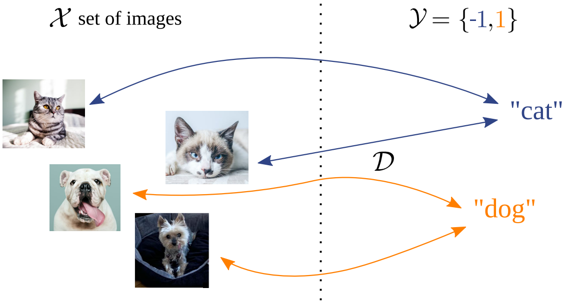
Byzantine-Robustness in Federated Learning

Learning with adversarial data

Rafael Pinot, Sorbonne Université
Nirupam Gupta, University of Copenhagen

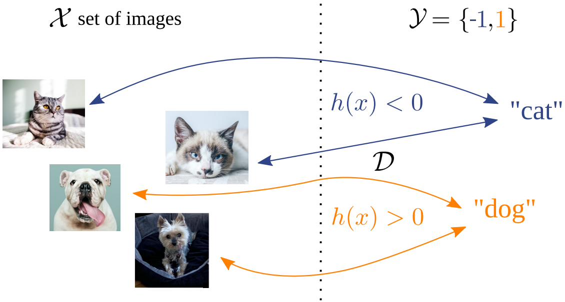
What is Federated Learning (FL)?

Supervised Learning (Example of Image Classification)



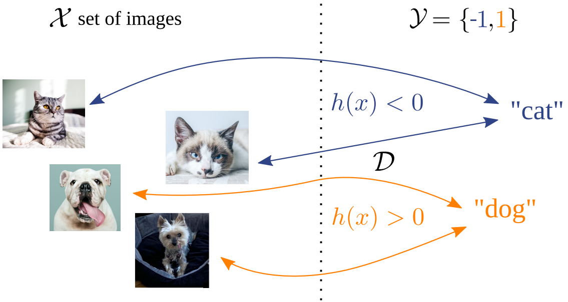
- **Assumption:** A ground-truth distribution $\mathcal{D}$ linking $\mathcal{X}$ and $\mathcal{Y}$

Supervised Learning (Example of Image Classification)



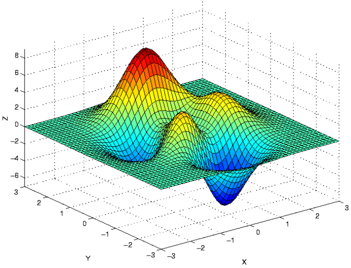
- **Assumption:** A ground-truth distribution $\mathcal{D}$ linking $\mathcal{X}$ and $\mathcal{Y}$
- **Goal:** Use $\mathcal{D}$ to design $h : \mathcal{X} \to \mathbb{R}$ matching images $\mathcal{X}$ to labels $\mathcal{Y}$

Supervised Learning (Example of Image Classification)



- **Assumption:** A ground-truth distribution $\mathcal{D}$ linking $\mathcal{X}$ and $\mathcal{Y}$
- **Goal:** Use $\mathcal{D}$ to design $h : \mathcal{X} \to \mathbb{R}$ matching images $\mathcal{X}$ to labels $\mathcal{Y}$
  1) Define a loss function $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}^+$ and a hypothesis class $\mathcal{H}$
  2) Find $h \in \mathcal{H}$ to minimize the expected error $\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell\left(h(x), y\right)\right]$

Supervised Training in the Centralized Setting



- Given a set of m **training examples**:

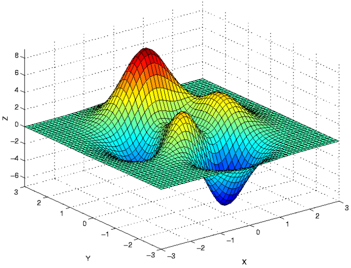
$$\mathcal{S} := \{(x_1, y_1), ..., (x_m, y_m)\} \sim \mathcal{D}^m$$

- Parameterized $\mathcal{H} := \{h_\theta \mid \theta \in \mathbb{R}^d\}$

- Minimize the **empirical risk** (ERM):

$$\mathcal{L}(\theta) := \frac{1}{m} \sum_{i=1}^{m} \ell\left(h_\theta\left(x_i\right), y_i\right)$$

Supervised Training in the Centralized Setting



- Given a set of m **training examples**:

$$\mathcal{S} := \{(x_1, y_1), ..., (x_m, y_m)\} \sim \mathcal{D}^m$$

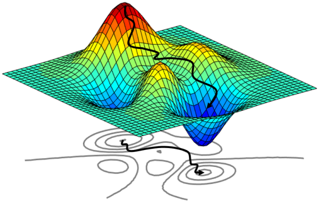
- Parameterized $\mathcal{H} := \{h_\theta \mid \theta \in \mathbb{R}^d\}$
- Minimize the **empirical risk** (ERM):

$$\mathcal{L}(\theta) := \frac{1}{m}\sum_{i=1}^{m} \ell\left(h_\theta\left(x_i\right), y_i\right)$$

Learning objective: Assuming $\mathcal{L}$ admits a minimum on $\mathbb{R}^d$, we seek an ε-approximate solution to the ERM, i.e., $\hat{\theta}$ s.t.

$$\mathcal{L}\left(\hat{\theta}\right) - \mathcal{L}^* \leq \varepsilon, \text{ where } \mathcal{L}^* = \min_{\theta \in \mathbb{R}^d} \mathcal{L}\left(\theta\right).$$

Stochastic Gradient Descent (SGD) in the Centralized Setting



- **Simple** and **efficient** method
- Well understood theoretically
- **Massively used** in practice
  (especially for deep learning tasks)

- Start with an arbitrary parameter θ_1
- At every step $t = 1, \cdots, T$ do:
  - Sample a data point $(x, y) \sim \text{Unif}\left(\mathcal{S}\right)$
  - Compute a stochastic gradient $g_t := \nabla_{\theta_t} \ell\left(h_{\theta_t}\left(x\right), y\right)$
  - Update the parameter $\theta_{t+1} = \theta_t - \gamma_t g_t$

1. Datacenter distributed learning
   → Train a model on a single massive dataset
   → Distribution **limits computations/storage**

Federated/Distributed Machine Learning

1. Datacenter distributed learning

   → Train a model on a single massive dataset
   → Distribution **limits computations/storage**

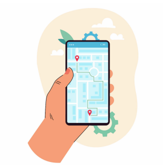
2. Cross-silo distributed/federated learning

   → Datacenters are **geo-distributed**
   → Keeping data locally is safer

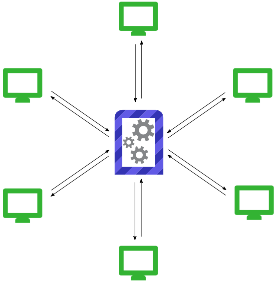
1. Datacenter distributed learning
   - → Train a model on a single massive dataset
   - → Distribution **limits computations/storage**

2. Cross-silo distributed/federated learning
   - → Datacenters are **geo-distributed**
   - → Keeping data locally is safer

3. Cross-device distributed/federated learning
   - → Same distribution/security requirement
   - → **Less computational power** per device
   - → More diversity in the data (**heterogeneity**)



Federated Machine Learning: Problem Statement

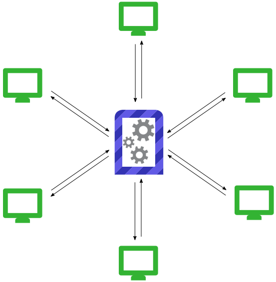


- Server-based communications with n nodes and a (trusted) central server
- **Nodes** hold the data locally $(\mathcal{S}_i)_{i \in [n]}$

$$\mathcal{L}_i(\theta) := \frac{1}{\mid \mathcal{S}_i \mid} \sum_{(x,y) \in \mathcal{S}_i} \ell\left(h_\theta(x), y\right)$$

- The **server** coordinates the training

Federated Machine Learning: Problem Statement



- Server-based communications with n nodes and a (trusted) central server
- **Nodes** hold the data locally $(\mathcal{S}_i)_{i \in [n]}$

$$\mathcal{L}_i(\theta) := \frac{1}{\mid \mathcal{S}_i \mid} \sum_{(x,y) \in \mathcal{S}_i} \ell\left(h_\theta(x), y\right)$$

- The **server** coordinates the training

Training objective: Finding an ε-approximate solution to the ERM for the loss function defined as $\mathcal{L}(\theta) := \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_i(\theta)$

Some Additional Information on the Problem Statement (1/2)

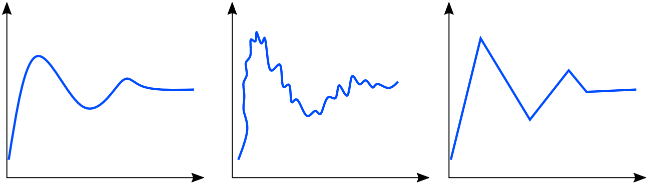
The problem is L-smooth and μ-PL

Some Additional Information on the Problem Statement (1/2)

The problem is L-**smooth** and μ-**PL**

- $\exists L < \infty$ s.t. for all $\theta,\ \theta' \in \mathbb{R}^d$ and any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we have

$$\|\nabla\ell(h_\theta(x), y) - \nabla\ell(h_{\theta'}(x), y)\| \leq L\big\|\theta - \theta'\big\|.$$

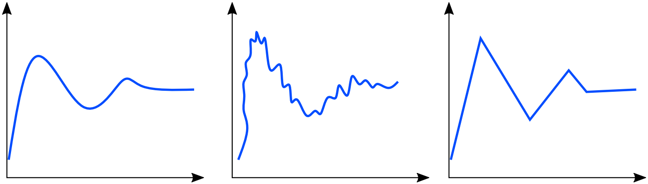


Some Additional Information on the Problem Statement (1/2)

The problem is L-**smooth** and μ-**PL**

- $\exists L < \infty$ s.t. for all $\theta,\ \theta' \in \mathbb{R}^d$ and any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we have

$$\|\nabla \ell(h_\theta(x), y) - \nabla \ell(h_{\theta'}(x), y)\| \leq L \big\|\theta - \theta'\big\|.$$



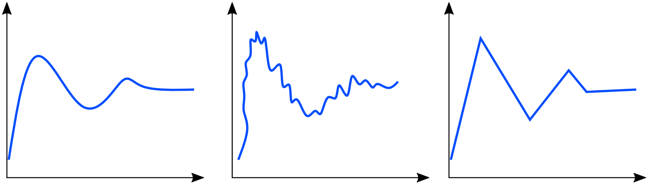
- $\exists \mu < \infty$ s.t. for all $\theta \in \mathbb{R}^d$, we have

$$\|\nabla \mathcal{L}(\theta)\|^2 \geq 2\mu \left(\mathcal{L}(\theta) - \mathcal{L}^*\right) \quad \text{(Polyak's inequality)}$$

The problem is L-**smooth** and μ-**PL**

- $\exists L < \infty$ s.t. for all $\theta,\ \theta' \in \mathbb{R}^d$ and any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we have

$$\|\nabla \ell(h_\theta(x), y) - \nabla \ell(h_{\theta'}(x), y)\| \leq L \big\| \theta - \theta' \big\|.$$



- $\exists \mu < \infty$ s.t. for all $\theta \in \mathbb{R}^d$, we have

$$\|\nabla \mathcal{L}(\theta)\|^2 \geq 2\mu \left(\mathcal{L}(\theta) - \mathcal{L}^* \right) \quad \text{(Polyak's inequality)}$$

$\rightarrow$ Numerical examples neural-network for image classification

Some Additional Information on the Problem Statement (2/2)

- All local datasets have the same size m (everything can be adapted)
  $\rightarrow$ We make this assumption, just for simplicity

Some Additional Information on the Problem Statement (2/2)

- All local datasets have the same size m (everything can be adapted)
  $\rightarrow$ We make this assumption, just for simplicity

- Stochastic gradients have bounded stochasticity

There exists $\sigma < \infty$ s.t. for all $i \in [n]$ and $\theta \in \mathbb{R}^d$,

$$\frac{1}{m} \sum_{(x,y) \in \mathcal{S}_i} \left\| \nabla \ell \left(h_\theta(x), y \right) - \nabla \mathcal{L}_i(\theta) \right\|^2 \leq \sigma^2$$

Some Additional Information on the Problem Statement (2/2)

- All local datasets have the same size m (everything can be adapted)
  $\rightarrow$ We make this assumption, just for simplicity

- Stochastic gradients have bounded stochasticity

  There exists $\sigma < \infty$ s.t. for all $i \in [n]$ and $\theta \in \mathbb{R}^d$,

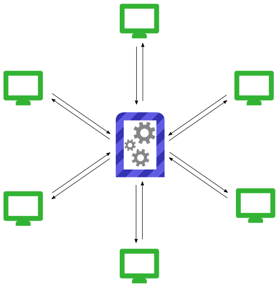
  $$\frac{1}{m} \sum_{(x,y) \in \mathcal{S}_i} \left\| \nabla \ell \left(h_\theta(x), y \right) - \nabla \mathcal{L}_i(\theta) \right\|^2 \leq \sigma^2$$

- Bounded gradient heterogeneity between the nodes

  There exists $G < \infty$ s.t. for all $\theta \in \mathbb{R}^d$,

  $$\frac{1}{n} \sum_{i \in [n]} \left\| \nabla \mathcal{L}_i(\theta) - \nabla \mathcal{L}(\theta) \right\|^2 \leq G^2$$

Distributed Stochastic Gradient Descent (DSGD)



At every step $t = 1, \ldots, T$

- **node** i computes & sends

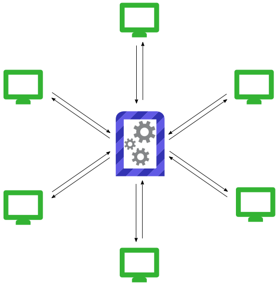
$$g_t^{(i)} = \nabla_{\theta_t} \ell \left(h_{\theta_t} \left(x_i \right), y_i \right),$$

where $(x_i, y_i) \sim \text{Unif}(\mathcal{S}_i)$.

- **Server** updates & broadcasts

$$\theta_{t+1} = \theta_t - \gamma_t \frac{1}{n} \sum_{i=1}^{n} g_t^{(i)}$$

Distributed Stochastic Gradient Descent (DSGD)



At every step $t = 1, \ldots, T$

- **node** i computes & sends
$$g_t^{(i)} = \nabla_{\theta_t} \ell\left(h_{\theta_t}\left(x_i\right), y_i\right),$$
where $(x_i, y_i) \sim$ Unif$(\mathcal{S}_i)$.
- **Server** updates & broadcasts
$$\theta_{t+1} = \theta_t - \gamma_t \frac{1}{n} \sum_{i=1}^{n} g_t^{(i)}$$

Standard convergence see e.g. Koloskova et al. (2020) :

For some $(\gamma_t)_{t \in [T]}$, $\hat{\theta}$ gives an ε-approximate solution (in expectation), with

$$\varepsilon \in \mathcal{O}\left(\frac{\mathcal{K}_{\mathcal{L}} \sigma^2}{nT}\right), \text{ and } \mathcal{K}_{\mathcal{L}} := \frac{L}{\mu}.$$

Distributed Learning is So Great!

- So distributed learning is easy to implement, efficient and trendy ...
- This means that we can use it for many **great applications**

Distributed Learning is So Great!

- So distributed learning is easy to implement, efficient and trendy ...
- This means that we can use it for many **great applications**

Personal Devices

Distributed Learning is So Great!

- So distributed learning is easy to implement, efficient and trendy ...
- This means that we can use it for many **great applications**

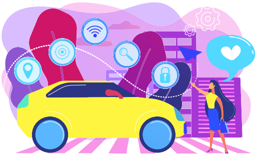
Decision-making Tools

Personal Devices

Distributed Learning is So Great!

- So distributed learning is easy to implement, efficient and trendy ...
- This means that we can use it for many **great applications**

AI-driven Technologies



Decision-making Tools

Personal Devices

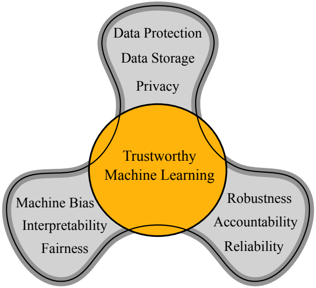
Things Can Go Wrong 1/2

Things Can Go Wrong 2/2

Things can go wrong in many ways …

Things Can Go Wrong 2/2

Things can go wrong in many ways ...



- **Since the 80's:** privacy preserving database analysis is a primary concern
- **More recently:** fairness/robustness to adversarial examples
- Some are more specific to Federated Learning (Byzantine failures)

Robustness to Byzantine Nodes

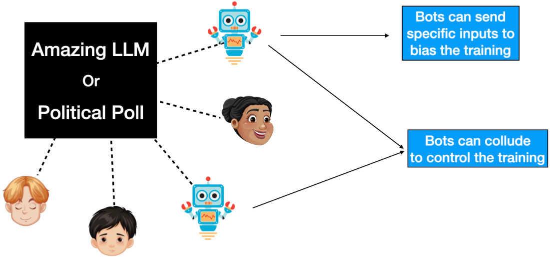
In Practice, Misbehaving Nodes Are Inevitable

- **Software bugs** and **Hardware crashes** can occur
  $\rightarrow$ Add errors/arbitrary values in the computations

In Practice, Misbehaving Nodes Are Inevitable

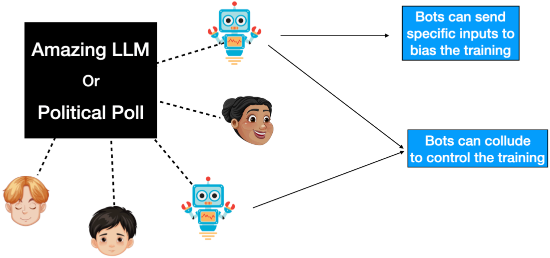
- **Software bugs** and **Hardware crashes** can occur
  → Add errors/arbitrary values in the computations

- Some nodes may have **poisoned or irrelevant** data or can get **hacked**



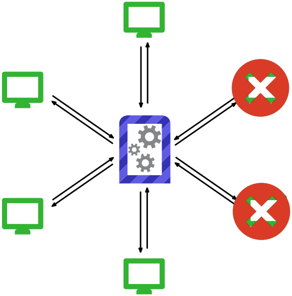
In Practice, Misbehaving Nodes Are Inevitable

- **Software bugs** and **Hardware crashes** can occur
  $\longrightarrow$ Add errors/arbitrary values in the computations
- Some nodes may have **poisoned or irrelevant** data or can get **hacked**



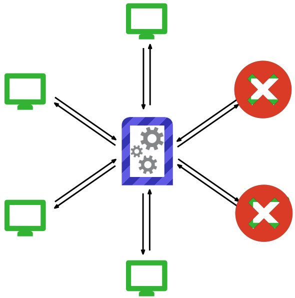
Challenge: We do not know which nodes may misbehave (nor how)

The Byzantine Threat Model



- We take the **Byzantine threat model** inherited from Lamport et al. (1982)
- Up to $f < n/2$ nodes may be bad
- When i is Byzantine we have

$$g_t^{(i)} = *, \ \forall t \in [T] \quad \textit{(Synchrony)}$$



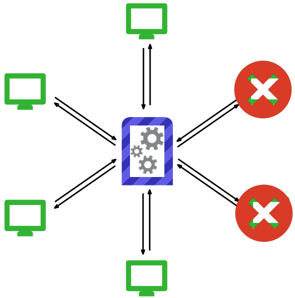
- We take the **Byzantine threat model** inherited from Lamport et al. (1982)
- Up to $f < n/2$ nodes may be bad
- When i is Byzantine we have

$$g_t^{(i)} = *, \ \forall t \in [T] \quad \textit{(Synchrony)}$$

New objective: Denote H the set of honest (non-Byzantine) nodes. We seek an ε-approximate solution to the ERM for the loss function defined as

$$\mathcal{L}_H(\theta) := \frac{1}{n-f} \sum_{i \in H} \mathcal{L}_i(\theta) \qquad \text{(a.k.a. } (f, \varepsilon)\text{-Byzantine resilience)}$$

The Byzantine Threat Model



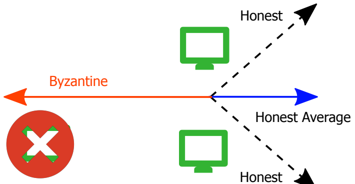
- We take the **Byzantine threat model** inherited from Lamport et al. (1982)
- Up to $f < n/2$ nodes may be bad
- When i is Byzantine we have

$$g_t^{(i)} = *, \ \forall t \in [T] \quad \textit{(Synchrony)}$$

New objective: Denote H the set of honest (non-Byzantine) nodes. We seek an ε-approximate solution to the ERM for the loss function defined as

$$\mathcal{L}_H(\theta) := \frac{1}{n-f} \sum_{i \in H} \mathcal{L}_i(\theta) \quad \text{(a.k.a. } (f, \varepsilon)\text{-Byzantine resilience)}$$

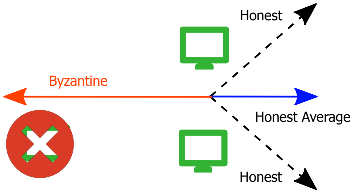
$\rightarrow$ Despite the f Byzantine nodes (and not knowing H a priori)



Recall update rule at the server:

$$\theta_{t+1} = \theta_t - \gamma_t \frac{1}{n} \sum_{i=1}^{n} g_t^{(i)}$$

Hence is arbitrarily manipulable by a **single** Byzantine node.



Recall update rule at the server:

$$\theta_{t+1} = \theta_t - \gamma_t \frac{1}{n} \sum_{i=1}^{n} g_t^{(i)}$$

Hence is arbitrarily manipulable by a **single** Byzantine node.

A standard approach to confer Byzantine robustness:

Replace the averaging with a **non-linear** aggregation rule $A : \mathbb{R}^{d \times n} \to \mathbb{R}^d$:

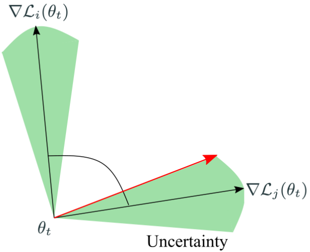
$$\theta_{t+1} = \theta_t - \gamma_t \, A\left(g_t^{(1)}, \ldots, g_t^{(n)}\right)$$

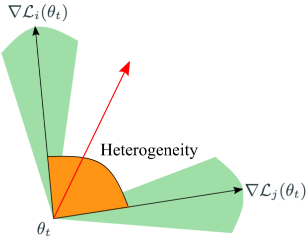
$\rightarrow$ Choosing A is close to the **robust mean estimation** problem

What are the Bottlenecks ?

- Some robust estimation schemes can be adapted, but **beware of assumptions** on the distributions
- We have to be careful about of the **model drift accumulation**

Two main bottlenecks: uncertainty and heterogeneity

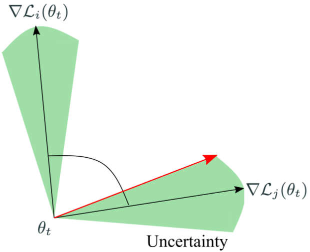


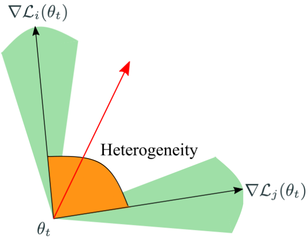


What are the Bottlenecks ?

- Some robust estimation schemes can be adapted, but **beware of assumptions** on the distributions
- We have to be careful about of the **model drift accumulation**

Two main bottlenecks: uncertainty and heterogeneity



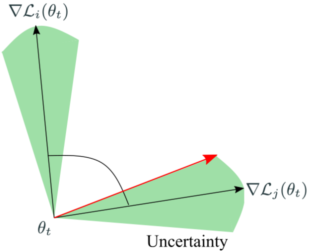


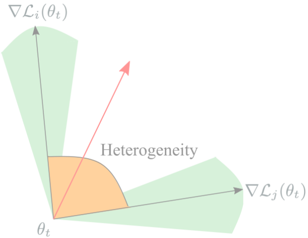
$\rightarrow$ This only arises due to the presence of Byzantine nodes

What are the Bottlenecks ?

- Some robust estimation schemes can be adapted, but **beware of assumptions** on the distributions
- We have to be careful about of the **model drift accumulation**

Two main bottlenecks: uncertainty and heterogeneity





$\rightarrow$ This only arises due to the presence of Byzantine nodes

What Can We Do About Uncertainty ?

Some Famous Aggregation Rules 1/2

Simple coordinate-wise solutions ($n = 5,\ f = 1,\ d = 2$):

Coordinate-wise median (CW-Med)

→ Compute the median per coordinate.

$$\text{CW-Med}\begin{pmatrix} 3 & 1 & 3 & 6 & 8 \\ 6 & 2 & 4 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$$

Simple coordinate-wise solutions ($n = 5$, $f = 1$, $d = 2$):

Coordinate-wise median (CW-Med)

→ Compute the median per coordinate.

$$\text{CW-Med}\begin{pmatrix} 3 & 1 & 3 & 6 & 8 \\ 6 & 2 & 4 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$$

Coordinate-wise trimmed mean (CW-TM)

→ Remove f biggest and f smallest coordinates on each dimension, and then average.

$$\text{CW-TM}\begin{pmatrix} 3 & \cancel{1} & 3 & 6 & \cancel{8} \\ \cancel{6} & 2 & 4 & 3 & \cancel{1} \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}$$

Some Famous Aggregation Rules 1/2

Simple coordinate-wise solutions ($n = 5$, $f = 1$, $d = 2$):

Coordinate-wise median (CW-Med)

$\rightarrow$ Compute the median per coordinate.

$$\text{CW-Med}\begin{pmatrix} 3 & 1 & 3 & 6 & 8 \\ 6 & 2 & 4 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$$

Coordinate-wise trimmed mean (CW-TM)

$\rightarrow$ Remove f biggest and f smallest coordinates on each dimension, and then average.

$$\text{CW-TM}\begin{pmatrix} 3 & \cancel{1} & 3 & 6 & \cancel{8} \\ \cancel{6} & 2 & 4 & 3 & \cancel{1} \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}$$

Both these solutions have been analyzed, e.g., in Yin et al. (2018).

More sophisticated aggregations:

Geometric median Chen et al. (2017)

$\rightarrow$ Output a vector that realizes the geometric median of the send gradients, i.e.,

$$\mathrm{GM}\left(v_1, \ldots, v_n\right) \in \mathrm{argmin}_{v \in \mathbb{R}^d} \sum_{i=1}^{n} \|v - v_i\|.$$

More sophisticated aggregations:

Geometric median Chen et al. (2017)

→ Output a vector that realizes the geometric median of the send gradients, i.e.,

$$\mathrm{GM}\left(v_1, \ldots, v_n\right) \in \operatorname{argmin}_{v \in \mathbb{R}^d} \sum_{i=1}^{n} \|v - v_i\|.$$

MDA Rousseeuw (1985)

→ Choose a set S^* of $n-f$ indices with the smallest *diameter*. Then average over S^*, i.e.,

$$\mathrm{MDA}\left(v_1, \ldots, v_n\right) = \frac{1}{n-f} \sum_{i \in S^*} v_i.$$

Some Famous Aggregation Rules 2/2

More sophisticated aggregations:

Geometric median Chen et al. (2017)
→ Output a vector that realizes the geometric median of the send gradients, i.e.,

$$\mathrm{GM}\left(v_1, \ldots, v_n\right) \in \mathrm{argmin}_{v \in \mathbb{R}^d} \sum_{i=1}^{n} \|v - v_i\|.$$

MDA Rousseeuw (1985)
→ Choose a set S^* of $n - f$ indices with the smallest *diameter*. Then average over S^*, i.e.,

$$\mathrm{MDA}\left(v_1, \ldots, v_n\right) = \frac{1}{n-f} \sum_{i \in S^*} v_i.$$

But also **MeaMed** Xie et al. (2018), **Krum, Multi-Krum** Blanchard et al. (2017) ...

A First Step Towards Resilience

Common notion of (f, κ)-robust averaging Allouah et al. (2023) **:**

For any n vectors $v_1, \dots, v_n \in \mathbb{R}^d$ and any subset $S \subseteq [n]$ of size $n - f$,

$$\left\| A\left(v_1, \dots, v_n\right) - \overline{v}_S \right\|^2 \leq \frac{\kappa}{n-f} \sum_{i \in S} \left\| v_i - \overline{v}_S \right\|^2,$$

where $\overline{v}_S := \frac{1}{n-f} \sum_{i \in S} v_i$

A First Step Towards Resilience

Common notion of (f, κ)-robust averaging Allouah et al. (2023) **:**

For any n vectors $v_1, \ldots, v_n \in \mathbb{R}^d$ and any subset $S \subseteq [n]$ of size $n - f$,

$$\|A(v_1, \ldots, v_n) - \overline{v}_S\|^2 \leq \frac{\kappa}{n-f} \sum_{i \in S} \|v_i - \overline{v}_S\|^2,$$

where $\overline{v}_S := \frac{1}{n-f} \sum_{i \in S} v_i$

Quick sanity check: If $\sigma^2 = 0$ and $G = 0$ the honest workers are identical (full gradients on identical data)

$$\sum_{i \in S} \|v_i - \overline{v}_S\|^2 = 0$$

$\rightarrow$ The aggregation rule should mimic the majority voting scheme.

A First Step Towards Resilience

Common notion of (f, κ)-robust averaging Allouah et al. (2023) **:**

For any n vectors $v_1, \dots, v_n \in \mathbb{R}^d$ and any subset $S \subseteq [n]$ of size $n - f$,

$$\left\|A\left(v_1, \dots, v_n\right) - \overline{v}_S\right\|^2 \leq \frac{\kappa}{n-f} \sum_{i \in S} \left\|v_i - \overline{v}_S\right\|^2,$$

where $\overline{v}_S := \frac{1}{n-f} \sum_{i \in S} v_i$

$\rightarrow$ This definition is satisfied by many existing aggregation rules.

A First Step Towards Resilience

Common notion of (f, κ)-robust averaging Allouah et al. (2023) **:**

For any n vectors $v_1, \dots, v_n \in \mathbb{R}^d$ and any subset $S \subseteq [n]$ of size $n - f$,

$$\left\| A\left(v_1, \dots, v_n\right) - \overline{v}_S \right\|^2 \leq \frac{\kappa}{n-f} \sum_{i \in S} \left\| v_i - \overline{v}_S \right\|^2,$$

where $\overline{v}_S := \frac{1}{n-f} \sum_{i \in S} v_i$

$\rightarrow$ This definition is satisfied by many existing aggregation rules.

Agg.	CW-TM	GM	CW-Med	L.B.
κ	$\mathcal{O}\left(\frac{f}{n-2f}\right)$	$\mathcal{O}\left(1 + \frac{f}{n-2f}\right)$	$\mathcal{O}\left(1 + \frac{f}{n-2f}\right)$	$\Omega\left(\frac{f}{n-2f}\right)$

Applies to **Krum**, **Multi-Krum** Blanchard et al. (2017) and **MeaMed** Xie et al. (2018).

Common notion of (f, κ)-robust averaging Allouah et al. (2023):

For any n vectors $v_1, \dots, v_n \in \mathbb{R}^d$ and any subset $S \subseteq [n]$ of size $n - f$,

$$\left\|A\left(v_1, \dots, v_n\right) - \overline{v}_S\right\|^2 \leq \frac{\kappa}{n-f} \sum_{i \in S} \left\|v_i - \overline{v}_S\right\|^2,$$

where $\overline{v}_S := \frac{1}{n-f} \sum_{i \in S} v_i$

Convergence result in the homogeneous case ($G = 0$):

A First Step Towards Resilience

Common notion of (f, κ)-robust averaging Allouah et al. (2023):

For any n vectors $v_1, \dots, v_n \in \mathbb{R}^d$ and any subset $S \subseteq [n]$ of size $n - f$,

$$\left\| A\left(v_1, \dots, v_n\right) - \overline{v}_S \right\|^2 \leq \frac{\kappa}{n-f} \sum_{i \in S} \left\| v_i - \overline{v}_S \right\|^2,$$

where $\overline{v}_S := \frac{1}{n-f} \sum_{i \in S} v_i$

Convergence result in the homogeneous case $(G = 0)$:

If A is an (f, κ)-robust averaging, for some $(\gamma_t)_{t \in [T]}$, the algorithm satisfies (f, ε)-Byzantine resilience with

$$\varepsilon \in \mathcal{O}\left(\frac{\mathcal{K}_{\mathcal{L}_H} \sigma^2}{(n-f)T} + \kappa \sigma^2 \right)$$

Common notion of (f, κ)-robust averaging Allouah et al. (2023):

For any n vectors $v_1, \ldots, v_n \in \mathbb{R}^d$ and any subset $S \subseteq [n]$ of size $n - f$,

$$\left\| A\left(v_1, \ldots, v_n\right) - \overline{v}_S \right\|^2 \leq \frac{\kappa}{n-f} \sum_{i \in S} \left\| v_i - \overline{v}_S \right\|^2,$$

where $\overline{v}_S := \frac{1}{n-f} \sum_{i \in S} v_i$

Convergence result in the homogeneous case $(G = 0)$:

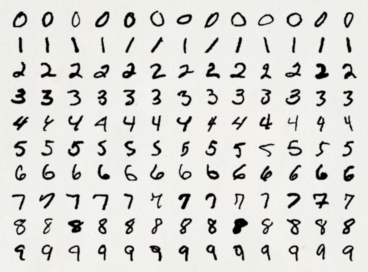
If A is an (f, κ)-robust averaging, for some $(\gamma_t)_{t \in [T]}$, the algorithm satisfies (f, ε)-Byzantine resilience with

$$\varepsilon \in \mathcal{O}\left(\frac{\mathcal{K}_{\mathcal{L}_H} \sigma^2}{(n-f)T} + \kappa \sigma^2\right)$$

$\rightarrow$ This incompressible error might be problematic in practice.

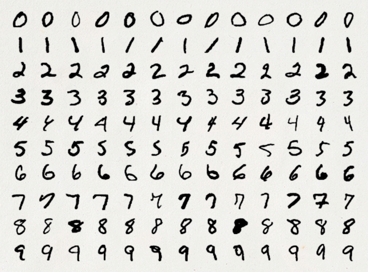
Some Numerical Observations: Model Setting

Learning task: MNIST hand-written digit image classification task with $n = 15$ nodes out of which $f = 5$ might be Byzantine.



Some Numerical Observations: Model Setting

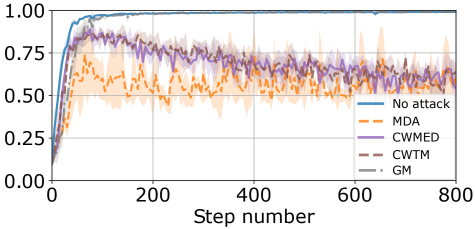
Learning task: MNIST hand-written digit image classification task with $n = 15$ nodes out of which $f = 5$ might be Byzantine.

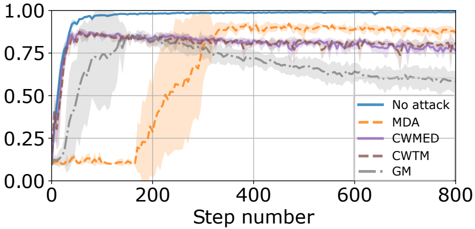


Adversarial behaviors: The Byzantine nodes apply either of the following:

- *Label-flipping*: shift the label of each image 0123456789 $\rightarrow$ 1234567890
- *Sign-flipping*: send the inverse of the local gradient $\mathbf{g}_\mathbf{t}^{(\mathbf{i})} \rightarrow -\mathbf{g}_\mathbf{t}^{(\mathbf{i})}$

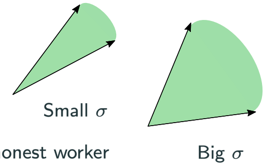
Training accuracy of a CNN along the learning procedure on MNIST. On the **left** *label-flipping* attack and on the **right** *sign-flipping* attack.





Recall the challenge we focus on here:

$$\mathbb{E}\left[\left\|g_t^{(i)} - \mathbb{E}\left[g_t^{(i)}\right]\right\|^2\right] \leq \sigma^2$$



Range of plausible gradients for an honest worker

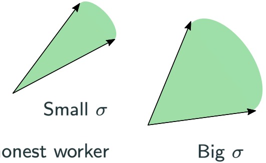
Let's reduce uncertainty!

- **Option 1:** Reduce the noise by using larger mini-batches?

Why is There Still a Gap?

Recall the challenge we focus on here:

$$\mathbb{E}\left[\left\|g_t^{(i)} - \mathbb{E}\left[g_t^{(i)}\right]\right\|^2\right] \leq \sigma^2$$



Range of plausible gradients for an honest worker

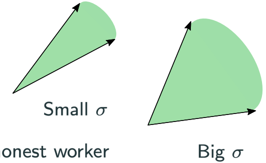
Let's reduce uncertainty!

- **Option 1:** Reduce the noise by using larger mini-batches? Inflates the computationnal cost of the method quite a lot.
- **Option 2:** Learn from past gradients using momentum?

Why is There Still a Gap?

Recall the challenge we focus on here:

$$\mathbb{E}\left[\left\|g_t^{(i)} - \mathbb{E}\left[g_t^{(i)}\right]\right\|^2\right] \leq \sigma^2$$



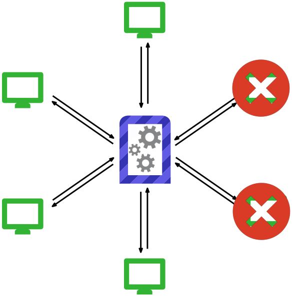
Range of plausible gradients for an honest worker

Let's reduce uncertainty!

- **Option 1:** Reduce the noise by using larger mini-batches? Inflates the computationnal cost of the method quite a lot.

- **Option 2:** Learn from past gradients using momentum? Obviously much better since this is what I will present in the next slide.

Controlling Gradient Variance with Momentum



- **Honest node** i computes

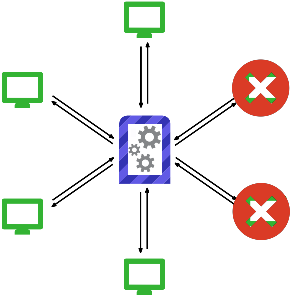
$$m_t^{(i)} = \beta_t m_{t-1}^{(i)} + (1 - \beta_t) g_t^{(i)},$$

where $m_0^{(i)} = 0$ and $\beta_t \in [0,\, 1)$.

- **Server** updates & broadcasts

$$\theta_{t+1} = \theta_t - \gamma_t A\left(m_t^{(1)}, \dots, m_t^{(n)}\right)$$

Controlling Gradient Variance with Momentum



- **Honest node** i computes

$$m_t^{(i)} = \beta_t m_{t-1}^{(i)} + (1 - \beta_t) g_t^{(i)},$$

where $m_0^{(i)} = 0$ and $\beta_t \in [0,\, 1)$.

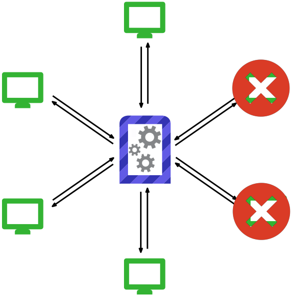
- **Server** updates & broadcasts

$$\theta_{t+1} = \theta_t - \gamma_t A\left(m_t^{(1)}, \ldots, m_t^{(n)}\right)$$

Using the above algorithm (with $\beta_t \equiv \beta$) we have

$$\frac{1}{H} \sum_{i \in H} \mathbb{E}\left[\left\| m_t^{(i)} - \overline{m}_t \right\|^2\right] \in \mathcal{O}\left(\frac{1-\beta}{1+\beta} \sigma^2\right), \text{ with } \overline{m}_t := \frac{1}{(n-f)} \sum_{i \in H} m_t^{(i)}.$$

Controlling Gradient Variance with Momentum



- **Honest node** i computes

$$m_t^{(i)} = \beta_t m_{t-1}^{(i)} + (1 - \beta_t) g_t^{(i)},$$

where $m_0^{(i)} = 0$ and $\beta_t \in [0,\, 1)$.

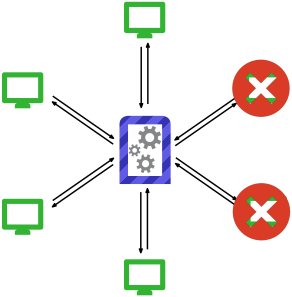
- **Server** updates & broadcasts

$$\theta_{t+1} = \theta_t - \gamma_t A\left(m_t^{(1)}, \dots, m_t^{(n)}\right)$$

Using the above algorithm (with $\beta_t \equiv \beta$) we have

$$\frac{1}{H}\sum_{i \in H} \mathbb{E}\left[\left\| m_t^{(i)} - \overline{m}_t \right\|^2\right] \in \mathcal{O}\left(\frac{1-\beta}{1+\beta}\sigma^2\right), \text{ with } \overline{m}_t := \frac{1}{(n-f)}\sum_{i \in H} m_t^{(i)}.$$

$\rightarrow$ β_t is driving the "noise reduction" but also creates a bias.



- **Honest node** i computes

$$m_t^{(i)} = \beta_t m_{t-1}^{(i)} + (1 - \beta_t) g_t^{(i)},$$

where $m_0^{(i)} = 0$ and $\beta_t \in [0,\, 1)$.

- **Server** updates & broadcasts

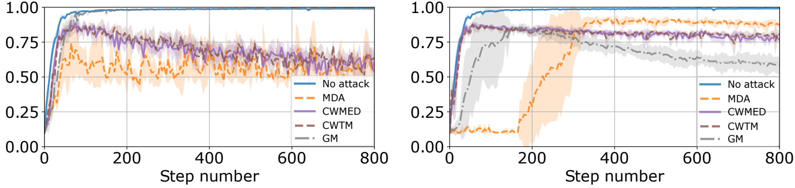
$$\theta_{t+1} = \theta_t - \gamma_t A\left(m_t^{(1)}, \dots, m_t^{(n)}\right)$$

Convergence result in the homogeneous case Farhadkhani et al. (2022, 2023):

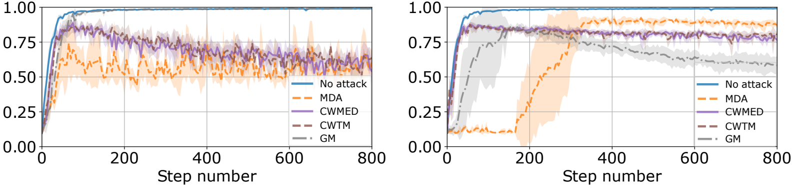
Assume A is an (f, κ)-robust averaging. For some $(\gamma_t)_{t \in [T]}$, setting $\beta_t := 1 - c\gamma_t, \forall t \in [T]$, the algorithm satisfies (f, ε)-Byzantine resilience with

$$\varepsilon \in \mathcal{O}\left(\left(\kappa + \frac{1}{(n-f)}\right) \frac{\mathcal{K}_{\mathcal{L}_H} \sigma^2}{T}\right)$$

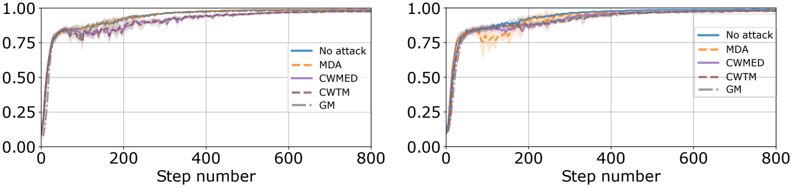
Impact of the Momentum on Byzantine Resilience



Same setting as before. **Up** without momentum ($\beta_t \equiv 0$)



Same setting as before. **Up** without momentum ($\beta_t \equiv 0$) and **down** with momentum ($\beta_t \equiv 0.99$)

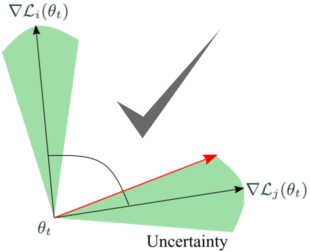


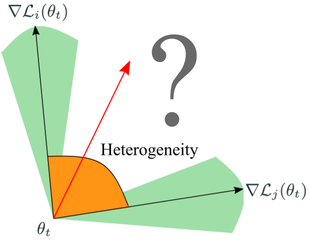
What About Heterogeneity?

What are the Bottlenecks? (repetitio)

- Some robust estimation schemes can be adapted, but **beware of assumptions** on the distributions
- We have to be careful about of the **model drift accumulation**

Two main bottlenecks: uncertainty and heterogeneity



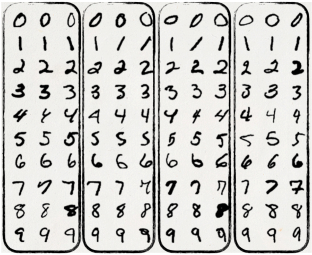


$\rightarrow$ This only arises due to the presence of Byzantine nodes

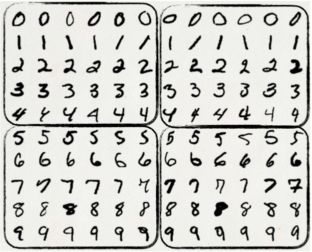
What Do We Mean By Heterogeneity?

(Updated) Heterogeneity Assumption: There exists $G^2 < \infty$ s.t. $\forall \theta \in \mathbb{R}^d$,

$$\frac{1}{n-f} \sum_{i \in H} \left\| \nabla \mathcal{L}_i(\theta) - \nabla \mathcal{L}_H(\theta) \right\|^2 \leq G^2$$



$G^2 \approx 0$



$G^2 \gg 0$

Simulating Heterogeneity

For **each class** $y \in \mathcal{Y}$, sample the proportion of this class' data-points held by each client using a **symmetric Dirichlet distribution** $\mathbb{D}_n(\alpha)$, with $\alpha > 0$.

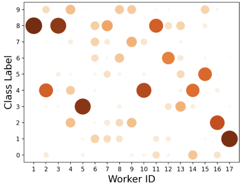
$\rightarrow$ Sampling in point from the simplex with concentration driven by α.
($\alpha = 1$ we get uniform sampling on the simplex)

For **each class** $y \in \mathcal{Y}$, sample the proportion of this class' data-points held by each client using a **symmetric Dirichlet distribution** $\mathbb{D}_n(\alpha)$, with $\alpha > 0$.

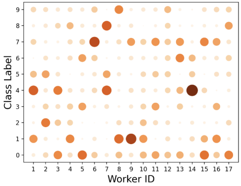
$\rightarrow$ Sampling in point from the simplex with concentration driven by α.

($\alpha = 1$ we get uniform sampling on the simplex)

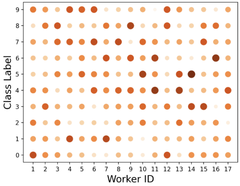
$\alpha = 0.1$



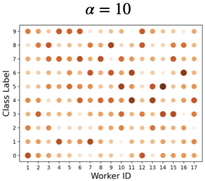
$\alpha = 1$

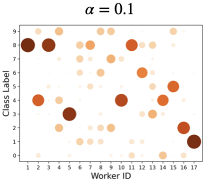


$\alpha = 10$

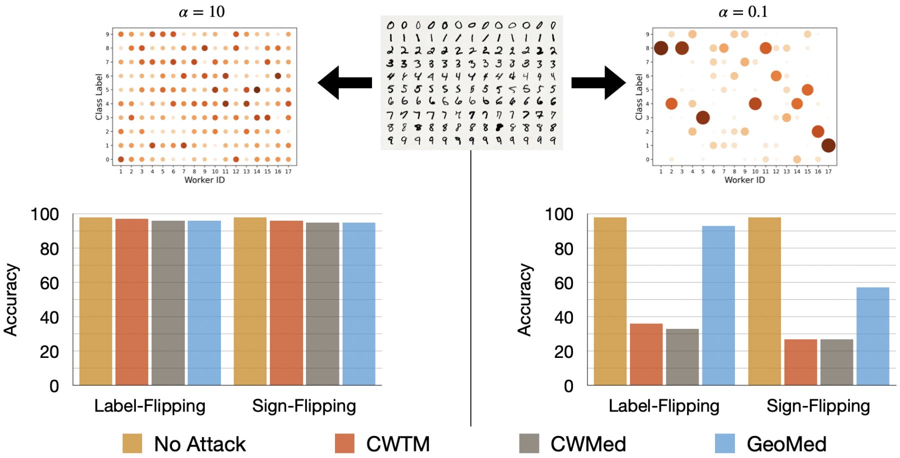


Some Numerical Observations on Heterogeneity





Some Numerical Observations on Heterogeneity



Training CNN on $n = 17$ nodes where $f = 4$ nodes are Byzantine. MNIST dataset split using Dirichlet distribution.

Does This Appear in Theory?

Lower bound see e.g. Karimireddy et al. (2022):

There exists a set of loss functions satisfying our assumptions for which we **cannot** reach an ϵ-solution unless $\epsilon \in \Omega\left(\frac{f}{n}G^2\right)$.

Does This Appear in Theory?

Lower bound see e.g. Karimireddy et al. (2022):

> There exists a set of loss functions satisfying our assumptions for which we **cannot** reach an ϵ-solution unless $\epsilon \in \Omega\left(\frac{f}{n} G^2\right)$.

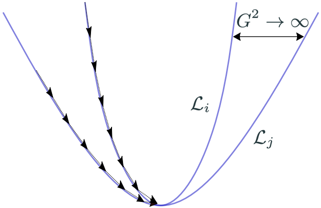
- Similar to uncertainty (indistinguishability)
- This is a very pessimistic bound

Matching upper bound see e.g. Allouah et al. (2023):

> Using the previous algorithm with momentum and A a (κ, f)-robust averaging, we have $\epsilon \in \mathcal{O}\left(\kappa G^2\right)$.

Heterogeneity is an Open Problem

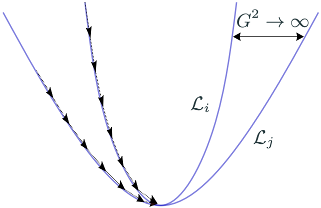
Is this not too pessimistic ?



- **Uniform bound** on the entire space $\mathbb{R}^d$
- Some parts of the space are more interesting than others.
- Even **criticized** in federated learning standard settings Wang et al. (2022)

Heterogeneity is an Open Problem

Is this not too pessimistic ?



- **Uniform bound** on the entire space $\mathbb{R}^d$
- Some parts of the space are more interesting than others.
- Even **criticized** in federated learning standard settings Wang et al. (2022)

We need more realistic (tighter) measurements of heterogeneity in distributed learning (only on the optima, or modular bounds).

Other Open Problems I Did Not Talk About

- Existing research is mostly focused on first-order methods
  → Little has been done on higher order/gradient free methods

- Similarly, research is mostly focused on federated settings
  → Little (a bit more though) has been done on decentralized methods

- Here we mainly focus on the robustness of the algorithm at training time
  → What about generalization? How to be robust to test-time triggered attacks such as "Backdoor attacks", see e.g. Nguyen et al. (2023)

- We did not mention other concerns (**privacy**, fairness, bias, etc.)
  → Seem conflicting, but ultimately, we need to combine them.

Thanks for listening!

References

Allouah, Y., Farhadkhani, S., Guerraoui, R., Gupta, N., Pinot, R., and Stephan, J. (2023). Fixing by mixing: A recipe for optimal Byzantine ML under heterogeneity. In *International Conference on Artificial Intelligence and Statistics*, pages 1232–1300. PMLR.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 119–129. Curran Associates, Inc.

Chen, Y., Su, L., and Xu, J. (2017). Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25.

Farhadkhani, S., Guerraoui, R., Gupta, N., Hoang, L.-N., Pinot, R., and Stephan, J. (2023). Robust collaborative learning with linear gradient overhead.

Farhadkhani, S., Guerraoui, R., Gupta, N., Pinot, R., and Stephan, J. (2022). Byzantine machine learning made easy by resilient averaging of momentums. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S., editors, *International Conference on Machine Learning,* *ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 6246–6283. PMLR.

Karimireddy, S. P., He, L., and Jaggi, M. (2022). Byzantine-robust learning on heterogeneous datasets via bucketing. In *International Conference on Learning Representations*.

Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., and Stich, S. (2020). A unified theory of decentralized SGD with changing topology and local updates. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5381–5393. PMLR.

Lamport, L., Shostak, R., and Pease, M. (1982). The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401.

Nguyen, T. D., Nguyen, T., Nguyen, P. L., Pham, H. H., Doan, K., and Wong, K.-S. (2023). Backdoor attacks and defenses in federated learning: Survey, challenges and future research directions.

Rousseeuw, P. J. (1985). Multivariate estimation with high breakdown point. *Mathematical statistics and applications*, 8(37):283–297.

Wang, J., Das, R., Joshi, G., Kale, S., Xu, Z., and Zhang, T. (2022). On the unreasonable effectiveness of federated averaging with heterogeneous data.

Xie, C., Koyejo, O., and Gupta, I. (2018). Generalized byzantine-tolerant sgd.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR.