Decentralized Optimization

Jingchang Liu

September 21, 2017

University of Science and Technology of China

Table of Contents

Decentralized Parallel SGD

Reference

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu, "Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent", NIPS 2017 (oral: rate below 1.2%)

Centralized and Decentralized optimization

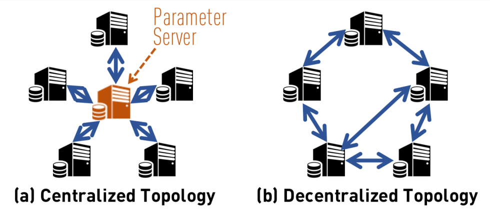


Figure 1: Different between Centralized and Decentralized optimization

Why decentralized optimization

- Underlying network topology.
- Less communication cost on the busiest node.
- Can decentralized algorithms be faster than its centralized counterpart?

Centralized SGD

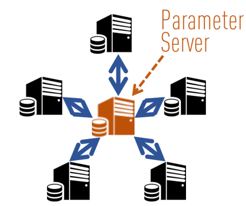


Figure 2: Centralized Topology

P-SGD

1. Formulation: $\min\limits_{x} \sum\limits_{i=1}^{n} f_i(x)$
2. x is located in the master.
3. Workers Calculate stochastic gradient: $\nabla f_i(x)$
4. Master Update x: $x := x - \eta \nabla f_i(x)$

Decentralized SGD

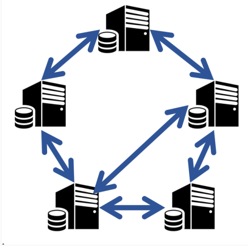


Figure 3: Centralized Topology

DP-SGD

1. x is located in each clients.
2. $x_1 = x_2 = \cdots = x_N$.

Framework

For each node i, repeat

1. Do gradient update with own data
2. Regularly exchange some information with neighbors
3. Combine information according to some police

Topology (V, W)

- V a set of n computational nodes, $V := \{1, 2, \cdots, n\}$
- $W \in \mathbb{R}^{n \times n}$, $(i) W_{ij} \in [0, 1]$, $\forall i, j$, $(ii) W_{ij} = W_{ji}$, $\forall i, j$, $(iii) \sum_j W_{ij} = 1$, $\forall i$

Result

- the local optimization variables in the nodes will converge together.

Algorithm

Algorithm 1 Decentralized Parallel Stochastic Gradient Descent (D-PSGD) on the ith node

Require: initial point $x_{0,i} = x_0$, step length γ, weight matrix W, and number of iterations K

for $k = 0, 1, 2, \ldots, K-1$ **do**
  Randomly sample $\xi_{k,i}$ from local data of the i-th node
  Compute a local stochastic gradient based on $\xi_{k,i}$ and current optimization variable $x_{k,i}$: $\nabla F_i(x_{k,i}; \xi_{k,i})$ [a]
  Compute the neighborhood weighted average by fetching optimization variables from neighbors: $x_{k+\frac{1}{2},i} = \sum_{j=1}^{n} W_{ij} x_{k,j}$ [b]
  Update the local optimization variable $x_{k+1,i} \leftarrow x_{k+\frac{1}{2},i} - \gamma \nabla F_i(x_{k,i}; \xi_{k,i})$[c]
end for
Output: $\frac{1}{n} \sum_{i=1}^{n} x_{K,i}$ [d]

Figure 4: Algorithm of DP-SGD

Iteration

- DP-SGD: $x_{(i)}^{k+1} = \sum_{j=1}^{n} w_{ij} x_{(j)}^{k} - \alpha^k \nabla f_{i_k}\left(x_{(i)}^{k}\right)$, for agent $i = 1, 2, \cdots, n$
- SGD: $x^{k+1} = x^k - \alpha^k \nabla f_i\left(x^k\right)$

Write together: $x^{k+1} = W x^k - \alpha^k \nabla f\left(x^k\right)$

Convergence rate analysis

- $\partial f\left(X_k\right) := \left[\nabla f_1(x_{k,1})\, \nabla f_2(x_{k,2})\, \cdots\, \nabla f_n(x_{k,n})\right]$

Th1

Under some assumptions(without convex), we have

$$\frac{1}{K}\left(\frac{1-\gamma L}{2}\sum_{k=0}^{K-1}\mathbb{E}\left\|\frac{\partial f(X_k)\mathbf{1}_n}{n}\right\|^2 + D_1\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\frac{X_k\mathbf{1}_n}{n}\right)\right\|^2\right)$$
$$\leq \quad \frac{f(0)-f^*}{\gamma K} + \frac{\gamma L}{2n}\sigma^2 + \frac{\gamma^2 L^2 n\sigma^2}{(1-\mu)D_2} + \frac{9\gamma^2 L^2 n\zeta^2}{(1-\sqrt{\mu})^2}$$

- Note: $\frac{X_k\mathbf{1}_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_{k,i}$

Convergence rate analysis

Corollary

Under the same assumptions as in Th1, set stepsize

$$\gamma = \frac{1}{2L + \sigma\sqrt{K/n}},$$

we have

$$\frac{\sum_{k=0}^{K} \mathbb{E}\left\|\nabla f\left(\frac{X_k \mathbf{1}_n}{n}\right)\right\|^2}{K} \leq \frac{8(f(0) - f^*)L}{K} + \frac{(8f(0) - 8f^* + 4L)\sigma}{\sqrt{Kn}}$$

- Note: the convergence rate is $O\left(\frac{1}{K} + \frac{1}{\sqrt{nK}}\right)$

Experiments

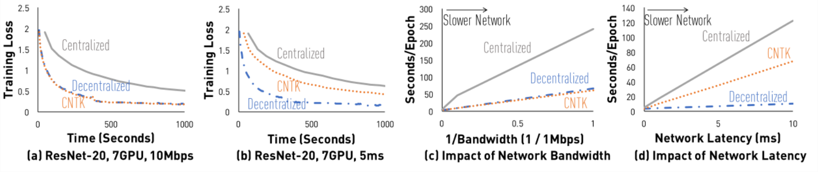


Figure 5: Comparison between D-PSGD and two centralized implementation

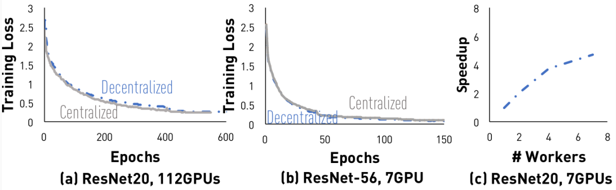
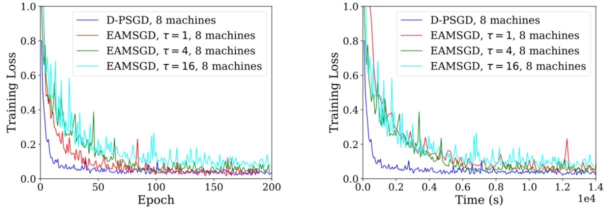


Figure 6: Convergence Rate and D-PSGD Speedup

Experiments



(a) Iteration vs Training Loss
(b) Time vs Training Loss

Figure 7: Convergence comparison between D-PSGD and EAMSGD

Experiments

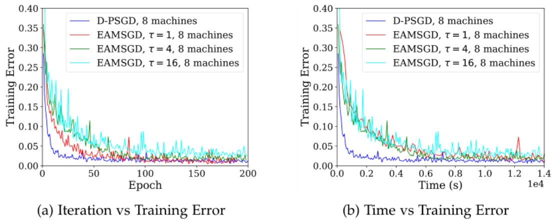


Figure 8: Training Error comparison between D-PSGD and EAMSGD

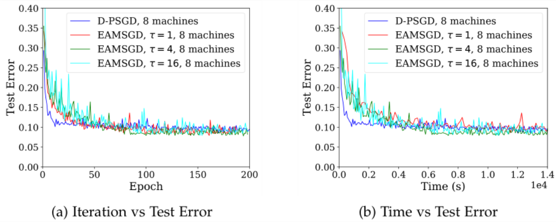
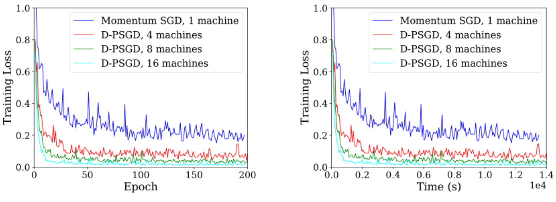


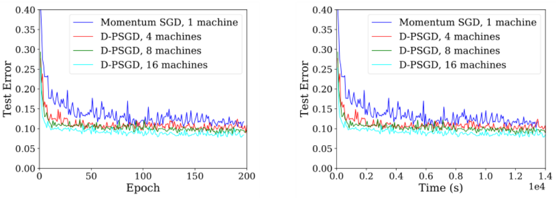
Figure 9: Test Error comparison between D-PSGD and EAMSGD

Experiment



(a) Iteration vs Training Loss (b) Time vs Training Loss

Figure 10: Training Loss on 1, 4, 8 and 16 machines



(a) Iteration vs Test Error (b) Time vs Test Error

Figure 11: Test Error on 1, 4, 8 and 16 machines

EXTRA: accelerate DP-SGD

Reference

Wei Shi, Qing Ling, Gang Wu, Wotao Yin: EXTRA: An exact first-order algorithm for decentralized consensus optimization. SIAM Journal on Optimization, 25(2): 944-966, 2015

Convergence Rate

- DP-SGD: similar to SGD
- EXTRA:
  - general convex: $O\left(\frac{1}{k}\right)$
  - (restricted) strongly convex: linear rate

inexact convergence of DP-SGD

1. $x^{k+1} = Wx^k - \alpha^k \nabla f(x^k), \;\; x^\infty = Wx^\infty - \alpha \nabla f(x^\infty)$
2. Consensus of x, $\;\; x^\infty = Wx^\infty, \;\; \nabla f(x^\infty) = 0$
3. $\nabla f_i(x_{(i)}^\infty) = 0, \forall i$
4. The same point $x_{(i)}^\infty$ simultaneously minimizes f_i for all agent i.

EXTRA iteration

Derivation

1. $x^{k+2} = Wx^{k+1} - \alpha \nabla f(x^{k+1})$
2. $x^{k+1} = \bar{W}x^k - \alpha \nabla f(x^k),\ \bar{W} = \frac{I+W}{2}$
3. $x^{k+2} - x^{k+1} = Wx^{k+1} - \bar{W}x^k - \alpha \nabla f(x^{k+1}) + \alpha \nabla f(x^k)$
4. $x^{k+2} = (I+W)x^{k+1} - \bar{W}x^k - \alpha[\nabla f(x^{k+1}) - \nabla f(x^k)]$
5. $x_{(i)}^{k+2} = x_{(i)}^{k+1} + \sum_{j=1}^{n} w_{ij} x_{(j)}^{k+1} - \sum_{j=1}^{n} \bar{w}_{ij} x_{(j)}^{k} - \alpha \left[\nabla f_i\left(x_{(i)}^{k+1}\right) - \nabla f_i\left(x_{(i)}^{k}\right)\right]$

Choose $\alpha > 0$ and mixing matrices $W \in \mathbb{R}^{n\times n}$ and $\tilde{W} \in \mathbb{R}^{n\times n}$;
Pick any $\mathbf{x}^0 \in \mathbb{R}^{n\times p}$;
1. $\mathbf{x}^1 \leftarrow W\mathbf{x}^0 - \alpha \nabla \mathbf{f}(\mathbf{x}^0)$;
2. **for** $k = 0, 1, \cdots$ **do**
$\quad \mathbf{x}^{k+2} \leftarrow (I+W)\mathbf{x}^{k+1} - \tilde{W}\mathbf{x}^k - \alpha\left[\nabla \mathbf{f}(\mathbf{x}^{k+1}) - \nabla \mathbf{f}(\mathbf{x}^k)\right]$;
end for

Figure 12: EXTRA algorithm

Experiments: Decentralized least squares

Problem

- $\min\limits_{x} f(x) = \frac{1}{n} \sum\limits_{i=1}^{n} \frac{1}{2} \left\| M_i x - y_i \right\|_2^2$

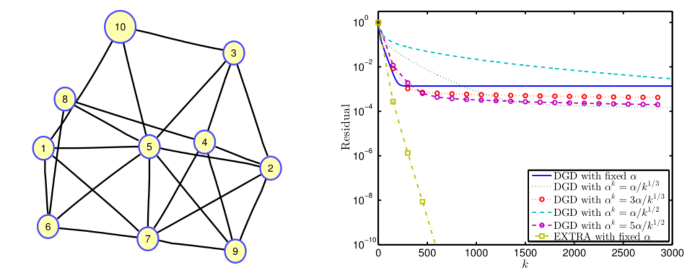


Figure 13: Plot of residual $\frac{\left\|x^k - x^*\right\|_F}{\left\|x^0 - x^*\right\|_F}$

Problem

- $\min_x f(x) = \frac{1}{n} \sum_{i=1}^{n} \left\{ \frac{1}{m_i} \sum_{j=1}^{m_i} \ln \left(1 + \exp\left(-\left(M_{(i)j} x\right) y_{(i)j}\right)\right) \right\}$

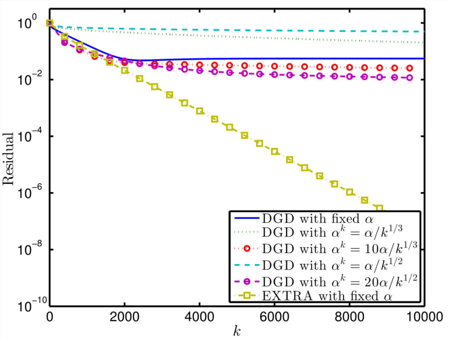


Figure 14: Plot of residual $\frac{\left\|x^k - x^*\right\|_F}{\left\|x^0 - x^*\right\|_F}$

More about decentralized optimization

Asynchronous Decentralized Optimization

Synchronous algorithm

- Wait until receives all necessary input.
- Send out until all of its neighbors finish computation.

Asynchronous algorithm

Each agent i asynchronous do:

1. Compute using the information it has available.
2. Send out x to neighbors.

Reference

Tianyu Wu, Kun Yuan, Qing Ling, Wotao Yin, Ali Sayed: Decentralized consensus optimization with asynchrony and delays. IEEE Transactions on Signal and Information Processing over Networks

Decentralized Optimization + SAGA

Iterations

- DGD: $x_n^{k+1} = \sum_{m=1}^{N} w_{nm} x_m^k - \alpha \nabla f_n\left(x_n^k\right)$
- EXTRA:
  $x_n^{k+1} = x_n^k + \sum_{m=1}^{N} w_{nm} x_m^k - \sum_{m=1}^{N} \bar{w}_{nm} x_m^{k-1} - \alpha\left[\nabla f_n(x_n^k) - \nabla f_n(x_n^{k-1})\right]$
- DSA:

$$
\begin{aligned}
\bar{g}_n^k &= \nabla f_{n,i_n^k}(x_n^k) - \nabla f_{n,i_n^k}(x_n^k) + \frac{1}{q_n}\sum_{i=1}^{q_n} \nabla f_{n,i}(y_{n,i}^k) \\
x_n^{k+1} &= x_n^k + \sum_{m=1}^{N} w_{nm} x_m^{k-1} - \sum_{m=1}^{N} \bar{w}_{nm} x_m^{k-1} - \alpha\left[\bar{g}_n^k - \bar{g}_n^{k-1}\right]
\end{aligned}
$$

Reference

A. Mokhtari and A. Ribeiro. Dsa: decentralized double stochastic averaging gradient algorithm. Journal of Machine Learning Research, 17(61):135, 2016

Experiments

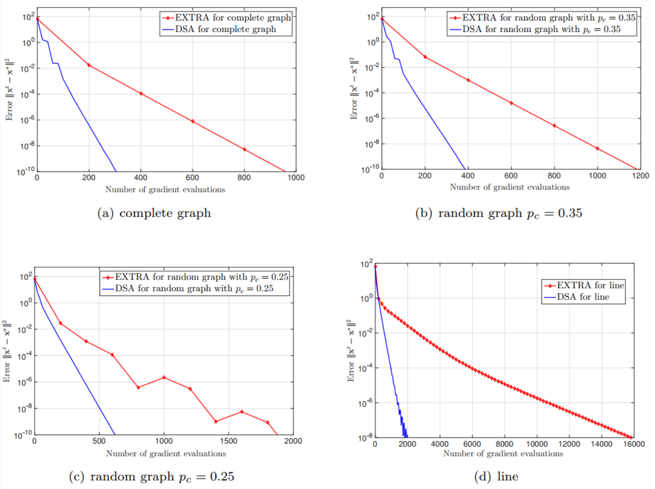


Figure 15: Convergence paths of DSA and EXTRA for different network topology.

Conclusion

Conclusions

- Decentralized optimization is different from centralized optimization.
- Decentralized optimization may be faster as centralized algorithm lies on high communication cost on the central node.
- EXTRA is a wonderful algorithm for decentralized optimization.
- Some ideas in centralized algorithm like variance reduction can be transferred to decentralized cases.
- There are still some works to do in decentralized optimization.

Q & A