Federated Learning

Communication-Efficient Learning of Deep Learning Networks from Decentralized Data (AISTATS 2017)

What is Federated Learning?

Collaborative Machine Learning without Centralized Training Data

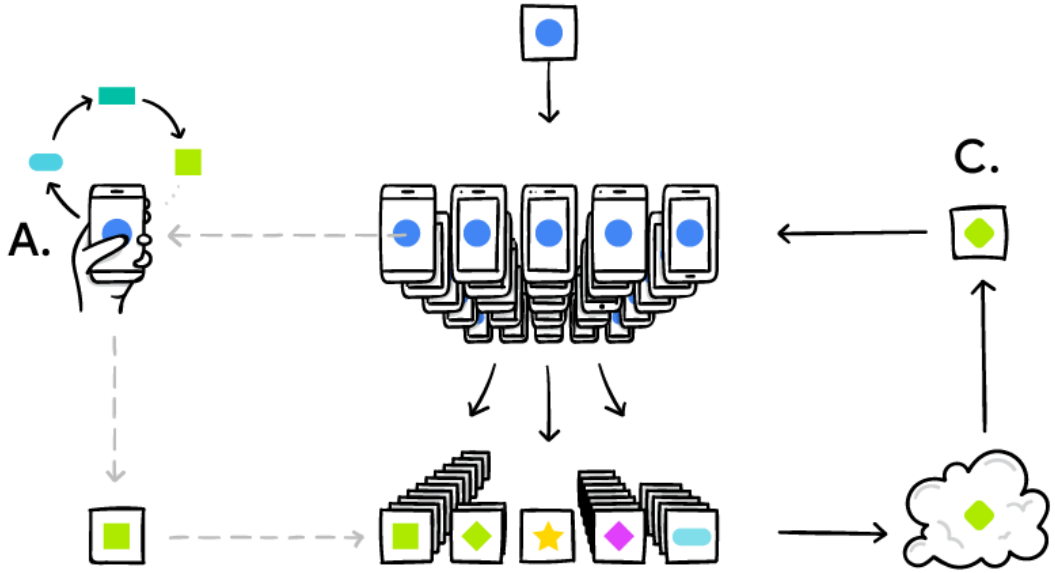
Standard approach:

- Machine Learning approaches require **centralizing the training data** on one machine or in a datacenter
- Needs of **secure and robust cloud infrastructures** for processing this data
- **No user interaction**

Federated Learning

It **enables**:

- Mobile devices to **collaboratively learn a shared prediction**
- **All training data are on the device**
- Local methods that are locally trained: instead, **the model trains on the device (*client*) as well**

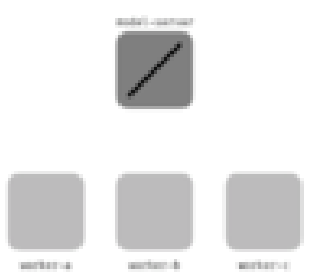
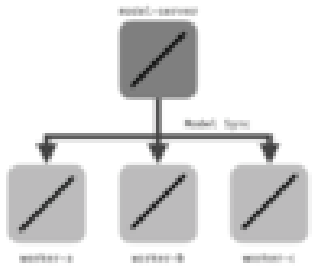
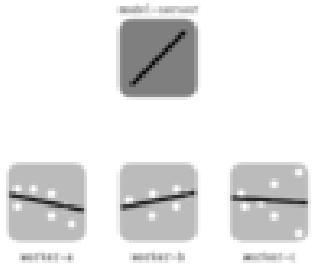
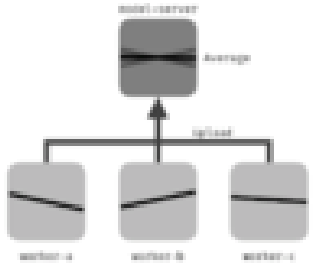


https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

Federated Learning

- Settings:
  - Centralized:
    - Central server: organization and coordinate all the participating nodes during the training
    - The server is responsible for the nodes selection and for the aggregation
    - The server may become a bottleneck of the system
  - Decentralized:
    - The nodes are able to coordinate themselves to obtain the global model
    - This setup prevents single point failures
  - Heterogeneous:
    - Clients equipped with very different computation and communication capabilities

Centralized Scenario

Step 1	Step 2	Step 3	Step 4
			
Central server chooses a statistical model to be trained	Central server transmits the initial model to several nodes	Nodes train the model locally with their own data	Central server pools model results and generate one global mode without accessing any data

Federated Learning

Pros:

- **Personalized** models
- Mobile devices are **widespread** (high number of devices and data!)
  - Fast processors (including GPUs)
- **Privacy compliant**

Applications:

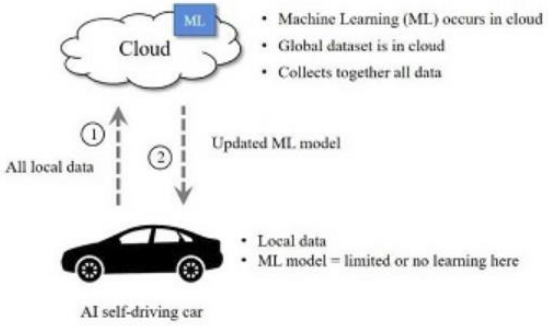
- Language models can improve speech recognition and text entry from the user
- Image models can select good photos taken by users

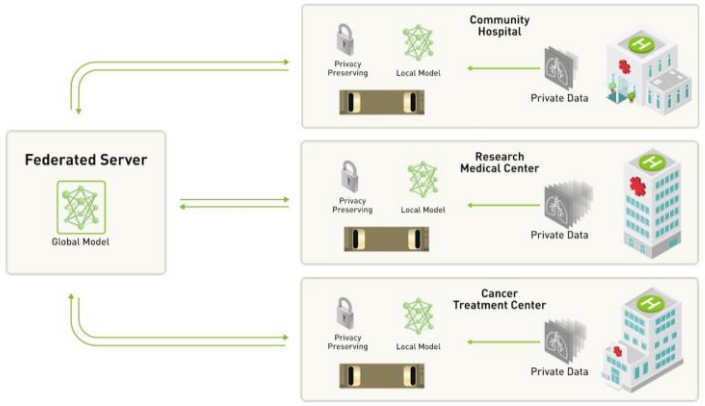
Federated Learning

- Cons & problems
  - Communication costs >> computation costs
  - Non-IID: local data are not representative of the whole population
  - Unbalanced: local data can be unbalanced
  - Limited Communication: mobile devices are frequently offline or on slow conn.
  - Massively Distribuited: #clients >> avg sample data per client

Use Cases of Federated Learning

- **Transportation**: self-driving cars
  - Limit the volume of data transfer
  - Personalized training for models
- **Industry 4.0**: smart manufacturing
  - Guarantee the privacy
- **Medicine**: digital health
  - Improve patient privacy
  - Data Protection





https://en.wikipedia.org/wiki/Federated_learning

Federated SGD (FedSGD)

- **SGD can be applied natively to FL** optimization problem
  - A single batch gradient computation is done per round of communication
  - Gradients are averaged by the server proportionally to the number of training samples on each node,
  - Mean gradient is used to make a gradient descent step

- This approach:
  - **Computationally efficient**
  - **Requires a very large number of rounds**

FederatedAveraging SGD (FedAVG)

- FedAVG is a generalization of FedSGD
- Differences:
  - Allows local nodes to perform more than one batch update on local data
  - Exchanges the updated weights rather than the gradients

- If all local nodes start from the **same initialization, averaging the gradients is strictly equivalent to averaging the weights themselves**