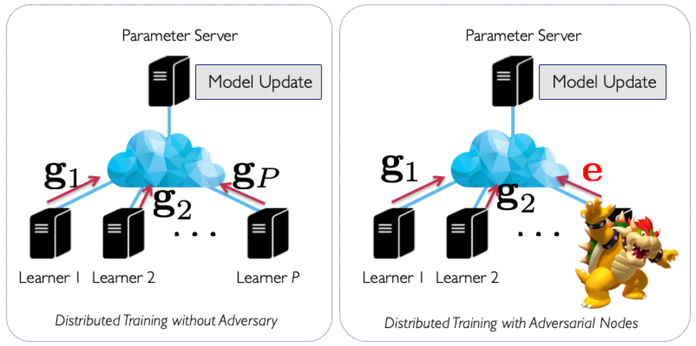
DETOX: A Redundancy-based Framework for Faster and More Robust Gradient Aggregation

Shashank Rajput, Hongyi Wang*, Zachary Charles, Dimitris Papailiopoulos*
University of Wisconsin-Madison
rajput3@wisc.edu, hongyiwang@cs.wisc.edu, zcharles@wisc.edu, dimitris@papail.io

Introduction

Challenge: Byzantine-resilience of distributed SGD

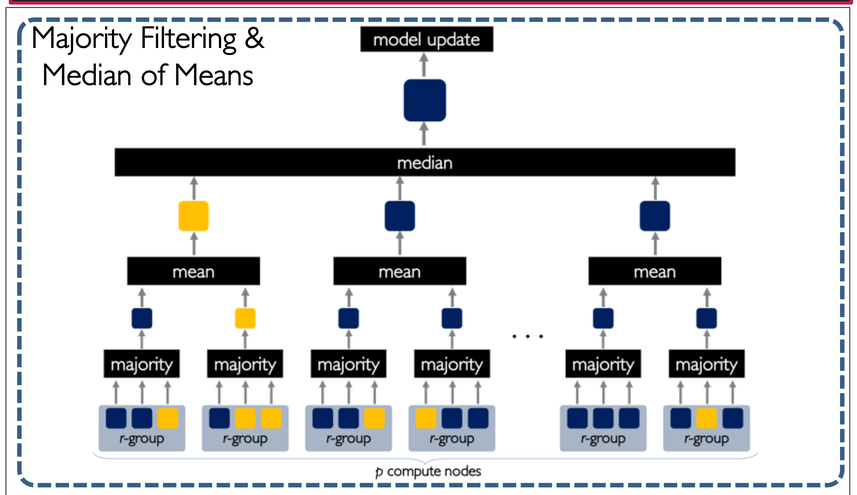
- Distributed Training vulnerable to Byzantine system failures
- Vanilla SGD fails to converge under a single Byzantine-error
- Two major approaches for defense:
  - Robust aggregation (computationally slow)
  - "Large group" majority voting (scales badly)



Key Idea:

- Filter Byzantine machines via minimal **algorithmic redundancy**
- Improve robustness and speed using **median of means**

DETOX



Step I: Majority filtering

- Form small groups, each group computes the **same** gradients.
- Most of the Byzantine gradients lose majority – filtered out.

Step II: Median of means approach to robustness

- Gives good robustness guarantees and low compute complexity.
- The "median" can be replaced by any other robust aggregator.

DETOX Guarantees

Theorem 1: Majority voting needs only logarithmic redundancy to reduce the effective number Byzantine workers to a constant.

- Only exponentially few Byzantine gradients survive majority filtering
- High probability bound!

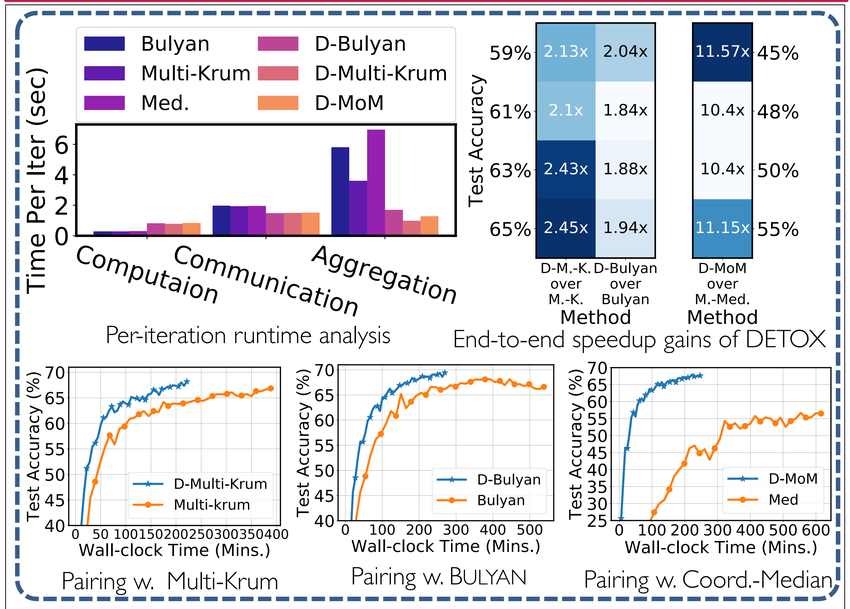
Theorem 2: 'Median' of means after majority voting achieves same error as that with uncorrupted gradients. OPTIMAL!

- 'Median' = Geometric Median, Coordinate-wise median or Trimmed mean
- Virtually no dependence on fraction or number of Byzantine machines

Computational complexity:

- Only logarithmic increase in computation for compute nodes.
- Aggregation speed at parameter server is now linear!

Experiments: Scalability



Experimental Setup:

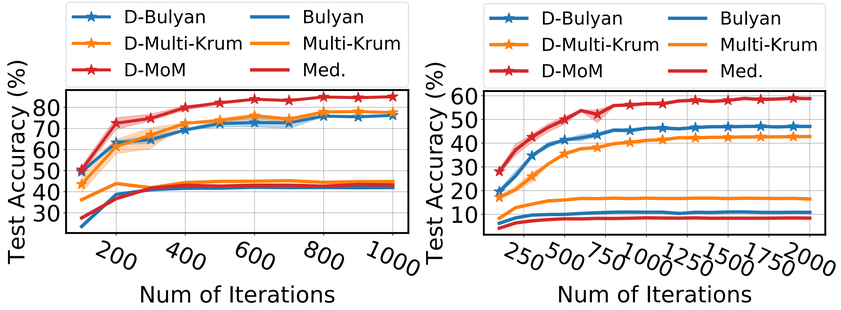
- 46 m5.2xlarge instances on AWS EC2
- VGG-13-BN on CIFAR-100; ResNet18 on CIFAR-10
- 5 out of 45 compute nodes deploying *reverse gradient* attack

Main observation:

Applying DETOX leads to significant **speedups**. Up to **an order of magnitude** end-to-end speedup is observed.

Experiments: Byzantine-resilience

Experimental study I: Defending the ALIE Byzantine attack [1]



ResNet-18 trained on CIFAR-10 | VGG-13BN trained on CIFAR-100

Methods	ResNet-18	VGG13-BN
D-Multi-krum	80.3%	42.98%
D-Bulyan	76.8%	46.82%
D-Med.	**86.21%**	**59.51%**
Multi-krum	45.24%	17.18%
Bulyan	42.56%	11.06%
Med.	43.7%	8.64%

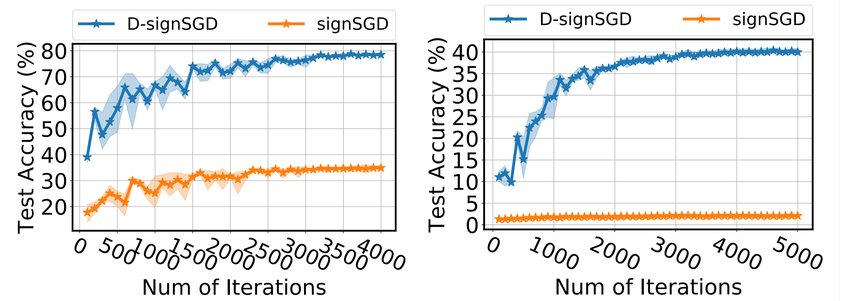
Summery of the defensing performances

ALIE attack:

Byzantine nodes conduct:
Calculate mean: μ_i and
Standard deviation: $\sigma_j, \forall j \in [d]$
across all local calculated gradients
Byzantine nodes send: $\mu_j + z \cdot \sigma_j$
z is treated as a hyper-parameter

In our experiment, we simulate the ALIE attack on the parameter server side

Experimental study II: Pairing DETOX with signSGD [2]



ResNet-18 trained on CIFAR-10 | VGG-13BN trained on CIFAR-100

Main observation:

Applying DETOX leads to significant **gains** in Byzantine resilience of the robust aggregation methods.

Reference

[1] Moran Baruch et al. "A Little Is Enough: Circumventing Defenses For Distributed Learning". https://arxiv.org/abs/1902.06156
[2] Jeremy Bernstein et al. "signSGD with Majority Vote is Communication Efficient and Fault Tolerant". https://arxiv.org/abs/1810.05291

* Authors contributed **equally** to this research and are listed **alphabetically**

Acknowledgement: This research is supported by an NSF CAREER Award #1844951, a Sony Faculty Innovation Award, an AFOSR&AFRL Center of Excellence Award FA9550-18-1-0166, and an NSF TRIPODS Award #1740707.