Stochastic Decentralized Methods

Mingyi Hong

University Of Minnesota

Outline

- Standard SGD Method and Its Proof
- Stochastic Decentralized Algorithms and Applications
- Numerical Performance
- Proof for a Stochastic Gradient Tracking Algorithm

Centralized SGD Algorithms

Vanilla SGD Algorithm: Convergence Analysis

$$\begin{array}{ll} \text{minimize}_{\boldsymbol{x}} & F(\boldsymbol{x}) := \mathbb{E}[f(\boldsymbol{x}, \xi)] \\ \text{subject to} & \boldsymbol{x} \in \mathbb{R}^n \end{array}$$

- $f(\boldsymbol{x}, \xi)$ (objective or cost function) is differentiable for every ξ
- ξ is a random variables.

Example

Consider the finite-sum problem we discussed before

$$\begin{aligned} \text{minimize}_{\boldsymbol{x}} \quad & F(\boldsymbol{x}) := \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}, \{\mathbf{a}_i, b_i\}) \\ \text{subject to} \quad & \boldsymbol{x} \in \mathbb{R}^n \end{aligned}$$

- where $\{\mathbf{a}_i, b_i\}_{i=1}^{N}$ are N random samples
- If we draw the indices i from a uniform distribution over $i = 1 \cdots N$, then

$$F(\boldsymbol{x}) = \mathbb{E}_i[f(\boldsymbol{x}, \{\mathbf{a}_i, b_i\})].$$

The SGD Algorithm

- Consider the following GD algorithm

$$\begin{aligned}\boldsymbol{x}^{r+1} &= \boldsymbol{x}^t - \nabla F(\boldsymbol{x}^t) \\ &= \boldsymbol{x}^t - \nabla \mathbb{E}[f(\boldsymbol{x}^t, \xi)] \\ &= \boldsymbol{x}^t - \mathbb{E}[\nabla f(\boldsymbol{x}^t, \xi)]\end{aligned}$$

- The gradient cannot be exactly evaluated
- Use a sample to approximate it

The SGD Algorithm

- The SGD algorithm, first sample $\mathbf{g}^r$ as an unbiased estimator of $\mathbb{E}[\nabla f(\boldsymbol{x}^r, \xi)]$
- Then perform

$$\boldsymbol{x}^{r+1} = \boldsymbol{x}^r - \eta^r \mathbf{g}^r.$$

Convergence Analysis (f is Strongly Convex)

- Let $F(\boldsymbol{x})$ be L-smooth, and
  - 1) $F(\boldsymbol{x})$ is strongly convex, with constant μ
  - 2) $F(\boldsymbol{x})$ is non-convex
- $F(\boldsymbol{x})$ is lower bounded, $F(\boldsymbol{x}) \geq \underline{F}$, $\forall\ \boldsymbol{x}$
- Let $\mathbf{g}^r$ be an unbiased estimator of $\nabla F(\boldsymbol{x})$:

$$\mathbb{E}[\mathbf{g}^r(\boldsymbol{x}, \xi)] = \nabla F(\boldsymbol{x}). \tag{1.1}$$

- Assume

$$\mathbb{E}[\|\mathbf{g}^r(\boldsymbol{x}, \xi)\|^2] \leq \sigma_g^2 + c_g \|\nabla F(\boldsymbol{x})\|^2 \tag{1.2}$$

Convergence Analysis

Theorem 1.1 (Convergence of SGD for Strongly Convex f)

Under the assumption in the previous page, where f is μ strongly convex, if $c_g = 0$, and $\eta_r \leq \frac{1}{r \times \mu}$, then SGD achieves the following rate

$$\mathbb{E}[\|\boldsymbol{x}^{T+1} - \boldsymbol{x}^*\|^2] = \mathcal{O}(1/T); \tag{1.3}$$

Further, if $c_g \neq 0$, then if we choose $\eta_r \leq \frac{1}{r c_g L^2 \mu}$, we have

$$\mathbb{E}[\|\boldsymbol{x}^{T+1} - \boldsymbol{x}^*\|^2] = \mathcal{O}(1/T) \tag{1.4}$$

Proof Steps

- First, let us suppose that $c_g = 0$
- From the fact that F is strongly convex, we have

$$F(\boldsymbol{x}^*) - F(\boldsymbol{x}^r) \geq \langle \nabla F(\boldsymbol{x}^r), \boldsymbol{x}^* - \boldsymbol{x}^r \rangle + \frac{\mu}{2}\|\boldsymbol{x}^r - \boldsymbol{x}^*\|^2$$
$$F(\boldsymbol{x}^r) - F(\boldsymbol{x}^*) \geq \langle \nabla F(\boldsymbol{x}^*), \boldsymbol{x}^r - \boldsymbol{x}^* \rangle + \frac{\mu}{2}\|\boldsymbol{x}^r - \boldsymbol{x}^*\|^2$$

- Adding these together, we obtain

$$\begin{aligned} &\langle \nabla F(\boldsymbol{x}^r) - \nabla F(\boldsymbol{x}^*), \boldsymbol{x}^r - \boldsymbol{x}^* \rangle \\ &= \langle \nabla F(\boldsymbol{x}^r), \boldsymbol{x}^r - \boldsymbol{x}^* \rangle \geq \mu\|\boldsymbol{x}^r - \boldsymbol{x}^*\|^2 \end{aligned} \tag{1.5}$$

Proof Steps ($c_g = 0$)

- Similarly as the subgradient descent analysis, we have

$$\begin{aligned}\|\boldsymbol{x}^* - \boldsymbol{x}^{r+1}\|^2 &= \|\boldsymbol{x}^* - \boldsymbol{x}^r\|^2 + 2\langle \boldsymbol{x}^* - \boldsymbol{x}^r, \boldsymbol{x}^r - \boldsymbol{x}^{r+1}\rangle + \|\boldsymbol{x}^r - \boldsymbol{x}^{r+1}\|^2 \\ &= \|\boldsymbol{x}^* - \boldsymbol{x}^r\|^2 - 2\eta\langle \boldsymbol{x}^* + \boldsymbol{x}^r, \mathbf{g}^r\rangle + \eta^2\|\mathbf{g}^r\|^2\end{aligned}$$

- Taking an expectation, we obtain

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{x}^* - \boldsymbol{x}^{r+1}\|^2] &= \mathbb{E}[\|\boldsymbol{x}^* - \boldsymbol{x}^r\|^2] + 2\eta\mathbb{E}[\langle \boldsymbol{x}^* - \boldsymbol{x}^r, \mathbf{g}^r\rangle] + \eta^2\mathbb{E}[\|\mathbf{g}^r\|^2] \\ &= \mathbb{E}[\|\boldsymbol{x}^* - \boldsymbol{x}^r\|^2] + 2\eta\langle \boldsymbol{x}^* - \boldsymbol{x}^r, \nabla F(\boldsymbol{x}^r)\rangle + \eta^2\mathbb{E}[\|\mathbf{g}^r\|^2] \\ &\leq (1 - 2\mu\eta)\mathbb{E}[\|\boldsymbol{x}^* - \boldsymbol{x}^r\|^2] + \eta^2\sigma_g^2\end{aligned}$$

  where the last inequality comes from (1.5) and $c_g = 0$.

Proof Steps ($c_g = 0$)

- The final rate is proven by induction.
- At iteration $r = 1$, we have

$$\mathbb{E}\|\boldsymbol{x}^1 - \boldsymbol{x}^*\|^2 \le \max\{\mathbb{E}\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2, \sigma_g^2/\mu\} := L^0 \tag{1.6}$$

- Suppose for iteration r, the desired rate holds true, then for iteration $r+1$ (and use $\eta_r \le \frac{1}{r\mu^2}$)

$$\begin{aligned}
\mathbb{E}[\|\boldsymbol{x}^{r+1} - \boldsymbol{x}^*\|^2] &\le (1 - 2/r)\mathbb{E}[\|\boldsymbol{x}^r - \boldsymbol{x}^*\|^2] + 1/(\mu^2 r^2)\sigma_g^2 \\
&\le (1 - 2/r)L^0/r + 1/(\mu^2 r^2)\sigma_g^2 \\
&\le (1/r - 2/r^2)L^0 + L^0/r^2 \le (1/r - 1/r^2)L^0 \\
&\le \frac{L^0}{r+1}
\end{aligned}$$

Proof Steps ($c_g \neq 0$)

- Now consider the case $c_g \neq 0$
- Similarly as before, we have

$$\begin{aligned}\|\boldsymbol{x}^* - \boldsymbol{x}^{r+1}\|^2 &= \|\boldsymbol{x}^* - \boldsymbol{x}^r\|^2 + 2\langle \boldsymbol{x}^* - \boldsymbol{x}^r, \boldsymbol{x}^r - \boldsymbol{x}^{r+1}\rangle + \|\boldsymbol{x}^r - \boldsymbol{x}^{r+1}\|^2 \\ &= \|\boldsymbol{x}^* - \boldsymbol{x}^r\|^2 - 2\eta\langle \boldsymbol{x}^* - \boldsymbol{x}^r, \mathbf{g}^r\rangle + \eta^2\|\mathbf{g}^r\|^2\end{aligned}$$

- Taking an expectation, we obtain

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{x}^* - \boldsymbol{x}^{r+1}\|^2] &= \mathbb{E}[\|\boldsymbol{x}^* - \boldsymbol{x}^r\|^2] - 2\eta\mathbb{E}[\langle \boldsymbol{x}^* - \boldsymbol{x}^r, \mathbf{g}^r\rangle] + \eta^2\mathbb{E}[\|\mathbf{g}^r\|^2] \\ &= \mathbb{E}[\|\boldsymbol{x}^* - \boldsymbol{x}^r\|^2] - 2\eta\mathbb{E}[\langle \boldsymbol{x}^* - \boldsymbol{x}^r, \nabla F(\boldsymbol{x}^r)\rangle] + \eta^2\mathbb{E}[\|\mathbf{g}^r\|^2] \\ &\overset{(1.5)}{\leq} (1 - 2\mu\eta)\mathbb{E}[\|\boldsymbol{x}^* - \boldsymbol{x}^r\|^2] + \eta^2\mathbb{E}[\|\mathbf{g}^r\|^2]\end{aligned}$$

where the second equality uses the conditional expectation $\mathbb{E}[g] = \mathbb{E}[\mathbb{E}[g|x^1, \cdots, x^r]]$.

Proof Steps ($c_g \neq 0$)

- Next let us bound $\mathbb{E}[\|\mathbf{g}^r\|^2]$ as (by using the ubiaseness)

$$\begin{aligned}\mathbb{E}[\|\mathbf{g}^r\|^2] &\leq \sigma_g^2 + c_g\|\nabla F(\boldsymbol{x}^r)\|^2 \\ &= \sigma_g^2 + c_g\|\nabla F(\boldsymbol{x}^r) - \nabla F(\boldsymbol{x}^*)\|^2 \\ &\leq \sigma_g^2 + c_g L^2\|\boldsymbol{x}^r - \boldsymbol{x}^*\|^2.\end{aligned}$$

- Combining the previous two results, we have

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{x}^* - \boldsymbol{x}^{r+1}\|^2] &\leq (1-2\mu\eta)\mathbb{E}[\|\boldsymbol{x}^* - \boldsymbol{x}^r\|^2] + \eta^2(\sigma_g^2 + c_g L^2\|\boldsymbol{x}^r - \boldsymbol{x}^*\|^2) \\ &= (1-2\mu\eta-\eta^2 c_g L^2)\mathbb{E}[\|\boldsymbol{x}^* - \boldsymbol{x}^r\|^2] + \eta^2\sigma_g^2 \\ &\leq (1-\mu\eta)\mathbb{E}[\|\boldsymbol{x}^* - \boldsymbol{x}^r\|^2] + \eta^2\sigma_g^2\end{aligned}$$

  where the last step holds if we choose η small enough

Now we see the same pattern as in the previous case. The proof can be immediately completed.

Convergence Analysis (f is Non-Convex)

Theorem 1.2 (Convergence of SGD for Non-Convex f)

Under the assumption in the previous page, where f is non-convex, if $\mu_t = \mu \leq \frac{1}{Lc_g}$, then SGD achieves the following rate

$$\frac{1}{T}\sum_{r=1}^{T}[\|\nabla F(\boldsymbol{x}^r)\|^2] \leq \frac{2(F(\boldsymbol{x}^0) - \underline{F})}{T\eta} + L\eta\sigma_g^2 \tag{1.7}$$

Proof Steps

- From the descent lemma, we have

$$F(\boldsymbol{x}^{r+1}) \leq F(\boldsymbol{x}^r) + \langle \nabla F(\boldsymbol{x}^r), \boldsymbol{x}^{r+1} - \boldsymbol{x}^r \rangle + \frac{L}{2}\|\boldsymbol{x}^{r+1} - \boldsymbol{x}^r\|^2$$
$$= F(\boldsymbol{x}^r) - \eta \langle \nabla F(\boldsymbol{x}^r), \mathbf{g}^r \rangle + \frac{L\eta^2}{2}\|\mathbf{g}^r\|^2$$

- Taking expectation, and use unbiasedness we obtain

$$F(\boldsymbol{x}^{r+1}) \leq F(\boldsymbol{x}^r) - \eta \mathbb{E}[\langle \nabla F(\boldsymbol{x}^r), \mathbf{g}^r - \nabla F(x^r) \rangle] - \eta \|\nabla F(\boldsymbol{x}^r)\|^2$$
$$+ \frac{L\eta^2}{2}(\sigma_g^2 + c_g\|\nabla F(\boldsymbol{x})\|^2)$$
$$= F(\boldsymbol{x}^r) - \left(\eta - \frac{L\eta^2 c_g}{2}\right)\|\nabla F(\boldsymbol{x}^r)\|^2 + \frac{L\eta^2}{2}\sigma_g^2$$

Proof Steps

- Rearranging, and adding $r = 0, \cdots, T$

$$\left(\eta - \frac{L\eta^2 c_g}{2}\right) \frac{1}{T} \sum_{r=1}^{T} \|\nabla F(\boldsymbol{x}^r)\|^2 \leq \frac{F(\boldsymbol{x}^0) - F(\boldsymbol{x}^{T+1})}{T} + \frac{L\eta^2}{2}\sigma_g^2$$

- Let us pick η such that

$$\eta - \frac{L\eta^2 c_g}{2} \geq \eta/2, \text{ or } \eta \leq \frac{1}{Lc_g}$$

Then we obtain

$$\frac{1}{T} \sum_{r=1}^{T} \|\nabla F(\boldsymbol{x}^r)\|^2 \leq \frac{2(F(\boldsymbol{x}^0) - \underline{F})}{T\eta} + L\eta\sigma_g^2$$

Decentralized Stochastic Algorithms

Why is it important?

- Most of our previous discussion focus on the case where full local data is available every time
- So that you can take gradients $\nabla f_i(\boldsymbol{x})$ every time, or even perform $\min_{\boldsymbol{x}} f_i(\boldsymbol{x}) + \cdots$
- Is this a reasonable assumption?

Why is it important?

- In decentralized training, typically a local objective function is given by

$$f_i(\boldsymbol{x}) := \sum_{j=1}^{k} g_i^{(j)}(\boldsymbol{x}) \tag{2.1}$$

  where k is total number of data samples at each node, and $g_i^{(j)}(\boldsymbol{x})$ is the loss function evaluated at jth data sample

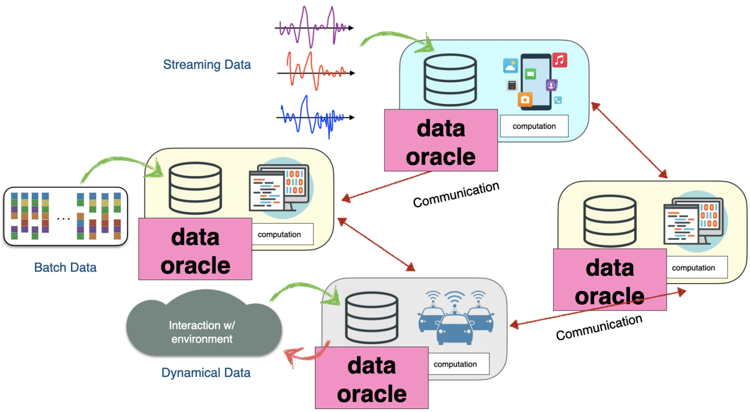
- Typically k is large, so mini-batch training implies that we have to sample the objective

Why is it important?

- Also in online optimization problem, data/observation is streaming in
- They may following certain distribution, but we do not know such distribution *a priori*
- In other problems (such as the wind-turbine problem in Lecture 1), we have to query a black box, and such queries will have noise

Why is it important?

Figure 2.1: Key Elements



Data Oracle: Describing the data acquisition process.

Problem formulation

$$
\begin{aligned}
\text{minimize}_{\boldsymbol{x}} \quad & F(\boldsymbol{x}) := \sum_{i=1}^{m} \mathbb{E}_{\xi_i}[g_i(\boldsymbol{x}, \xi_i)] := \sum_{i=1}^{m} f_i(\boldsymbol{x}) \\
\text{subject to} \quad & \boldsymbol{x} \in \mathbb{R}^n
\end{aligned}
$$

- We will define

$$
f_i(\boldsymbol{x}) := \mathbb{E}_{\xi_i}[g_i(\boldsymbol{x}, \xi_i)]
$$

- Similarly as the centralized setting, we consider the above problem
- $g_i(\boldsymbol{x}, \xi)$ can be viewed as a sample loss function
- ξ_i is the distribution of data in node i
- Can we directly extend the DGD algorithm?

The Stochastic DGD (SDGD) Algorithm

Input: $x^{(0)}$

For $r = 0, 1, \ldots, T$

Random sample ξ_i^r at each node

Calculate the stochastic gradient $\nabla_{x_i} g_i(x_i^r, \xi_i)$

Update:

$$x_i^{r+1} = \sum_{j \in \mathcal{N}_i} W_{ij} x_j^r - \alpha \times \nabla_{x_i} g_i(x_i^r, \xi_i^r), \ \forall\, i \tag{2.2}$$

or equivalently

$$\boldsymbol{x}^{r+1} = \mathbf{W}\boldsymbol{x}^r - \alpha \mathbf{d}^r, \ \text{where } \mathbf{d}^r = \{\nabla_{x_i} g_i(x_i^r, \xi_i^r)\}_{i=1}^m \tag{2.3}$$

End For

Algorithm 1: DSGD/SDGD

The SDGD Algorithm

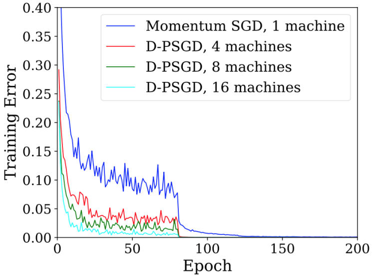
- A few variants appears in early works such as [Bianchi - Jakubowicz 13][1]
- Has received quite a lot of attention recently, due to the need to perform decentralized training [Lan et al 17] [Jiang 17] [2] [3]
- Has proven to be very useful in practice
- But requires some strong conditions for convergence

[1] Bianchi, P. and Jakubowicz, "Convergence of a Multi-Agent Projected Stochastic Gradient Algorithm for Non-Convex Optimization", IEEE TAC, 2013

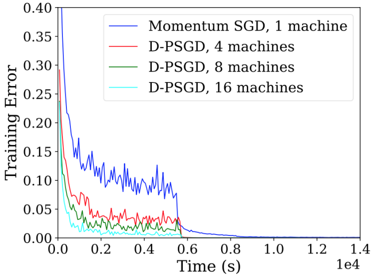
[2] X. Lian, et al, "Can decentralized algorithms outperform centralized algorithms?", in NeurIPS, 2017

[3] Z. Jianget al, "Collaborative deep learning in fixed topology networks," in NeurIPS, 2017

Decentralized Training in Practice



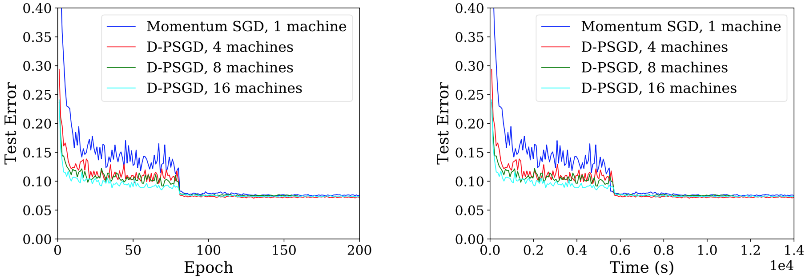
(a) Iteration vs Training Error



(b) Time vs Training Error

- ResNet-32 [He et al., 2016] on CIFAR-10 dataset
- Decrease the learning rate from 0.1 to 0.01 at epoch 80

Decentralized Training in Practice



(a) Iteration vs Test Error

(b) Time vs Test Error

- The test error after 160 epoch is 0.0715, 0.0746 and 0.0735, for 4, 8 and 16 machines, respectively.[4]
- Same level test accuracy compared to 0.0751 as reported in He et al. [2016] for centralized optimization.

[4] X. Lian, et al, "Can decentralized algorithms outperform centralized algorithms?", in NeurIPS, 2017

The D^2 Algorithm

Input: $x^{(0)}$
For $r = 0, 1, \ldots, T$
  Random sample ξ_i^r at each node
Calculate the stochastic gradient $\nabla_{x_i} g_i(x_i^r, \xi_i)$
Update:

$$x^{r+1} = 2A\boldsymbol{x}^r - A\boldsymbol{x}^{r-1} - \alpha A(\mathbf{d}^r - \mathbf{d}^{r-1}) \tag{2.4}$$

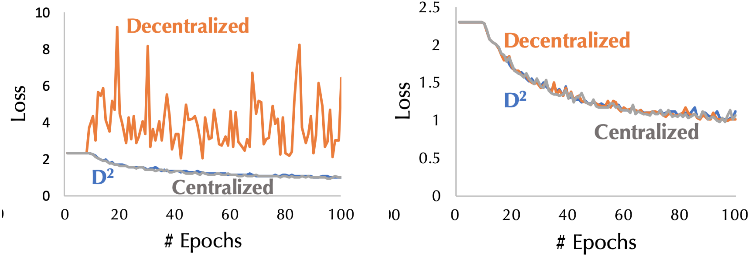
End For

Algorithm 2: D^2 Algorithm

The D^2 Algorithm

- This algorithm is related to SDGD, and also related to the gradient tracking algorithm to be introduced soon
- Reminiscent of the EXTRA algorithm, where the difference of the gradients are used
- It improves the convergence of SDGD in certain sense

D2 in Practice



- (left) heterogeneous data (right) homogeneous data
- LeNet on the CIFAR10 dataset
- comparison of DSGD, D2, and SGD.
- DSGD relies on the assumption that the data hosted on different workers are not too different

The Stochastic Gradient Push Algorithm

Input: $\boldsymbol{x}_i^{(0)} = \mathbf{z}_i^{(0)} \in \mathbb{R}^d, y_i^{(0)} = 1, \forall i$
For $r = 0, 1, \ldots, T$
    Random sample ξ_i^r at each node
Update:

$$\boldsymbol{x}_i^{r+1} = \sum_j \mathbf{W}_{ij} \left(\boldsymbol{x}_j^r - \alpha \nabla_{\mathbf{z}_j} g_i(\mathbf{z}_j^r, \xi_j^r) \right), \tag{2.5}$$

$$y_i^{r+1} = \sum_j \mathbf{W}_{ij} y_j^r, \text{(scalar PUSHSUM weight)} \tag{2.6}$$

$$\mathbf{z}_i^{r+1} = \boldsymbol{x}_i^r / y_i^r, \text{(de-biased parameter)} \tag{2.7}$$

End For

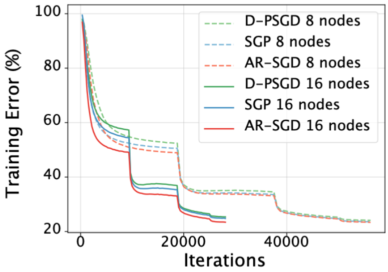
Algorithm 3: The SGP Algorithm

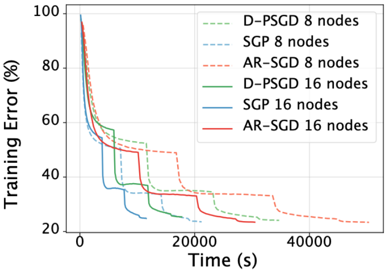
The Stochastic Gradient Push Algorithm

- same assumption as of DSGD
- equivalent to DSGD for undirected and fixed graph
- nonblocking communication [5]
- enables optimization over directed and time-varying graphs
- naturally enables asynchronous implementations

[5] AllReduce SGD: Blocks all nodes, DSGD: Blocks subsets of nodes

Stochastic Gradient Push in Practice





- ResNet-50 (He et al., 2016) on the ImageNet (Russakovsky et al., 2015)
- Iteration-wise and time-wise convergence over 10 Gbps Ethernet [6]

[6] AR-SGD: ALLREDUCE-SGD

GNSD: Gradient-tracking based Nonconvex Stochastic Decentralized Algorithm

Input: $x^{(0)}$
For $r = 0, 1, \ldots, T$
Random sample ξ_i^r at each node
Calculate the stochastic gradient $\nabla g_i(x_i^r, \xi_i)$
Update:

$$\boldsymbol{x}_i^{r+1} = \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} \boldsymbol{x}_j^r - \alpha \mathbf{y}_i^r, \tag{2.8}$$

$$\mathbf{y}_i^{r+1} = \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} \mathbf{y}_j^r + \nabla_{\boldsymbol{x}_i} g_i(\boldsymbol{x}_i^{r+1}, \xi_i^{r+1}) - \nabla_{\boldsymbol{x}_i} g_i(\boldsymbol{x}_i^r, \xi_i^r), \tag{2.9}$$

End For

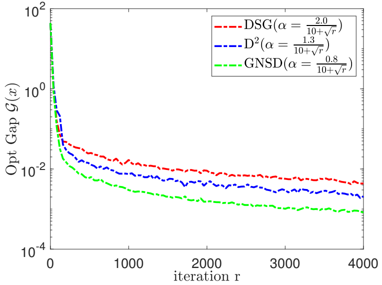
Algorithm 4: GNSD

GNSD in Theory

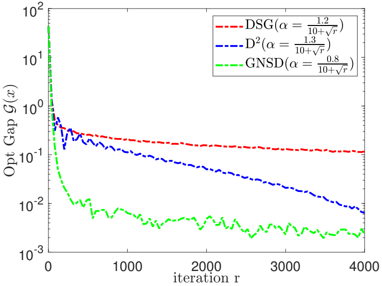
- Unified decentralized framework
- Support arbitrary heterogeneous data
- Support arbitrary doubly stochastic matrix
  - Condition on $\mathbf{W}$ is more relaxed (on the weight matrix)

$$\text{GNSD}: -1 < \lambda(W) < 1$$
$$\text{D}^2: -1/3 < \lambda(W) < 1$$

GNSD in Practice



(a) Identically distributed



(b) Non-identically distributed

Figure 2.2: Training CNN model on MNIST dataset, with different data distribution.

Implementation

Communication Implementations

Before training:

- Initialize the communication network
- 1: identify the agents
- 2: initialize the links (i.e., neighbours' address, id, communication schedule)
- 3: initialize communication threads (or subprocess) and buffers

During training:

- Prepare the data (e.g., vectorize the model, compress the gradient, add noises)
- Communicate with the neighbours
- Process the collected data (e.g., taking mean or median, reform model from vetor)

Communication Implementations

- Machine learning platform:
  - Tensorflow
  - Pytorch
  - PaddlePaddle
  - Microsoft Cognitive Toolkit, Keras, etc.
- Communication backend: gRPC, MPI, ZMQ, PySyft, NCCL, etc.
- Network topology: ring, star, grid, hierarchical, etc.
- Communication type: synchronized or asynchronized, P2P or collective

Example Code

Code example using MPI and Tensorflow:
Import modules and define the function:

```
import numpy as np
from mpi4py import MPI
def comm_fn(var, comm, graph, weight, isvar=True):
    ...
```

Example Code

Iterate over the list of data that need to communicate:

```
...
def comm_fn(var, comm, graph, weight, isvar=True):
    list1=[]
    list2=[]
    for v in var:
        # 1. Pre-process the data
        # 2. Communicate with neighbour (send/receive)
        # 3. Post-process the received data
    ...
```

Communication Implementations

Pre-process the data (vectorize the data):

```
...
for v in var:
    x=v.numpy()
    x=np.reshape(x,-1)                # Vectorize the model
    buff=np.tile(x,(graph.node,1)) # Create data buffer
    ...
...
```

Communication Implementations

Point-to-point asynchronous communication using Isend and Recv:

```
...
for v in var:
    ...
    for i in range(graph.node): # Send data
        if (weight[i]>0) and not (i==graph.id):
            req=comm.Isend(x, dest=i, tag=graph.id)
            list2.append(req)
    for i in range(graph.node): # Receive data
        if (weight[i]>0) and not (i==graph.id):
            comm.Recv(buff[i], source=i, tag=i)
    while len(list2)>0:          # Synchronization
        req=list2.pop()
        req.wait()
    ...
...
```

Communication Implementations

Post-process the data (weighted average and reshape):

```
...
for v in var:
    ...
    t = np.dot(weight, buff) # Weighted average
    t = np.reshape(t, np.shape(v)) # Reshape
    if (isvar):
        v.assign(t)
    else:
        list1.append(t)
...
```

Algorithm Implementations

- A useful reference code for the Stochastic Gradient Push Algorithm [7]

[7] `https://github.com/facebookresearch/stochastic_gradient_push`

Numerical Results

[discussion using the survey paper.]

Theoretical Analysis

Convergence Analysis

- We plan to select the GNSD algorithm to analyze
- For the proof of SGSD, similar to non-convex DGD + SGD (try yourself)
- For GNSD, the analysis is useful because, it reduces to the deterministic gradient tracking analysis

Assumptions for both SDGD and GNSD

- Lipschitz smoothness

$$\|\nabla_{\boldsymbol{x}_i} f_i(\boldsymbol{x}_i) - \nabla_{\boldsymbol{x}_i'} f_i(\boldsymbol{x}_i')\| \leq L\|\boldsymbol{x}_i - \boldsymbol{x}_i'\|, \forall i \tag{4.1}$$

- Unbiased stochastic gradient,

$$\mathbb{E}_{\xi_i}[\nabla_{\boldsymbol{x}_i} g_i(\boldsymbol{x}_i, \xi_i)] = \nabla f_i(\boldsymbol{x}_i), \forall i \tag{4.2}$$

- Bounded gradient variance,

$$\mathbb{E}_{\xi_i}\|\nabla_{\boldsymbol{x}_i} g_i(\boldsymbol{x}_i, \xi_i) - \nabla f_i(\boldsymbol{x}_i)\|^2 \leq \sigma^2, \forall i \tag{4.3}$$

- Doubly stochastic mixing matrix $W \in \mathbb{R}^{n \times n}$:

$$|\underline{\lambda}_{\max}(W)| := \eta < 1, \quad A\mathbf{1} = \mathbf{1}. \tag{4.4}$$

where $\underline{\lambda}_{\max}(W)$ denotes the second largest eigenvalue of W.

Key properties

- Contraction Property (if W satisfies the assumption above):

$$\begin{aligned}\|W\boldsymbol{x}^r - \mathbf{1}\bar{x}^r\| &= \|W(\boldsymbol{x}^r - \mathbf{1}\bar{x}^r)\| \\ &\leq \underline{\lambda}_{\max}(W)\|\boldsymbol{x}^r - \mathbf{1}\bar{x}^r\| \\ &\leq \mu\|\boldsymbol{x}^r - \mathbf{1}\bar{x}^r\|^2, \mu \in (0,1)\end{aligned}$$

- Contraction of iterates

$$\mathbb{E}\|\boldsymbol{x}^{r+1} - \mathbf{1}\bar{x}^{r+1}\|^2 \leq \mu\mathbb{E}\|\boldsymbol{x}^r - \mathbf{1}\bar{x}^r\|^2 + \alpha^2 C_1,$$

where C_1 is some constant, α is the stepsize and $\bar{x}^r$ denotes the average of x^r over all nodes.

Proof of GNSD: Definitions

- Virtual sequence: $\{\underline{\mathbf{y}}^r\}$, to characterize the updates by using the true gradients as the counterpart of (2.9).

$$\underline{\mathbf{y}}^{r+1} := W\underline{\mathbf{y}}^r + \nabla_{\boldsymbol{x}} F(\boldsymbol{x}^{r+1}) - \nabla_{\boldsymbol{x}} F(\boldsymbol{x}^r), \forall r \geq 1 \tag{4.5}$$

where $\underline{\mathbf{y}}^1 := \nabla F(\boldsymbol{x}^1)$.

- Average sequence:

$$\bar{g}(\boldsymbol{x}^r) = \frac{1}{m}\sum_{i=1}^{m} \nabla g_i(\boldsymbol{x}^r, \xi^r)$$

$$\bar{\boldsymbol{x}}^r := \frac{1}{m}\mathbf{1}^T\boldsymbol{x}^r$$

$$\bar{\mathbf{y}}^r := \frac{1}{m}\mathbf{1}^T\mathbf{y}^r$$

$$\bar{\underline{\mathbf{y}}}^r := \frac{1}{m}\mathbf{1}^T\underline{\mathbf{y}}^r = \frac{1}{m}\sum_{i=1}^{m} \nabla_{\boldsymbol{x}} f_i(\boldsymbol{x}^r), \quad [why?]$$

Proof of GNSD: Average Iterates

- Average iterates

$$
\begin{aligned}
\bar{\boldsymbol{x}}^{r+1} &= \bar{\boldsymbol{x}}^r - \frac{\alpha}{m}\mathbf{1}^T\mathbf{y}^r = \bar{\boldsymbol{x}}^r - \frac{\alpha}{n}\mathbf{1}^T(\mathbf{y}^r - \mathbf{1}\underline{\bar{\mathbf{y}}}^r + \mathbf{1}\underline{\bar{\mathbf{y}}}^r) \\
&= \bar{\boldsymbol{x}}^r - \alpha\underline{\bar{\mathbf{y}}}^r - \frac{\alpha}{m}\mathbf{1}^T(\mathbf{y}^r - \mathbf{1}\underline{\bar{\mathbf{y}}}^r). \qquad (4.6) \\
\bar{\mathbf{y}}^{r+1} &= \bar{\mathbf{y}}^r + \bar{g}(\boldsymbol{x}^{r+1}) - \bar{g}(\boldsymbol{x}^r), \qquad (4.7)
\end{aligned}
$$

- We have the following

$$
\begin{aligned}
\bar{\boldsymbol{x}}^{r+1} &= \bar{\boldsymbol{x}}^r - \frac{\alpha}{n}\mathbf{1}^T\mathbf{y}^r = \bar{\boldsymbol{x}}^r - \frac{\alpha}{n}\mathbf{1}^T(\mathbf{y}^r - \mathbf{1}\underline{\bar{\mathbf{y}}}^r + \mathbf{1}\underline{\bar{\mathbf{y}}}^r) \qquad (4.8) \\
&= \bar{\boldsymbol{x}}^r - \alpha\underline{\bar{\mathbf{y}}}^r - \frac{\alpha}{n}\mathbf{1}^T(\mathbf{y}^r - \mathbf{1}\underline{\bar{\mathbf{y}}}^r). \qquad (4.9)
\end{aligned}
$$

Proof of GNSD: Bound iterates with deterministic counterpart

Lemma 4.1

(Bounded Variance) The iterates $\{\mathbf{y}^r\}$ are generated by GNSD. Under assumptions, we have

$$\mathbb{E}\|\mathbf{y}^r - \underline{\mathbf{y}}^r\|^2 \leq \kappa\sigma^2, \tag{4.10}$$

where $\kappa := (1+\widetilde{\eta}/(1-\eta))^2 m^2$ and $\|\mathbf{W} - \mathbf{I}\| := \widetilde{\eta}$.

Proof of GNSD: Descent on Objective

Lemma 4.2

(Descent Lemma) Assume the sequence $(\boldsymbol{x}^r, \mathbf{y}^r)$ *is generated by GNSD. We have*

$$\mathbb{E}\left[f(\bar{\boldsymbol{x}}^{r+1})\right] \leq \mathbb{E}\left[f(\bar{\boldsymbol{x}}^r)\right] - \left(\alpha - \left(\frac{\alpha\beta}{2} + \alpha^2 L\right)\right)\mathbb{E}\|\underline{\bar{\mathbf{y}}}^r\|^2$$
$$+ \frac{\alpha}{2\beta}\frac{L^2}{m}\mathbb{E}\|\boldsymbol{x}^r - \mathbf{1}\bar{\boldsymbol{x}}^r\|^2 + \frac{\alpha^2 L\sigma^2}{m},$$

where β *is some constant.*

Proof Steps

- Lipschitz continuity
  $f(\bar{\boldsymbol{x}}^{r+1}) \le f(\bar{\boldsymbol{x}}^r) + \langle \nabla f(\bar{\boldsymbol{x}}^r), \bar{\boldsymbol{x}}^{r+1} - \bar{\boldsymbol{x}}^r \rangle + \frac{L}{2}\|\bar{\boldsymbol{x}}^{r+1} - \bar{\boldsymbol{x}}^r\|^2$
- definition of average iterates (4.8)
- Cauchy-Schwarz inequality

$$
\begin{aligned}
f(\bar{\boldsymbol{x}}^{r+1}) \le & f(\bar{\boldsymbol{x}}^r) + \frac{\alpha}{2\beta}\|\nabla f(\bar{\boldsymbol{x}}^r) - \underline{\bar{\mathbf{y}}}^r\|^2 + \frac{\alpha\beta}{2}\|\underline{\bar{\mathbf{y}}}^r\|^2 - \alpha\|\underline{\bar{\mathbf{y}}}^r\|^2 \\
& - \alpha\langle \nabla f(\bar{\boldsymbol{x}}^r), \frac{1}{n}\mathbf{1}^T(\mathbf{y}^r - \underline{\mathbf{y}}^r)\rangle \\
& - \alpha\langle \nabla f(\bar{\boldsymbol{x}}^r), \frac{1}{n}\mathbf{1}^T(\underline{\mathbf{y}}^r - \mathbf{1}\underline{\bar{\mathbf{y}}}^r)\rangle \\
& + \alpha^2 L\|\underline{\bar{\mathbf{y}}}^r\|^2 + \alpha^2 L\|\bar{\mathbf{y}}^r - \mathbf{1}\underline{\bar{\mathbf{y}}}^r\|^2
\end{aligned}
$$

- $\mathbf{1}^T(\underline{\mathbf{y}}^r - \mathbf{1}\underline{\bar{\mathbf{y}}}^r) = 0$
- unbiasedness assumption

$$
\mathbb{E}_{\mathcal{F}^{r+1}}[\langle \nabla f(\bar{\boldsymbol{x}}^r), \frac{1}{n}\mathbf{1}^T(\mathbf{y}^r - \underline{\mathbf{y}}^r)\rangle | \mathcal{F}^r] = 0.
$$

Proof Steps

Take expectation on both sides and considering

- $\mathbb{E}\|\bar{\mathbf{y}}^r - \underline{\bar{\mathbf{y}}}^r\|^2 \leq \sigma^2/n$
- $\mathbb{E}\|\nabla f(\bar{\boldsymbol{x}}^r) - \underline{\bar{\mathbf{y}}}^r\|^2 \leq \frac{1}{n}\mathbb{E}\|\boldsymbol{x}^r - \mathbf{1}\bar{\boldsymbol{x}}^r\|^2$

we have

$$\begin{aligned}
\mathbb{E}\left[f(\bar{\boldsymbol{x}}^{r+1})\right] &\leq \mathbb{E}\left[f(\bar{\boldsymbol{x}}^r)\right] + \frac{\alpha}{2\beta}\mathbb{E}\|\nabla f(\bar{\boldsymbol{x}}^r) - \underline{\bar{\mathbf{y}}}^r\|^2 + \frac{\alpha\beta}{2}\mathbb{E}\|\underline{\bar{\mathbf{y}}}^r\|^2 \\
&- \alpha\mathbb{E}\|\underline{\bar{\mathbf{y}}}^r\|^2 + \alpha^2 L\mathbb{E}\|\underline{\bar{\mathbf{y}}}^r\|^2 + \frac{\alpha^2 L\sigma^2}{n} \\
&\leq \mathbb{E}\left[f(\bar{\boldsymbol{x}}^r)\right] + \left(-\alpha + \frac{\alpha\beta}{2} + \alpha^2 L\right)\mathbb{E}\|\underline{\bar{\mathbf{y}}}^r\|^2 + \frac{\alpha}{2\beta}\frac{L^2}{n}\mathbb{E}\|\boldsymbol{x}^r - \mathbf{1}\bar{\boldsymbol{x}}^r\|^2 \\
&+ \frac{\alpha^2 L\sigma^2}{n},
\end{aligned}$$

Proof of GNSD: Descent on the Averages

Lemma 4.3

(Iterates Contraction) Using the assumption of $\mathbf{W}$, *we have following contraction property of iterates generated by GNSD:*

$$\mathbb{E}\|\boldsymbol{x}^{r+1} - \mathbf{1}\bar{\boldsymbol{x}}^{r+1}\|^2 \leq (1+\beta)\eta^2\mathbb{E}\|\boldsymbol{x}^r - \mathbf{1}\bar{\boldsymbol{x}}^r\|^2$$
$$+ 3(1+\frac{1}{\beta})\alpha^2\mathbb{E}\|\underline{\mathbf{y}}^r - \mathbf{1}\underline{\bar{\mathbf{y}}}^r\|^2 + 6(1+\frac{1}{\beta})\alpha^2\kappa\sigma^2$$
$$\mathbb{E}\|\mathbf{y}^{r+1} - \mathbf{1}\bar{\mathbf{y}}^{r+1}\|^2 \leq 4nL^2\alpha^2(1+\frac{1}{\beta})^2\|\underline{\bar{\mathbf{y}}}^r\|^2$$
$$+ \left(L^2\eta^2(1+\beta)(1+\frac{1}{\beta}) + 4L^2(1+\frac{1}{\beta})^2\right)\mathbb{E}\|\boldsymbol{x}^r - \mathbf{1}\bar{\boldsymbol{x}}^r\|^2$$
$$+ \left((1+\beta)\eta^2 + 4L^2\alpha^2(1+\frac{1}{\beta})^2\right)\mathbb{E}\|\underline{\mathbf{y}}^r - \mathbf{1}\underline{\bar{\mathbf{y}}}^r\|^2$$
$$+ 4L^2\alpha^2(1+\frac{1}{\beta})^2\kappa\sigma^2$$

where β *is some constant such that* $(1+\beta)\eta^2 < 1$ *and* $\|\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T\| \leq 1$.

Proof Steps

- contraction property of the iterates, i.e.,

$$\|\mathbf{W}\boldsymbol{x}^r - \mathbf{1}\bar{\boldsymbol{x}}^r\| = \|\mathbf{W}(\boldsymbol{x}^r - \mathbf{1}\bar{\boldsymbol{x}}^r)\| \leq \eta\|\boldsymbol{x}^r - \mathbf{1}\bar{\boldsymbol{x}}^r\| \tag{4.11}$$

  where the inequality comes from $\mathbf{1}^T(\boldsymbol{x}^r - \mathbf{1}\bar{\boldsymbol{x}}^r) = 0$
- definition of (2.8) and the Cauchy-Schwartz inequality

$$\begin{aligned}
&\|\boldsymbol{x}^{r+1} - \mathbf{1}\bar{\boldsymbol{x}}^{r+1}\|^2 = \|\mathbf{W}\boldsymbol{x}^r - \alpha\mathbf{y}^r - \mathbf{1}(\bar{\boldsymbol{x}}^r - \alpha\bar{\mathbf{y}}^r)\|^2 \\
&\leq (1+\beta)\|\mathbf{W}\boldsymbol{x}^r - \mathbf{1}\bar{\boldsymbol{x}}^r\|^2 + (1+\frac{1}{\beta})\alpha^2\|\mathbf{y}^r - \mathbf{1}\bar{\mathbf{y}}^r\|^2 \\
&\leq (1+\beta)\eta^2\|\boldsymbol{x}^r - \mathbf{1}\bar{\boldsymbol{x}}^r\|^2 + 3(1+\frac{1}{\beta})\alpha^2\|\mathbf{y}^r - \underline{\mathbf{y}}^r\|^2 \\
&+ 3(1+\frac{1}{\beta})\alpha^2\|\underline{\mathbf{y}}^r - \mathbf{1}\bar{\underline{\mathbf{y}}}^r\|^2 + 3(1+\frac{1}{\beta})\alpha^2\|\mathbf{1}\bar{\mathbf{y}}^r - \mathbf{1}\bar{\underline{\mathbf{y}}}^r\|^2
\end{aligned}$$

- similar for $\mathbf{y}$

Proof of GNSD: Descent on the Potential

Lemma 4.4

(Potential Function) Constructing the potential function

$$P(\boldsymbol{x}^r) := \mathbb{E}\left[f(\bar{\boldsymbol{x}}^r)\right] + \frac{L^2\alpha}{2\beta^2\eta^2}\mathbb{E}\|\boldsymbol{x}^r - \mathbf{1}\bar{\boldsymbol{x}}^r\|^2 + \alpha^2\mathbb{E}\|\underline{\mathbf{y}}^r - \mathbf{1}\bar{\underline{\mathbf{y}}}^r\|^2,$$

then we have

$$P(\boldsymbol{x}^{r+1}) - P(\boldsymbol{x}^r) \leq -C_1\alpha\mathbb{E}\|\bar{\underline{\mathbf{y}}}^r\|^2 - \frac{L^2\alpha}{2\beta^2\eta^2}C_2\mathbb{E}\|\boldsymbol{x}^r - \mathbf{1}\bar{\boldsymbol{x}}^r\|^2$$
$$- \alpha^2 C_3\mathbb{E}\|\underline{\mathbf{y}}^r - \mathbf{1}\bar{\underline{\mathbf{y}}}^r\|^2 + \frac{\alpha^2 L\sigma^2}{n} + C_4 L^2\alpha^3\kappa n^2\sigma^2, \quad (4.12)$$

where C_1, C_2, C_3, C_4 are constants.

Proof of GNSD: Convergence to Mean

Theorem 4.5

If we pick $\alpha \sim \mathcal{O}(\frac{1}{\sqrt{T/n}})$, *then we have*

$$\frac{1}{T}\sum_{r=1}^{T}\mathbb{E}\|\underline{\bar{\mathbf{y}}}^r\|^2 + \mathbb{E}\|\boldsymbol{x}^r - \mathbf{1}\bar{\boldsymbol{x}}^r\|^2 \leq \mathcal{O}\left(\frac{\sigma^2}{\sqrt{nT}}\right)$$

where T *is large.*

- Global gradient + global consensus error diminish together!
- Algorithm convergence rate: $\mathcal{O}(\frac{1}{\sqrt{T}})$
- Linear speed up

Discussion: Convergence Conditions

- Among the algorithms that have been introduced, DSGD requires the strongest assumption, that is

$$\|\nabla f_i(\boldsymbol{x}) - F(\boldsymbol{x})\|^2 \leq \theta < \infty, \ \forall\ i \tag{4.13}$$

- This suggests that local functions are the global function are similar, or related in some sense

- For example, if

$$F(\boldsymbol{x}) := \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}[g(\boldsymbol{x}, \{\mathbf{a}_i, b_i\})]$$

and if the local data $\{\mathbf{a}_i, b_i\}$'s are distributed with the same distribution as $\{\mathbf{a}_i, b_i\}$'s, then the above condition will satisfy

Discussions: Convergence Conditions

- D^2 and GNSD do not require such a condition. However, D^2 requires certain additional assumptions on the weight matrix W
- Please see a recent survey for more detailed discussion [Chang et al][8]

[8] T.-H. Chang and M. Hong and H.-T. Wai and X. Zhang and S. Lu, "Distributed Learning in the Non-Convex World: From Batch to Streaming Data, and Beyond", IEEE Signal Processing Magazine, 2020

More Recent Research Development

- Decentralized optimization for stochastic problems have continued to receive strong research interests
- Recent topics of interest include
  - Communication efficient optimization (reduce communication burdens, faster overall training time)
  - Optimal methods (smallest communication and computation complexity to achieve certain error)
  - For more challenging problems such as mini-max problems (with applications in GAN)
  - $\cdots$
- These will be discussed in our last lecture