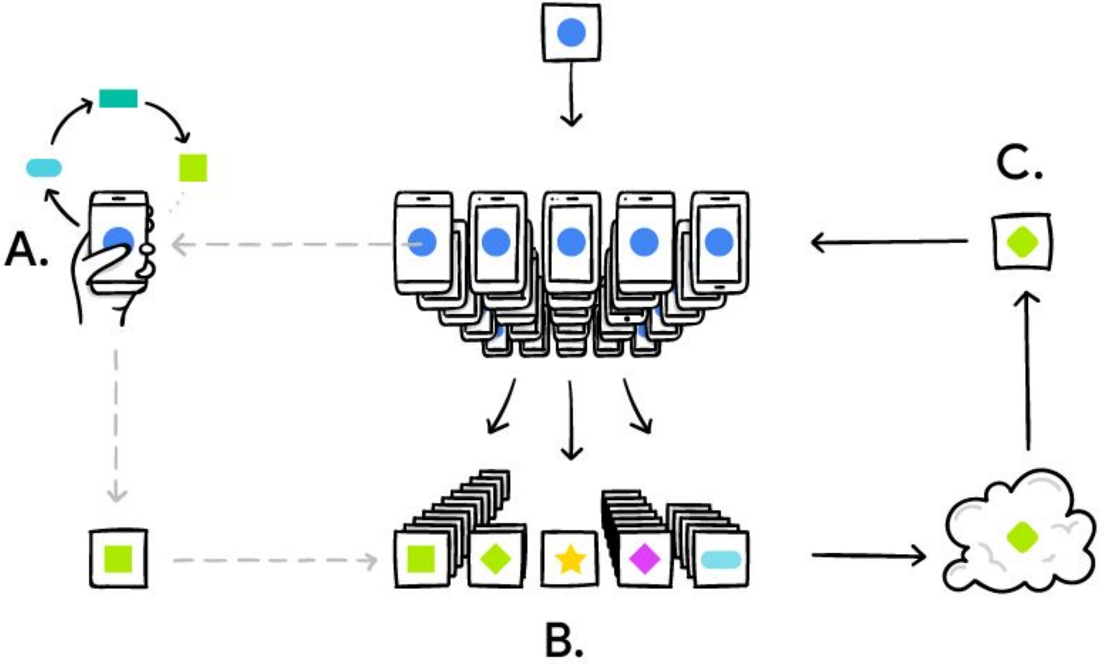
FEDERATED
MACHINE LEARNING

Oh Joon Kwon
(Mentor: Alex Galakatos, Andrew Crotty)

What is Federated ML and Why do we need it?



Problems with Large Scale Machine Learning

Simply scaling machine learning models does not work!

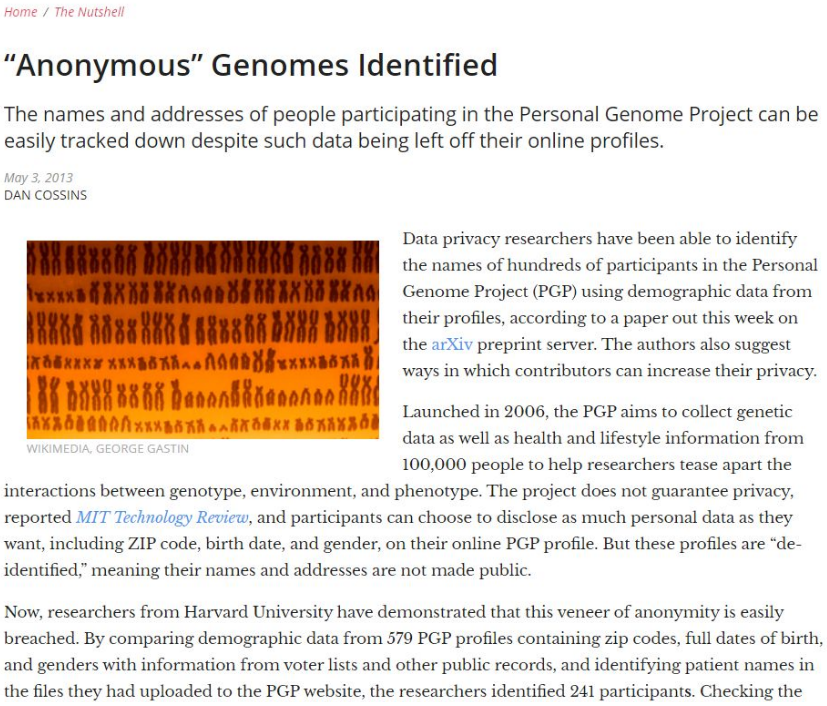
Must consider:

Privacy

Communication and Computation Costs

Unbalanced/Biased Dataset

Data Privacy



Home / The Nutshell

“Anonymous” Genomes Identified

The names and addresses of people participating in the Personal Genome Project can be easily tracked down despite such data being left off their online profiles.

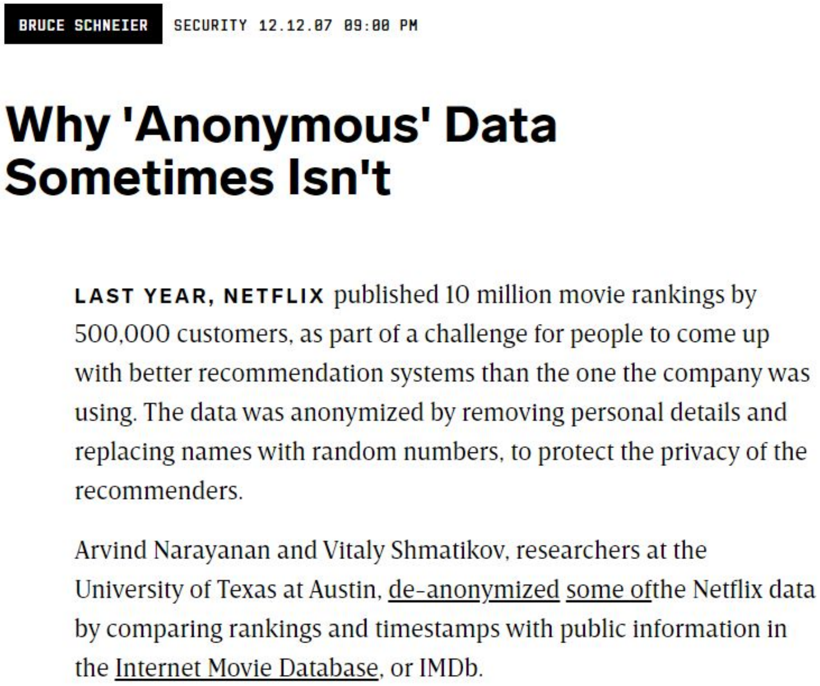
May 3, 2013
DAN COSSINS

WIKIMEDIA, GEORGE GASTIN

Data privacy researchers have been able to identify the names of hundreds of participants in the Personal Genome Project (PGP) using demographic data from their profiles, according to a paper out this week on the arXiv preprint server. The authors also suggest ways in which contributors can increase their privacy.

Launched in 2006, the PGP aims to collect genetic data as well as health and lifestyle information from 100,000 people to help researchers tease apart the interactions between genotype, environment, and phenotype. The project does not guarantee privacy, reported *MIT Technology Review*, and participants can choose to disclose as much personal data as they want, including ZIP code, birth date, and gender, on their online PGP profile. But these profiles are “de-identified,” meaning their names and addresses are not made public.

Now, researchers from Harvard University have demonstrated that this veneer of anonymity is easily breached. By comparing demographic data from 579 PGP profiles containing zip codes, full dates of birth, and genders with information from voter lists and other public records, and identifying patient names in the files they had uploaded to the PGP website, the researchers identified 241 participants. Checking the



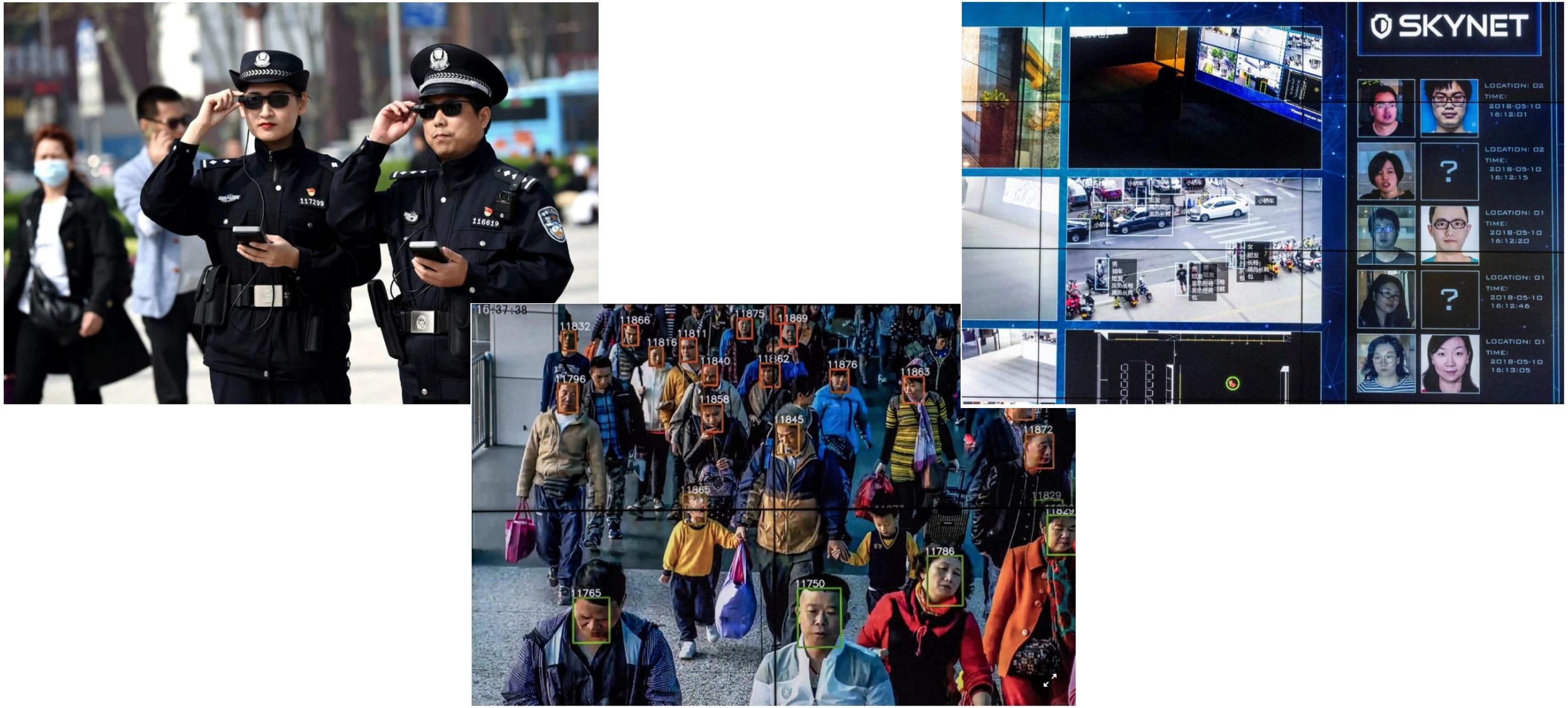
BRUCE SCHNEIER SECURITY 12.12.07 09:00 PM

Why 'Anonymous' Data Sometimes Isn't

LAST YEAR, NETFLIX published 10 million movie rankings by 500,000 customers, as part of a challenge for people to come up with better recommendation systems than the one the company was using. The data was anonymized by removing personal details and replacing names with random numbers, to protect the privacy of the recommenders.

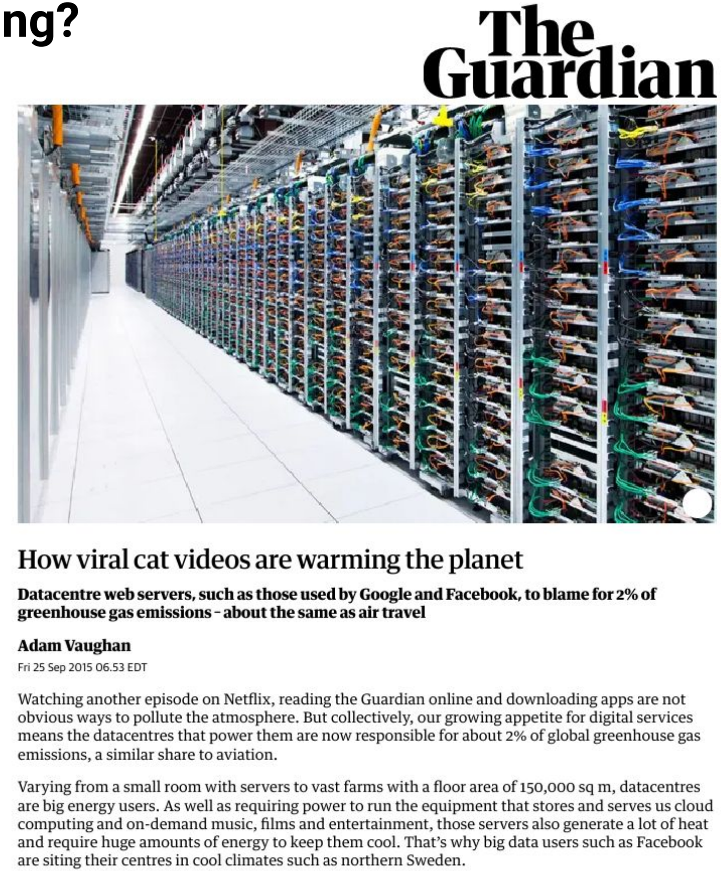
Arvind Narayanan and Vitaly Shmatikov, researchers at the University of Texas at Austin, de-anonymized some ofthe Netflix data by comparing rankings and timestamps with public information in the Internet Movie Database, or IMDb.

https://www.the-scientist.com/the-nutshell/anonymous-genomes-identified-39362
https://www.wired.com/2007/12/why-anonymous-data-sometimes-isnt/



https://www.nytimes.com/2018/07/08/business/china-surveillance-technology.html?auth=login-email&login=email

How about global warming?



The Guardian

How viral cat videos are warming the planet

Datacentre web servers, such as those used by Google and Facebook, to blame for 2% of greenhouse gas emissions – about the same as air travel

Adam Vaughan

Fri 25 Sep 2015 06.53 EDT

Watching another episode on Netflix, reading the Guardian online and downloading apps are not obvious ways to pollute the atmosphere. But collectively, our growing appetite for digital services means the datacentres that power them are now responsible for about 2% of global greenhouse gas emissions, a similar share to aviation.

Varying from a small room with servers to vast farms with a floor area of 150,000 sq m, datacentres are big energy users. As well as requiring power to run the equipment that stores and serves us cloud computing and on-demand music, films and entertainment, those servers also generate a lot of heat and require huge amounts of energy to keep them cool. That's why big data users such as Facebook are siting their centres in cool climates such as northern Sweden.

Possible Solutions for Privacy Concerns

Encryption (Centralized)

Homomorphic encryption - operates on encrypted data

Most secure, but expensive to compute

Suitable for statistical computation

Federated Learning (Decentralized)

Send trained model parameters to server

No direct exposure of private data

Problem Setup

Finite Sum Problem (for any ML)

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{\theta})$$

“ Minimize the mean loss on given data with weights ”

Federated Optimization Problem

We now have to consider:

Non-IID Datasets

Unbalanced

Massively Distributed

Limited Communication

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} f(\boldsymbol{\theta}) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(\boldsymbol{\theta}) \qquad F_k(\boldsymbol{\theta}) := \frac{1}{n_k} \sum_{i \in S_k} f_i(\boldsymbol{\theta}) \qquad n = \sum_{k=1}^{K} n_k \qquad |C_t| \gg \sum_{k=1}^{K} \frac{n_k}{K}$$

“ Minimize the weighted sum of mean loss of random set of clients ”

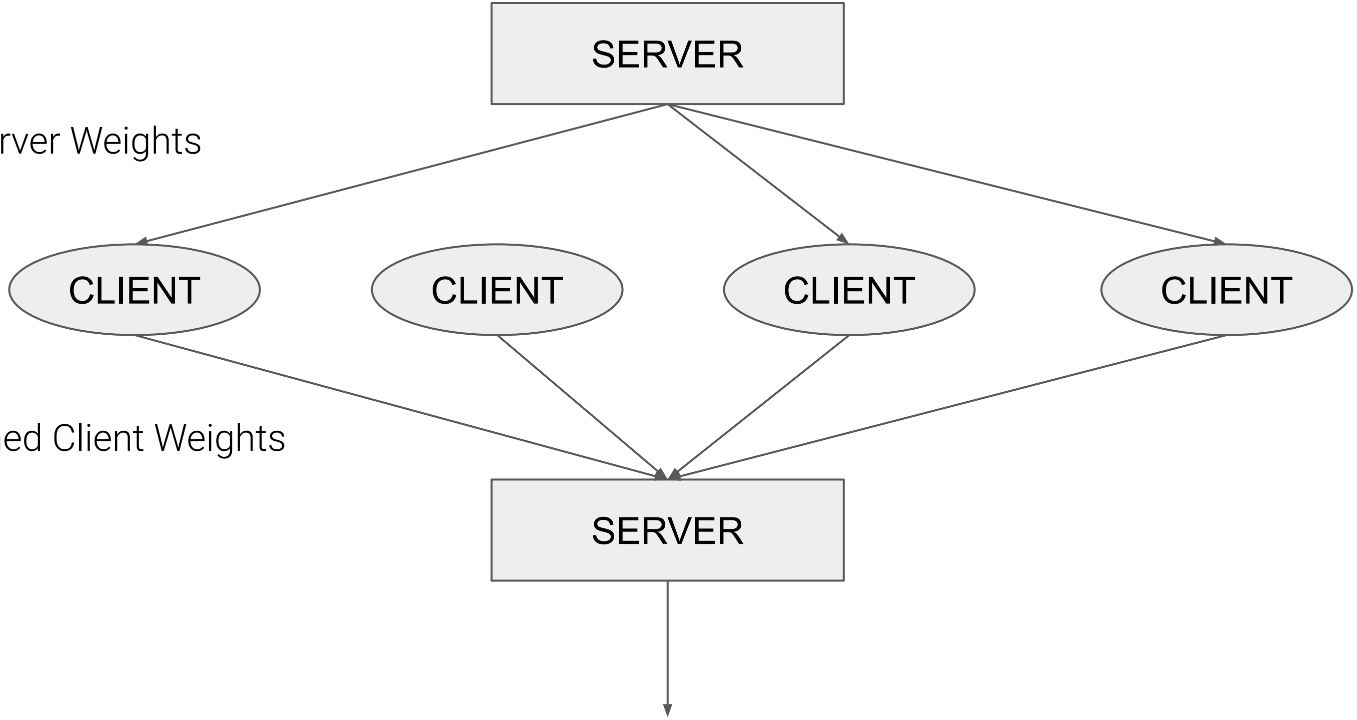
Time

1. Broadcast Server Weights

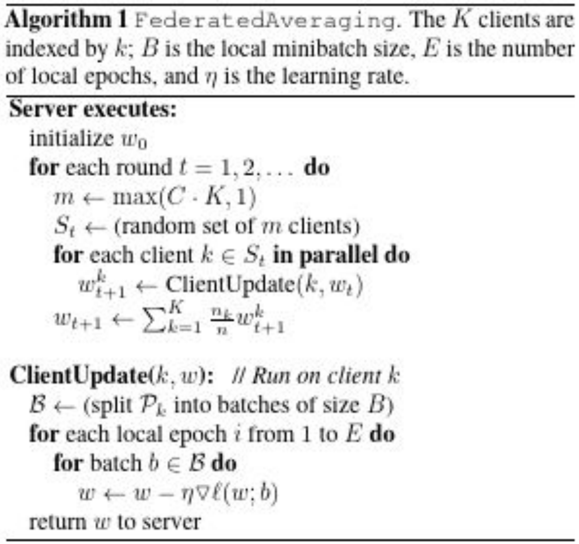
2. Train

3. Aggregate Trained Client Weights

4. Repeat



Federated Averaging and FedSGD



Algorithm 1 FederatedAveraging. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

```
Server executes:
  initialize $w_0$
  for each round $t = 1, 2, \dots$ do
    $m \leftarrow \max(C \cdot K, 1)$
    $S_t \leftarrow$ (random set of $m$ clients)
    for each client $k \in S_t$ in parallel do
      $w_{t+1}^k \leftarrow$ ClientUpdate$(k, w_t)$
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate$(k, w)$:   // Run on client $k$
  $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size $B$)
  for each local epoch $i$ from 1 to $E$ do
    for batch $b \in \mathcal{B}$ do
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return $w$ to server
```

Federated Averaging (FedAvg)

Shares updated parameters

Federated SGD (FedSGD)

Shares local gradients

Baseline algorithm for FedAvg

Special case of FedAvg:
Single local batch ($B = \infty$)
Single local epoch ($E = 1$)

https://arxiv.org/pdf/1602.05629.pdfz

Federated Averaging Demo

https://colab.research.google.com/drive/1p98m12ID-czEL2WyJSN1YI2tTExz71H3

NOTE: FedAvg can be applied to any ML models! (not just deep learning)

Convergence Rate on Non-IID (w.r.t. Local Epochs) :

$$\mathcal{O}\left(\frac{1}{E}\right)$$

https://arxiv.org/pdf/1907.02189.pdf

Federated Averaging is Not Perfect!

No huge improvement on client computational costs (for big model)

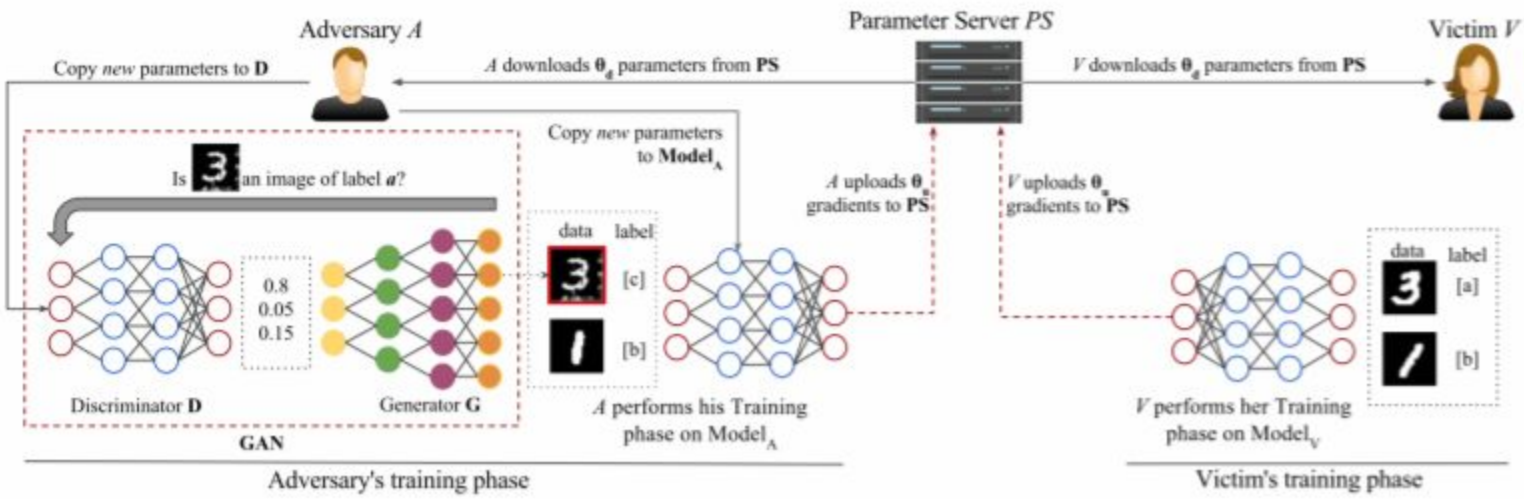
Information Leakage from shared weights

Imbalance in contribution of individual clients

- Some clients have more to gain from their share of contributions

Information Leakage from Collaborative Learning

Possible Attacks on Collaborative Learning Models



Adversary *A*
Copy *new* parameters to **D**
A downloads θ_d parameters from **PS**
Parameter Server *PS*
V downloads θ_d parameters from **PS**
Victim *V*
Copy *new* parameters to **Model**$_A$
Is an image of label *a*?
A uploads θ_u gradients to **PS**
V uploads θ_u gradients to **PS**
0.8
0.05
0.15
data
label
[c]
[b]
data
label
[a]
[b]
Discriminator **D**
Generator **G**
A performs his Training phase on Model$_A$
V performs her Training phase on Model$_V$
GAN
Adversary's training phase
Victim's training phase

https://arxiv.org/pdf/1702.07464.pdf

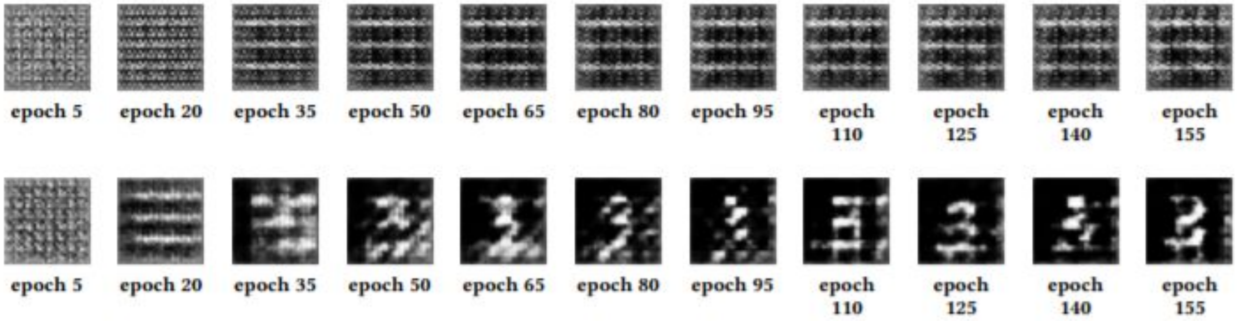


Figure 10: DCGAN with No influence vs. influence in Collaborative Learning for 3 (Three)

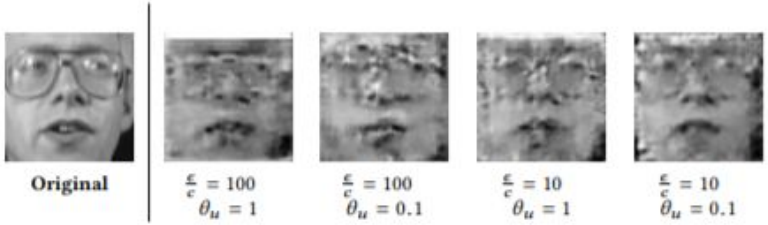
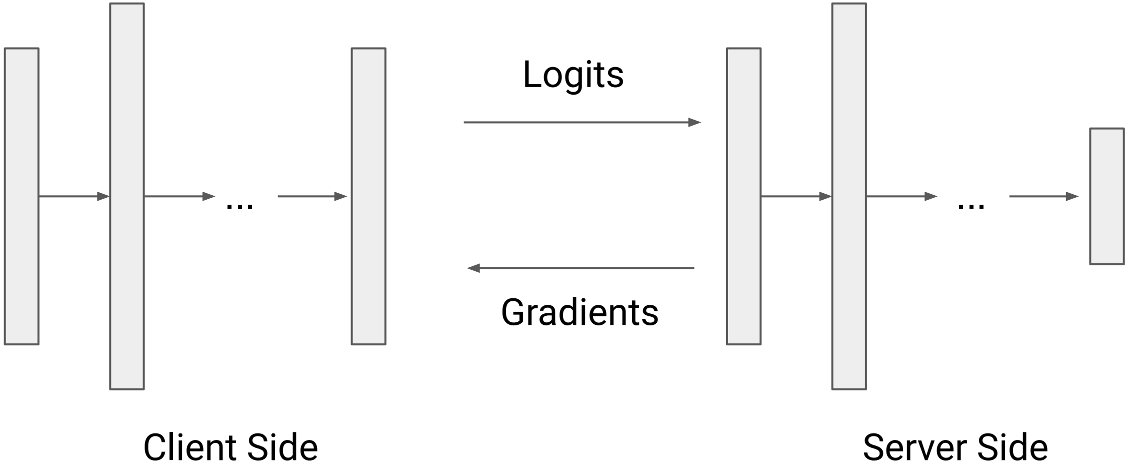

Original
$\frac{\epsilon}{c} = 100$ $\theta_u = 1$
$\frac{\epsilon}{c} = 100$ $\theta_u = 0.1$
$\frac{\epsilon}{c} = 10$ $\theta_u = 1$
$\frac{\epsilon}{c} = 10$ $\theta_u = 0.1$

Figure 11: Experimental results on the AT&T Dataset with 100% download (($\theta_d = 1$) and DP enabled. Unlike MNIST, images are noisier because this particular dataset is small and the accuracy of the model is significantly affected when upload rates are small.

Split Neural Networks (SplitNN)

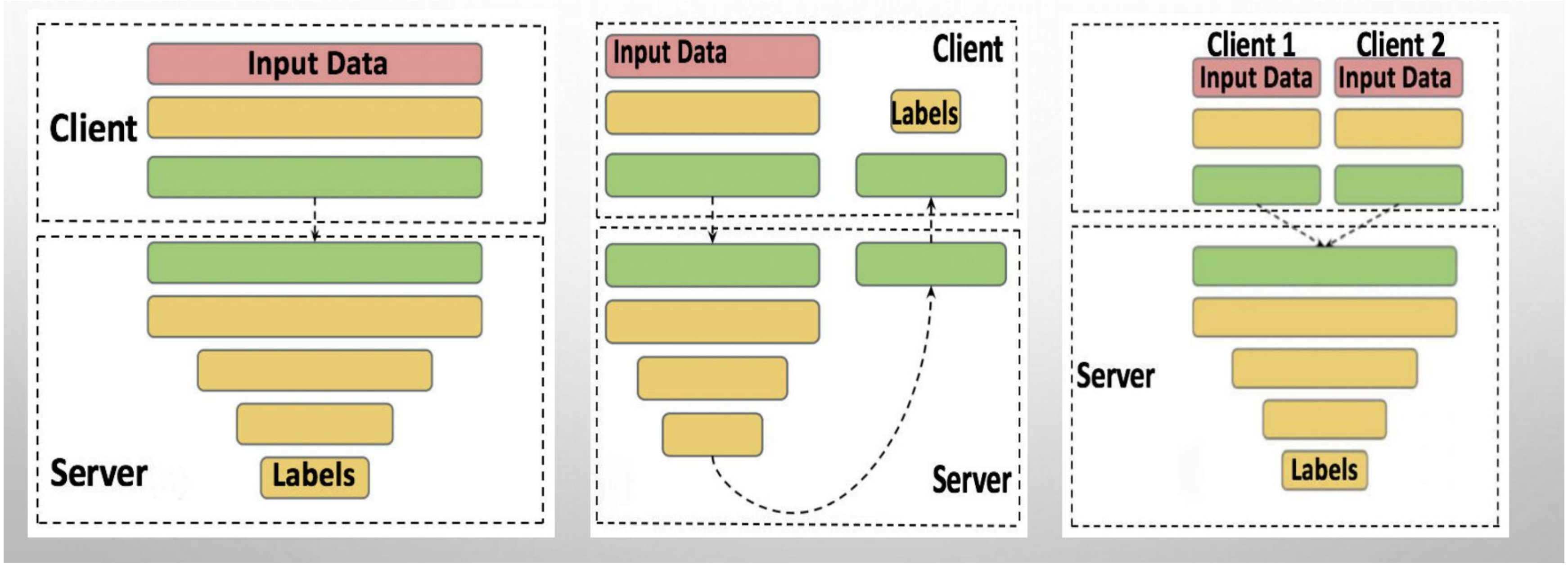
New Federated Learning Method (Split Learning)

Claims to have achieved lower computation costs and bandwidth, especially in large scale



https://arxiv.org/pdf/1812.00564.pdf

Configurations



Things to Consider

Problem : A client may not want to share labels to others (other clients, server)

- Sharing labels can lead to information leakage
- Which “metric” to optimize for training?
    - Constrained Covariance
    - Distance Correlation
    - Max Mean Discrepancy
    - Kernel Target Alignment
    - Maximal Information Coefficient

SplitNN Demo

https://colab.research.google.com/drive/1GG5HctuRoaQF1Yp6ko3WrJFtCDoQxT5_#scrollTo=esvT5OgzG6Fd

- Replication Demo for Vanilla Configuration (Label Sharing)

Future Works

- Completely Decentralized Learning Scheme
  - No need for Central Authority (Server)
- Room for Improvement in Computation/Communication Costs
- Reduction of Information Leakage
- Better published frameworks for easy implementation
- Game Theory Approach for Analysis
  - Balancing individual contributions

Github Link for Source Code

https://github.com/pulpiction/APMA-DRP-Project