STOCHASTIC GRADIENT PUSH FOR DISTRIBUTED DEEP LEARNING

International Conference on Machine Learning '19

facebook Artificial Intelligence Research

McGill

Mahmoud Assran
Nicolas Loizou
Nicolas Ballas
Mike Rabbat

Introduction

When training over high-latency networks, use asynchronous gossip-based aggregation instead of *AllReduce*

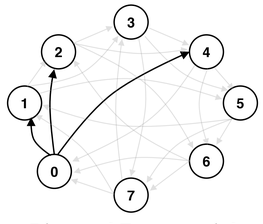
- Exactly averaging gradients across nodes in data-parallel training can be slow in high-latency networks
- ***PushSum***, proposed in control systems literature, is consensus-based alg. for iteratively aggregating information across nodes
- We study ***Stochastic Gradient Push* (*SGP*)**, blend of ***SGD* & *PushSum*,** for asynchronous distributed training in bandwidth-limited / high-latency networks

PushSum Background

Iterative averaging over directed time-varying graphs
(Kempe et al., IEEE Symposium on Foundations of Computer Science, 2003.)

- Our goal is to compute $\frac{1}{n}\sum_{i=1}^{n} y_i^{(0)}$, where $y_i^{(0)} \in \mathbb{R}^d$ is a variable at the i^{th} node
- Algorithm is iterative. Each node $i \in \{1,2,\dots,n\}$, is initialized with the variable to be averaged, $y_i^{(0)} \in \mathbb{R}^d$, and a scalar push-sum weight $w_i^{(0)} = 1$
- **Algorithm in Matrix Form:** $\mathbf{Y}^{(0)} = [y_i^{(0)}]_{i=1}^n, \quad \mathbf{w}^{(0)} = [w_i^{(0)}]_{i=1}^n$
  $\mathbf{P}_{j,i}^{(t)} \geq 0, \quad \mathbf{1}^T\mathbf{P}^{(t)} = \mathbf{1}$ ← *structure of P defines graph topology*
  Iteratively Compute: $\mathbf{Y}^{(t+1)} = \mathbf{P}^{(t)}\mathbf{Y}^{(t)}, \qquad \mathbf{w}^{(t+1)} = \mathbf{P}^{(t)}\mathbf{w}^{(t)}$

$$\lim_{K\to\infty} \Pi_{t=0}^{K}\mathbf{P}^{(t)} = \pi\mathbf{1}^T \implies \begin{matrix} y_i^{(\infty)} = \pi_i \sum_{j=1}^{n} y_j^{(0)} \\ w_i^{(\infty)} = \pi_i n \end{matrix} \implies \frac{y_i^{(\infty)}}{w_i^{(\infty)}} = \frac{1}{n}\sum_{j=1}^{n} y_j^{(0)}$$



Directed Exponential Graph

- Run the PushSum algorithm over directed and potentially time-varying communication topologies
- We use a Directed Exponential Graph topology for our experiments

Blending *SGD* and *PushSum*

Distributed optimization using Stochastic Gradient Push

Key Idea: Distributed optimization over directed time-varying graphs naturally enables asynchronous communication

Initialize: model parameters, $x_i^{(0)} \in \mathbb{R}^d$, de-biased estimate of model parameters, $z_i^{(0)} = x_i^{(0)}$, and push-sum weight $w_i^{(0)} = 1$

Each node repeat:

1. $x_i^{(k+1/2)} = x_i^{(k)} - \gamma \nabla F_i(z_i^{(k)}; \xi_i^{(k)})$ ← *local SGD step*
2. If no messages received in last τ iterations, block and wait to receive message from peers ← *algorithmically bound message staleness*
3. $x_i^{(k+1)}, w_i^{(k+1)} \leftarrow \textbf{PushSum}(x_i^{(k+1/2)}, w_i^{(k)})$ ← *1-iteration of nonblocking gossip*
4. $z_i^{(k+1)} = x_i^{(k+1)}/w_i^{(k+1)}$ ← *de-bias estimate of average model*

Stochastic Gradient Push (Nedic et al., IEEE TAC, 2016)	$\tau = 0$
τ-Overlap Stochastic Gradient Push	$\tau > 0$

Analysis

Convergence rate for smooth non-convex loss functions

n nodes cooperate to solve

$$\min_{x_{i\in[n]}} \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\xi_i\sim D_i} F_i(x_i;\xi_i)$$
$$\text{subject to} \quad x_i = x_j, \quad (i,j \in [n])$$

Define $f_i(x_i) = \mathbb{E}_{\xi_i\sim D_i} F_i(x_i;\xi_i)$, *and* $f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(x_i)$

Assume that $\tau < \infty$, $f_i(\cdot)$ *is L-smooth, and*

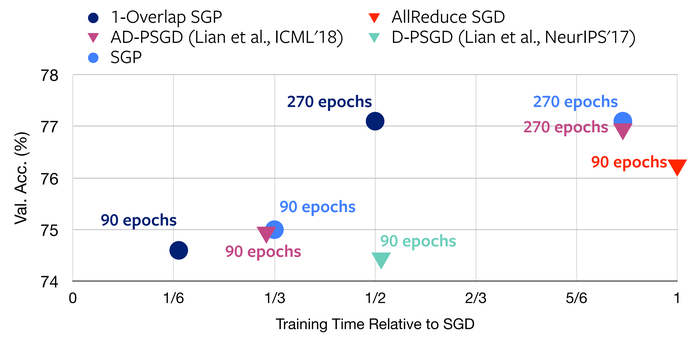
$$\mathbb{E}_{\xi\sim D_i} \left\| \nabla F_i(x_i;\xi) - \nabla f_i(x_i) \right\| \leq \sigma^2$$
$$\frac{1}{n}\sum_{i=1}^{n} \left\| \nabla f_i(x_i) - \nabla f(x) \right\|^2 \leq \zeta^2$$

Main Convergence Result:

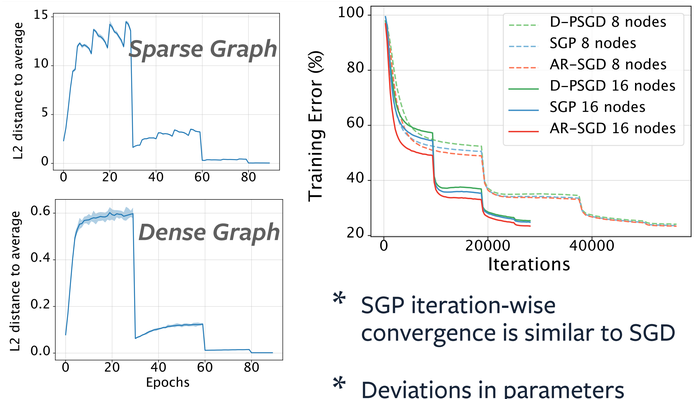
$$\frac{1}{nK}\sum_{k=0}^{K-1}\sum_{i=1}^{n} \mathbb{E} \left\| \nabla f(z_i^{(k)}) \right\|^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{nK}}\right)$$

Empirical Evaluation

ImageNet, ResNet-50



32 nodes (256 V100 GPUs) interconnected via 10 Gbps Ethernet



L2 Distance of parameters form average throughout training

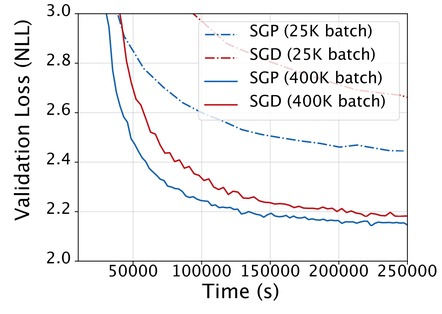
* SGP iteration-wise convergence is similar to SGD

* Deviations in parameters related to graph connectivity & learning-rate

WMT'16 En-De, Tranformer

BLEU-Score:
SGP 27.5; SGD 26.9

* SGP can train better models than AllReduce-SGD in less time



Resources

Code:
github.com/facebookresearch/stochastic_gradient_push

Applications in Deep Reinforcement Learning:
M. Assran, J. Romoff, N. Ballas, J. Pineau, M. Rabbat, Gossip-based Actor-Learner Architectures for Deep Reinforcement Learning, preprint, 2019