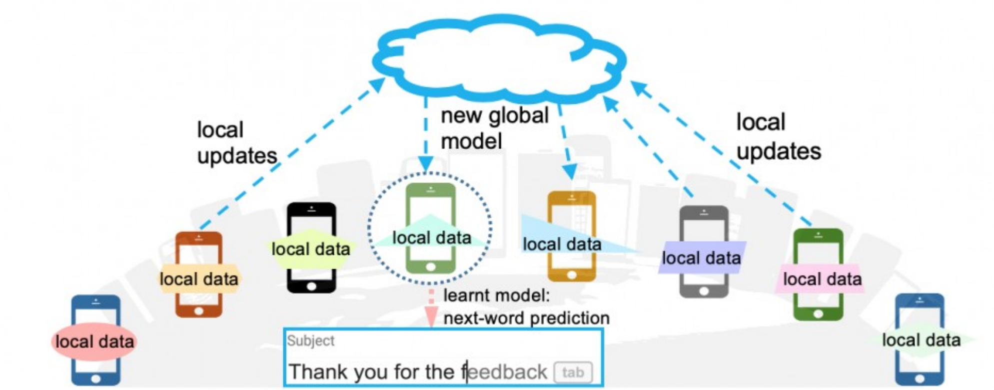
Federated / Distributed Learning

Multiple computational nodes

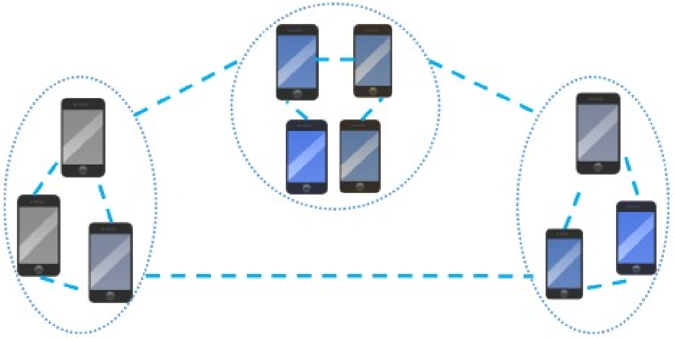
or



Types

- **Distributed: single node not powerful enough**
- **Federated: data locality!**
  - Cross-silo: companies collaborating
  - Cross-device: edge device

- **(Fully decentralized)**



Distributed (but centralized) Learning

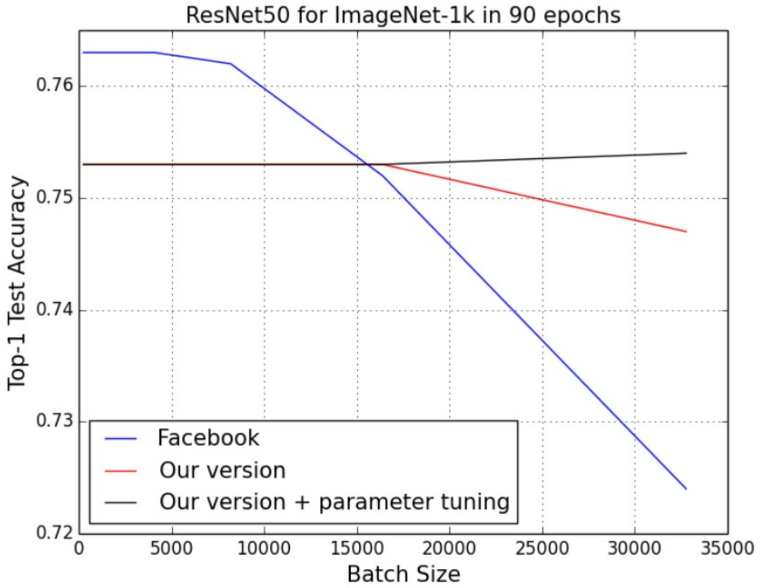
- **models too big for a single node**
- **SGD is difficult to parallelize**
- so we use **larger batch sizes**

Accurate, large minibatch sgd: **Training imagenet** in 1 **hour**

P Goyal, P Dollár, R Girshick, P Noordhuis… - arXiv preprint arXiv …, 2017 - arxiv.org

… In this paper, we empirically show that on the **ImageNet** … Specifically, we show no loss of accuracy when **training** with … optimization challenges early in **training**. With these simple …

☆ Save 99 Cite Cited by 2721 Related articles All 8 versions

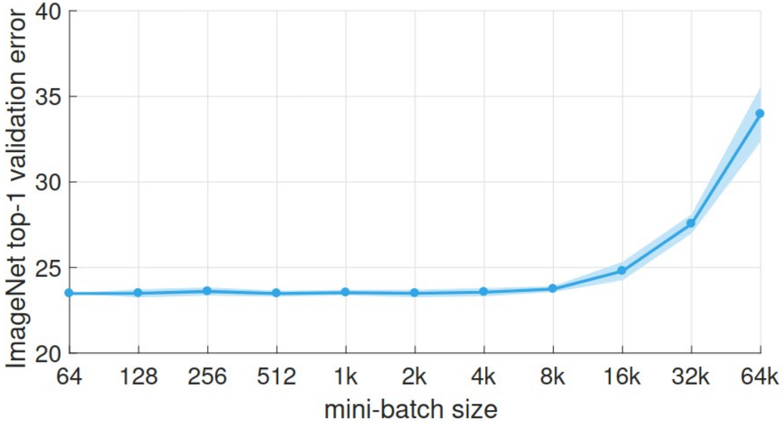


Images/plots taken from paper

Large batch sizes

- **poor generalization**
- **sharper minima**
- **partly resolved by larger step sizes**

Linear Scaling Rule: *When the minibatch size is multiplied by k, multiply the learning rate by k.*



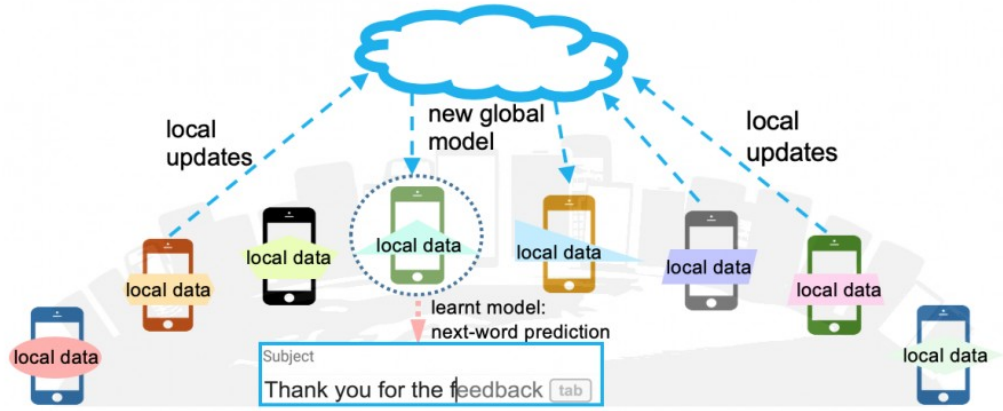
- and many more tricks: **LARS**

[PDF] **Scaling sgd batch size to 32k for imagenet training**
Y You, I Gitman, B Ginsburg - arXiv preprint arXiv:1708.03888, 2017 - fid3024.github.io
… we increase the **batch size** from 128 to 8192 for AlexNet model. For ResNet50 model, we successfully **scaled** the **batch size** to 32768 in **ImageNet training**. Large **batch** can make full …
Save Cite Cited by 299 Related articles All 6 versions

Images/plots taken from paper

Federated learning

- Few papers in 2016, over 3k in 2020
- Data locality / privacy is key; stateless clients
- Bottleneck: communication
  - Upload quite slow
- Applications:
  - Gboard keybord, “Hey Siri”, …
  - Health record, pharmaceutical



Images/plots taken from paper

Mitigate communication bottleneck

- Communicate less frequently
- Compress information
- Use more devices

Communication-efficient learning of deep networks from decentralized data
B McMahan, E Moore, D Ramage… - Artificial intelligence …, 2017 - proceedings.mlr.press

We investigate both of these approaches, but the speedups we achieve are due primarily to adding more computation on each client, once a minimum level of parallelism over clients is used.

Save Cite Cited by 6872 Related articles All 5 versions

Images/plots taken from paper

About compression

- **reduce precision: Q**(uantized)**SGD**
- already **single or half**
- **use just the sign: signSGD/TernGRAD**
- still scales **linearly in dimension**
- **Top-k (rank-k) compressors**

*sign*SGD (Quantization)

$$\text{signsgd}\left(\begin{bmatrix}-0.2\\0.1\\0.3\\-1.8\end{bmatrix}\right) \Rightarrow \begin{bmatrix}-1\\1\\1\\-1\end{bmatrix}$$

Communicating less frequently

- run multiple step on clients
- proposed in first FL papers
- problems with client drift

DON'T USE LARGE MINI-BATCHES, USE LOCAL SGD

Tao Lin
EPFL, Switzerland
tao.lin@epfl.ch

Sebastian U. Stich
EPFL, Switzerland
sebastian.stich@epfl.ch

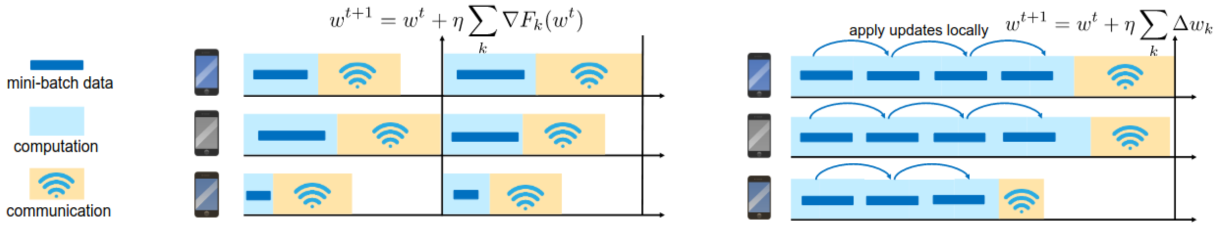
Kumar Kshitij Patel
IIT Kanpur, India
kumarkshitijpatel@gmail.com

Martin Jaggi
EPFL, Switzerland
martin.jaggi@epfl.ch

Server executes:
initialize x_0
for each round $t = 1, 2, \ldots,$ T **do**
$S_t \leftarrow$ (random set of M clients)
for each client $i \in S_t$ **in parallel do**
$x^i_{t+1} \leftarrow \text{ClientUpdate}(i, x_t)$
$x_{t+1} \leftarrow \sum_{k=1}^{M} \frac{1}{M} x^i_{t+1}$

ClientUpdate(i, x)**:**
for local step $j = 1, \ldots, K$ **do**
$x \leftarrow x - \eta \nabla f(x; z)$ for $z \sim \mathcal{P}_i$
return x to server

Algorithm 1: Federated Averaging (local SGD), when all clients have the same amount of data.



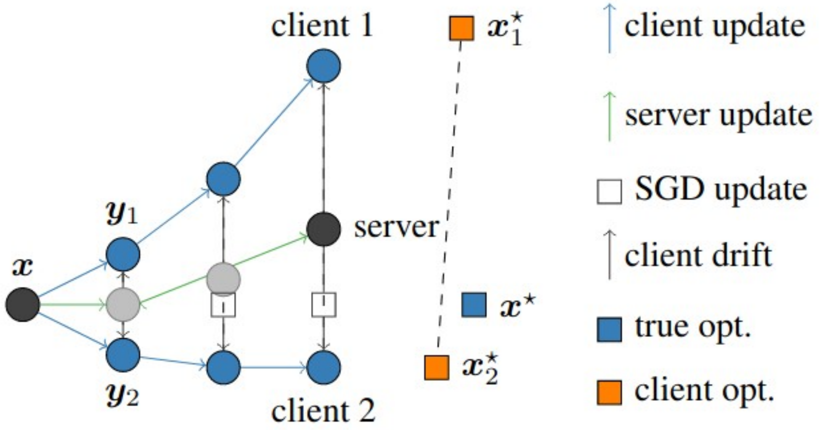
Images/plots taken from paper

About client drift

Scaffold: Stochastic controlled averaging for federated learning
SP Karimireddy, S Kale, M Mohri… - International …, 2020 - proceedings.mlr.press
… a new **Stochastic** Controlled Averaging algorithm (**SCAFFOLD**) which … **SCAFFOLD** is at least as fast as SGD and converges for arbitrarily heterogeneous data. • We show **SCAFFOLD** …

- **clients converge to different solutions…**
- **Scaffold** (requires **stateful** clients)

on client $i \in \mathcal{S}$ **in parallel do**
  initialize local model $\boldsymbol{y}_i \leftarrow \boldsymbol{x}$
  for $k = 1, \ldots, K$ **do**
    compute mini-batch gradient $g_i(\boldsymbol{y}_i)$
    $\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i - \eta_l \left(g_i(\boldsymbol{y}_i) - \boldsymbol{c}_i + \boldsymbol{c}\right)$
  end for
  $\boldsymbol{c}_i^+ \leftarrow$ (i) $g_i(\boldsymbol{x})$, or (ii) $\boldsymbol{c}_i - \boldsymbol{c} + \frac{1}{K\eta_l}(\boldsymbol{x} - \boldsymbol{y}_i)$
  communicate $(\Delta \boldsymbol{y}_i, \Delta \boldsymbol{c}_i) \leftarrow (\boldsymbol{y}_i - \boldsymbol{x}, \boldsymbol{c}_i^+ - \boldsymbol{c}_i)$
  $\boldsymbol{c}_i \leftarrow \boldsymbol{c}_i^+$
end on client



Images/plots taken from paper

FL is particularly vulnerable to attacks

- **Types**
  - evasion attacks (at inference time)
  - poisoning attack (at training time)
- **Byzantine client:** can send arbitrary model updates
- **Defenses:**
  - **Robust aggregation** (median-based, trimmed mean, …)
  - **Data redundancy / shuffling** (not data local…)

Questions?