DRACO: Robust Distributed Training against Adversaries

Lingjiao Chen, Hongyi Wang, Zachary Charles, Dimitris Papailiopoulos

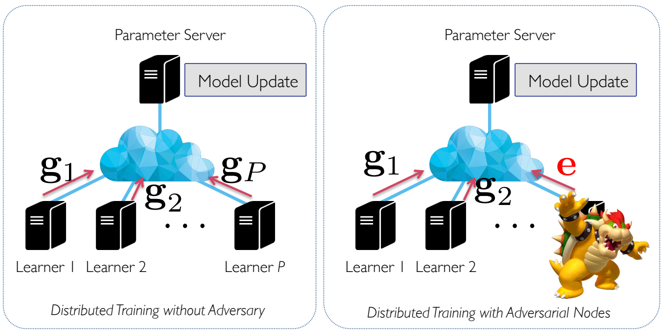
University of Wisconsin-Madison

{lchen, hongyiwang}@cs.wisc.edu, zcharles@math.wisc.edu, dimitris@papail.io

Introduction

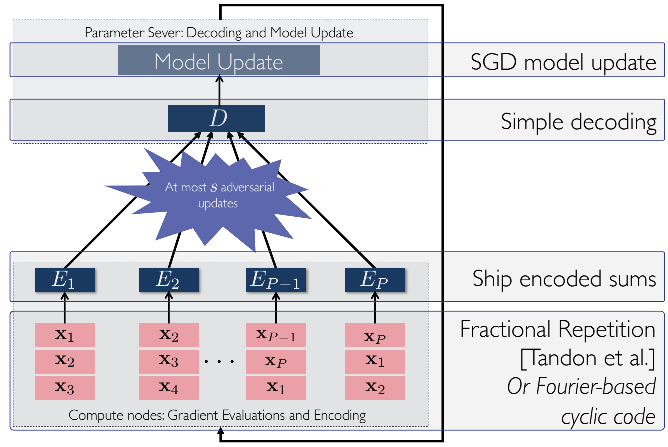
Challenge: robustness of distributed optimization algorithms
- Distributed Training **vulnerable to attacks**
- Vanilla SGD is **not robust against a single adversary**

Goal: build a **robust** version of SGD that is:
- Computational **cheap**
- **Black-box** convergence guarantee



Key Idea:
- Defend via **algorithmic redundancy**
- Borrow tools from **coding theory**

DRACO: Robust SGD via Coding Theory



Pros:
- Small PS complexity
- DRACO model = SGD model
- Convergence for free

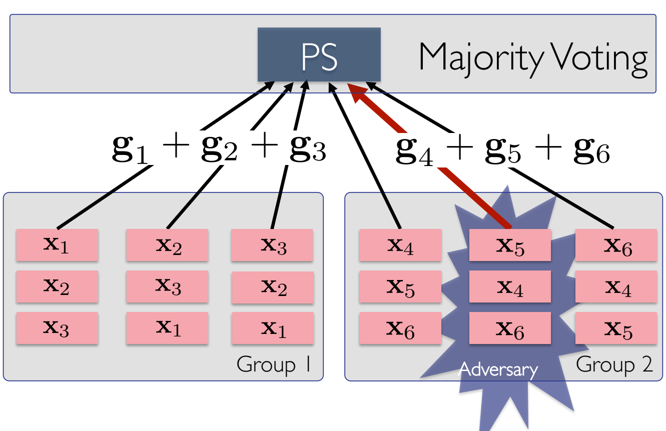
Cons:
- Redundancy seems wasteful

Thm1:
To tolerate s adversaries, we need r > 2s

Thm2:
DRACO is optimal

Concept

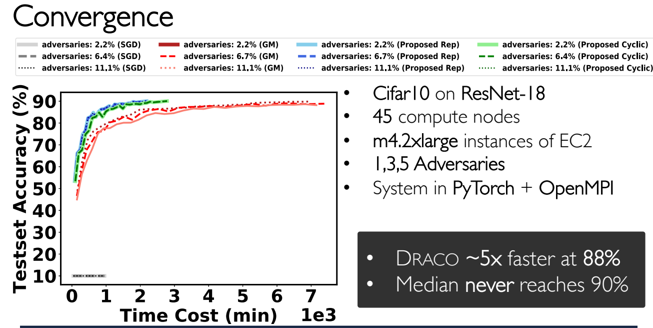
Defend via Majority



- Each group computes the **same** sum of gradients
- PS uses **majority** to select **true** sum of **gradients**
- if **fewer than half** of nodes/ group are adversarial,
  => **majority returns true gradient**

Experiments

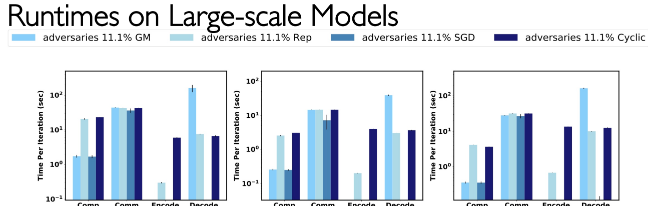
Convergence



- **Cifar10** on **ResNet-18**
- **45** compute nodes
- **m4.2xlarge** instances of EC2
- **1,3,5** Adversaries
- System in **PyTorch** + **OpenMPI**

- DRACO **~5x** faster at **88%**
- Median **never** reaches 90%

Runtimes on Large-scale Models



(a) ResNet-152, Rev Grad (b) VGG-19, Rev Grad (c) AlexNet, Rev Grad

- DRACO **~5x** slower during compute
- Median **~20x** slower during averaging

Acknowledgement: The experimental part of this work was supported in part by AWS EC2 credits from Amazon. We thank Jeffrey Naughton and Remzi Arpaci-Dusseau for invaluable discussions