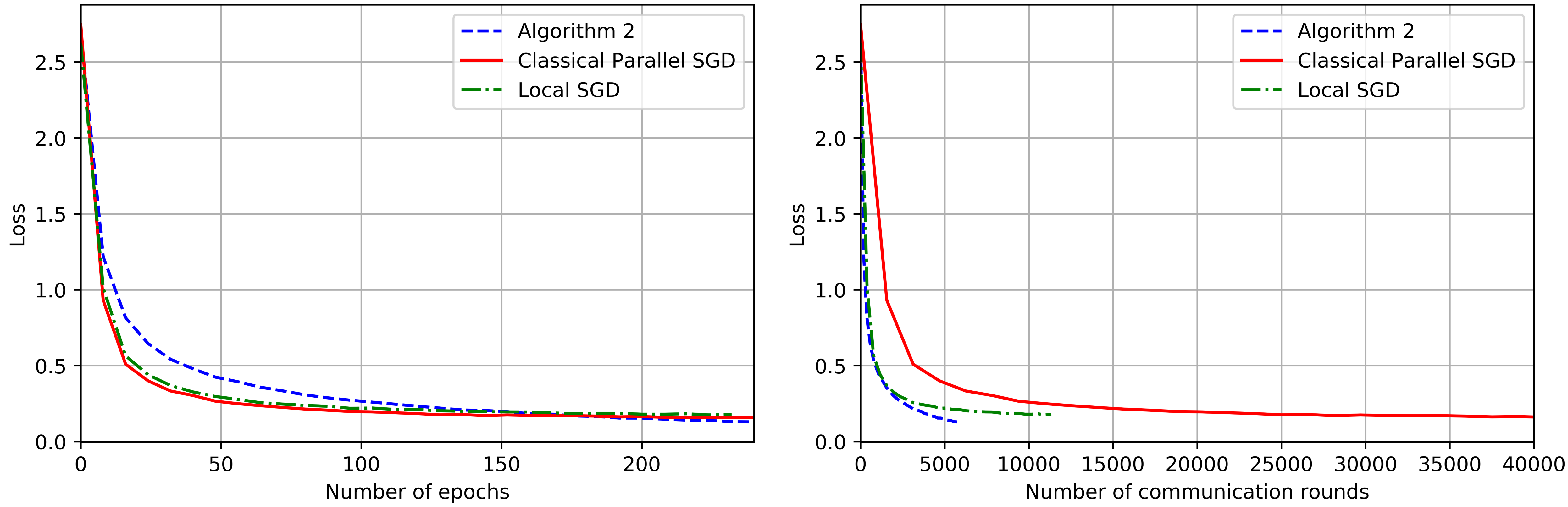
Distributed Logistic Regression: N=10





Experiments

Training ResNet20 over Cifar10: N=8



Thanks!

Poster on Wed Jun 12th 06:30 -- 09:00 PM @ Pacific Ballroom #103