Federated Learning

New statistical challenges from decentralized data

Statistics, Machine Learning, Deep Learning

CPU/GPU

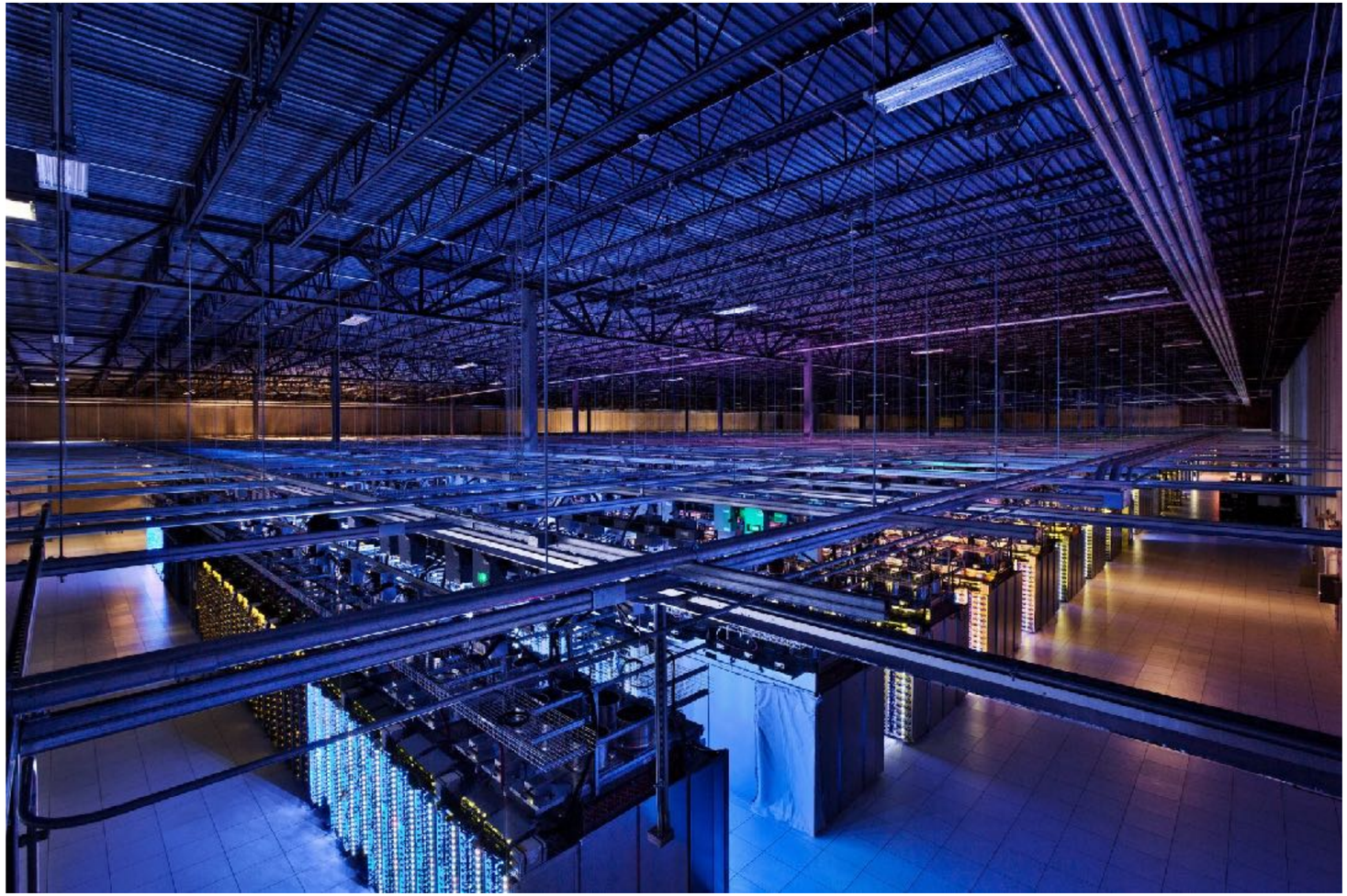
Memory

Google Search I'm Feeling Lucky

Google offered in: 한국어

Large-scale Computing

Large-scale Computing



Collect  **Analyze**

Collect

Systems setting

Analyze

Large-scale Computing

Machines fail

One server may stay up 3 years (1000 days)

Large-scale Computing

Machines fail

One server may stay up 3 years (1000 days)

If you have 1000 servers, expect to loose 1/day

Large-scale Computing

Machines fail

One server may stay up 3 years (1000 days)

If you have 1000 servers, expect to loose 1/day

People estimated Google had ~1M machines in 2011

Large-scale Computing

Machines fail

One server may stay up 3 years (1000 days)

If you have 1000 servers, expect to loose 1/day

People estimated Google had ~1M machines in 2011

1000 machines fail every day!

“Distributed Optimization Problem”

“Distributed Optimization Problem”

Communication Bottleneck

CoCoA

How can we address the communication bottleneck ?

Primal Problem

$$\min_{w \in \mathbb{R}^d} \left[P(\mathbf{w}) := \frac{\lambda}{2} ||w||^2 + \frac{1}{N} \sum_{i=1}^{N} f_i(\mathbf{w}^\top \mathbf{x}_i) \right]$$

Support Vector Machine

Regularized linear and logistic regression

Ordinal regression

$\geq$

Dual Problem

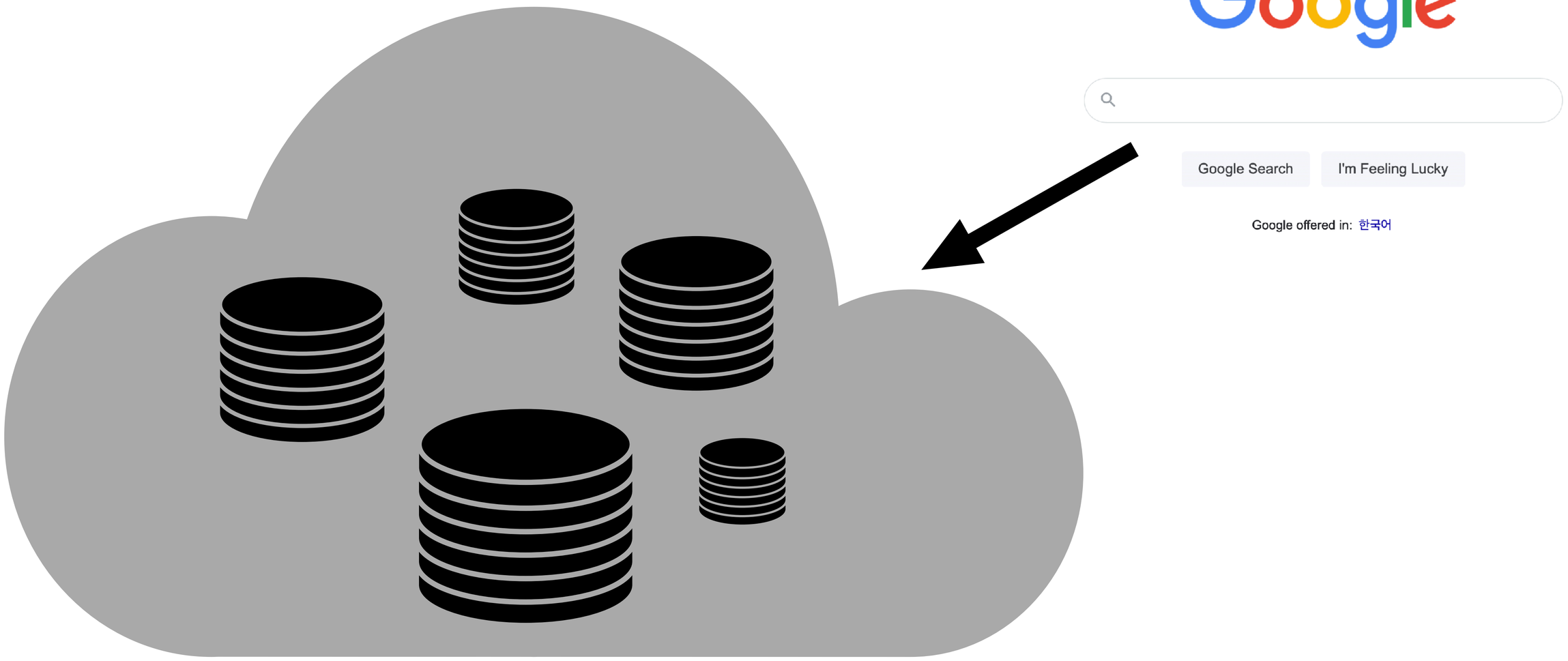
$$\max_{\alpha \in \mathbb{R}^N} \left[D(\alpha) := -\frac{\lambda}{2} ||A\alpha||^2 - \frac{1}{N} \sum_{i=1}^{N} f_i^*(-\alpha_i) \right]$$

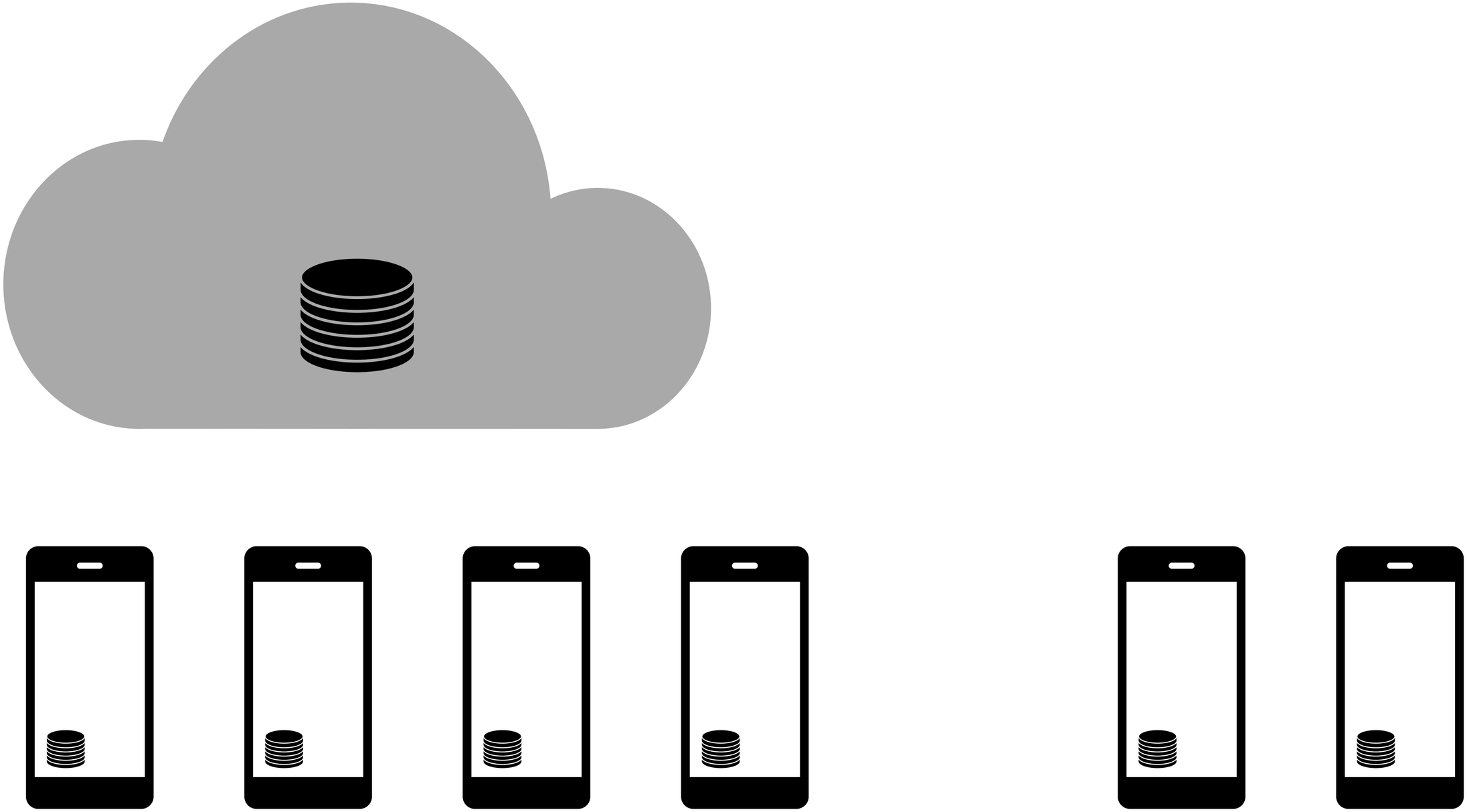
$$A_i := \frac{1}{\lambda n} \mathbf{x}_i$$

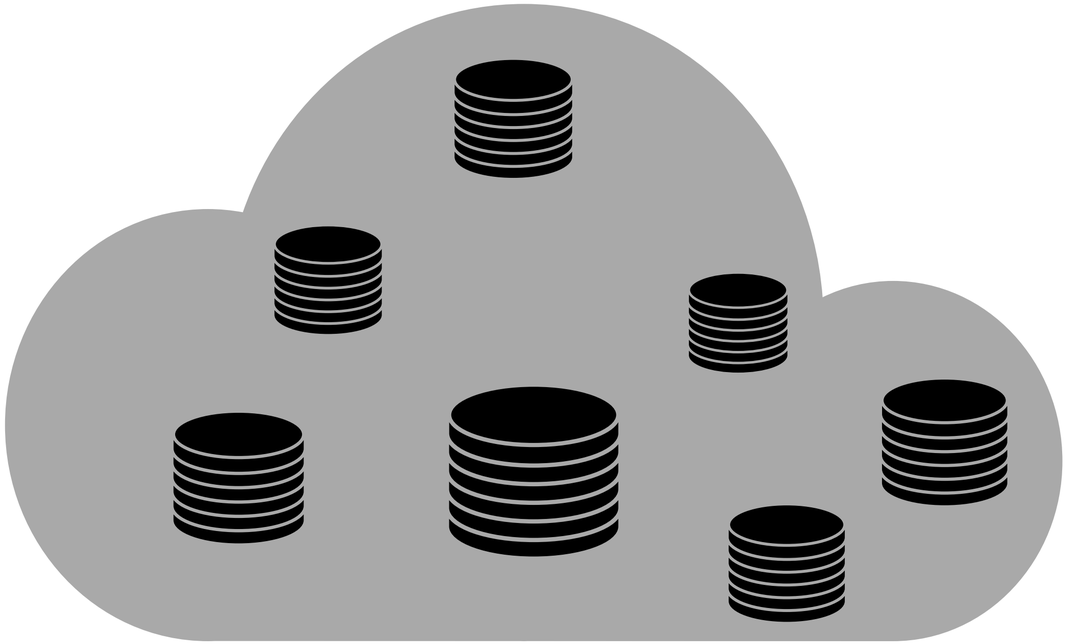
- Stopping criteria given by duality gap
- Good performance in practice

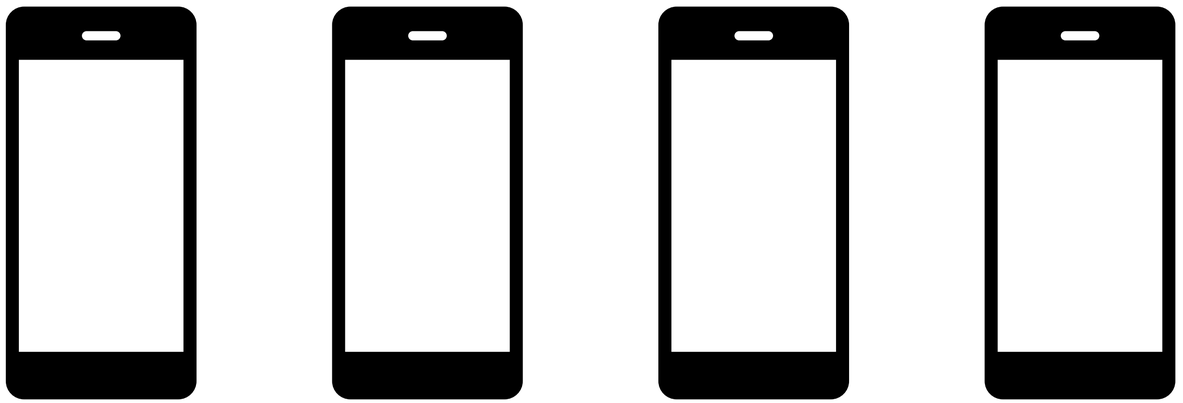
$f^*(y) = \sup_{x \in dom f} \left(y^\top x - f(x) \right)$ **is the conjugate of** f

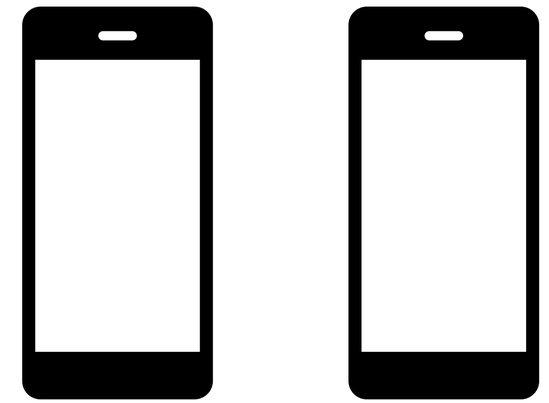
Communication-Efficient Distributed Dual Coordinate Ascent, NIPS, 2014

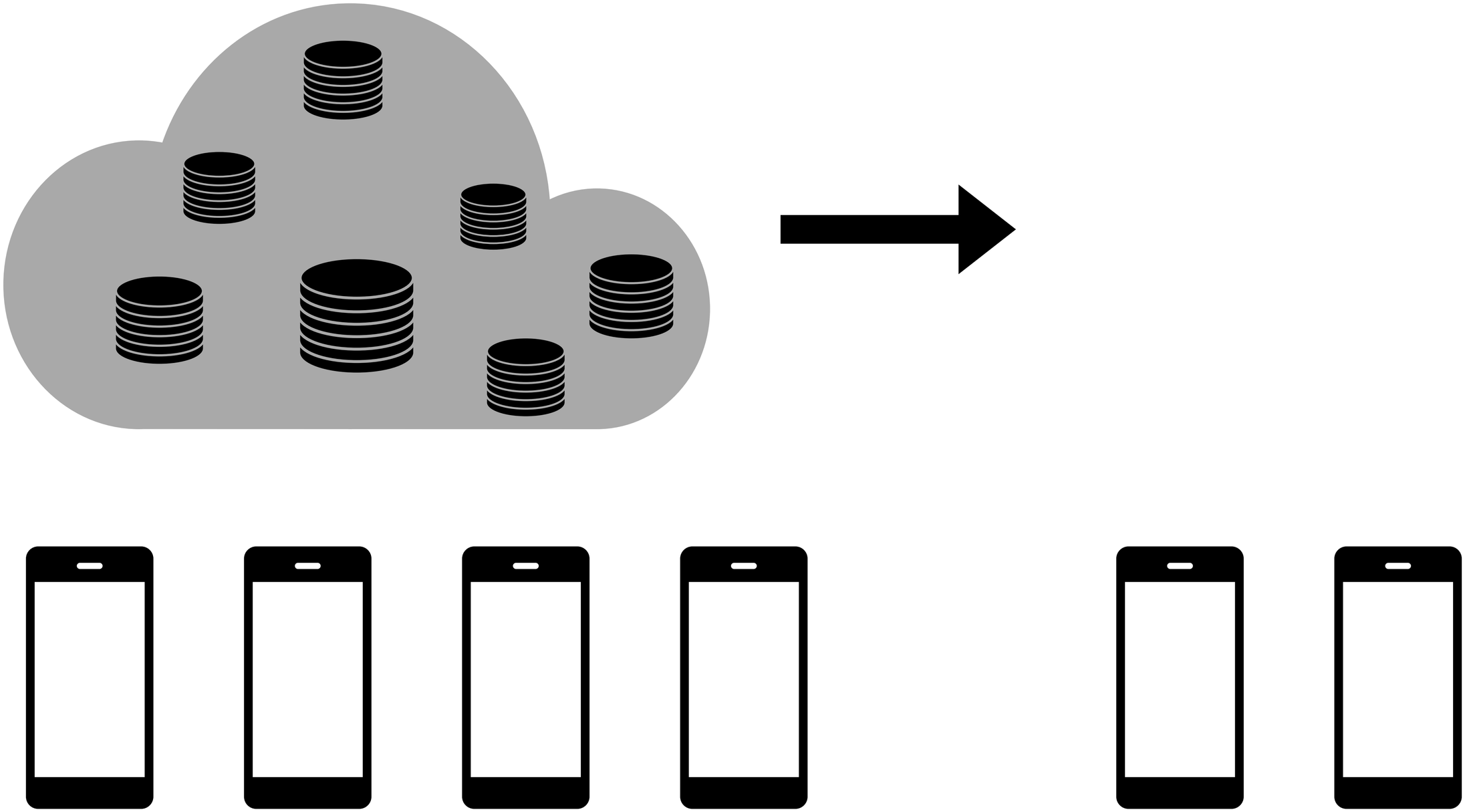

Google
Google Search
I'm Feeling Lucky
Google offered in: 한국어

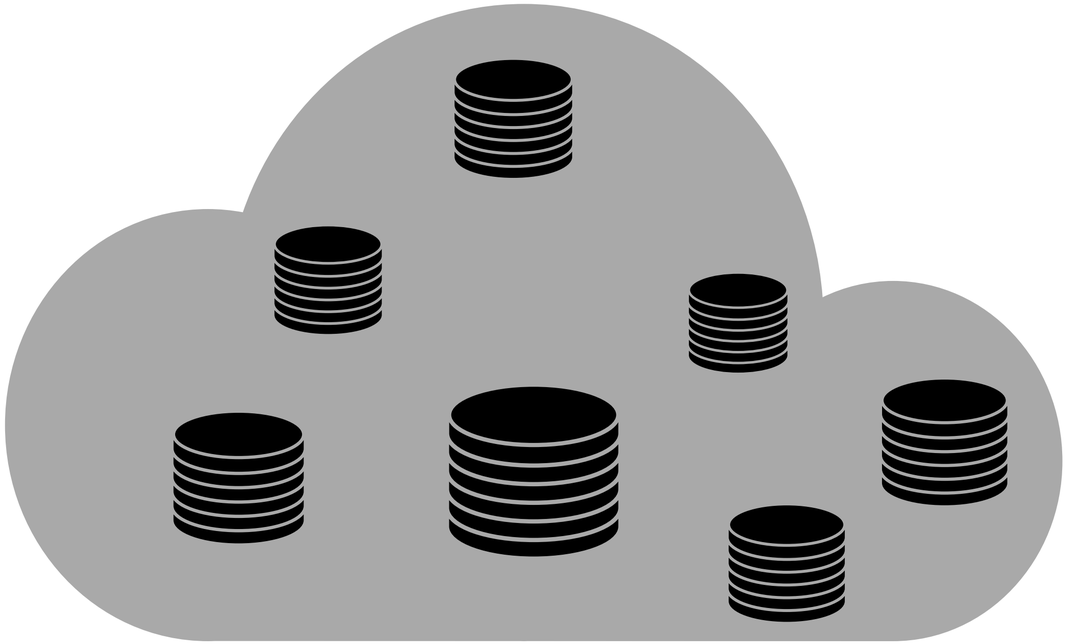


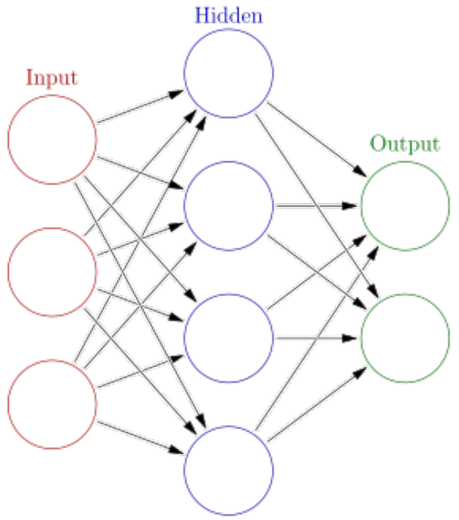


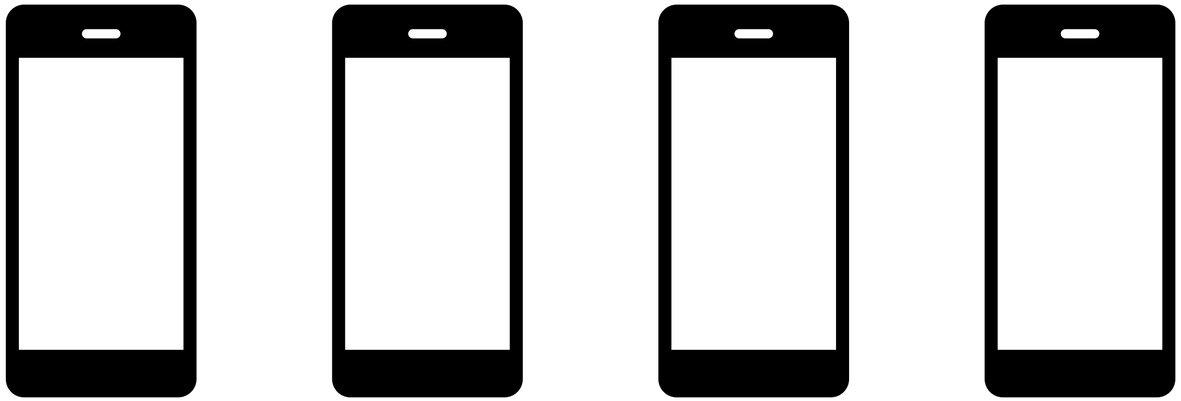


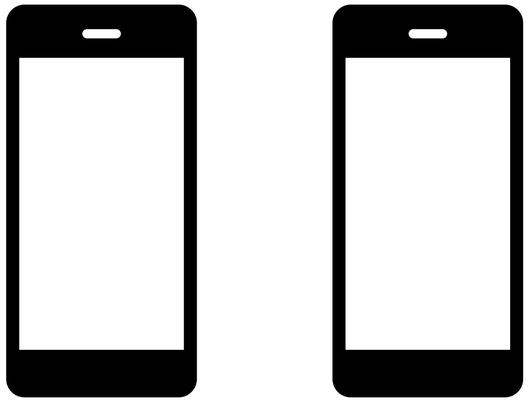


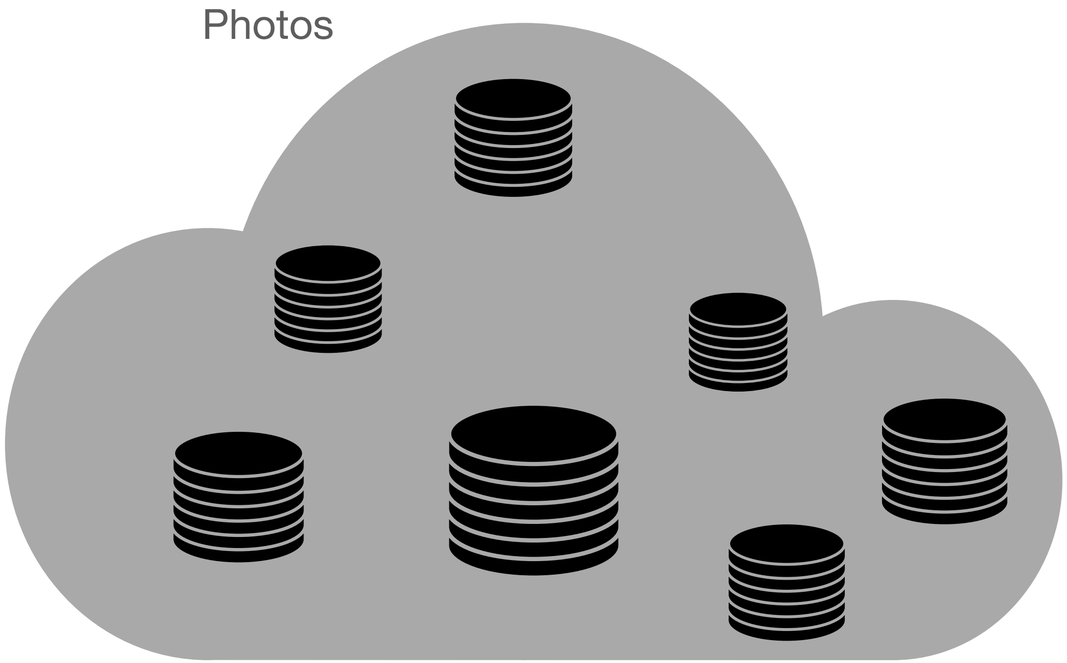


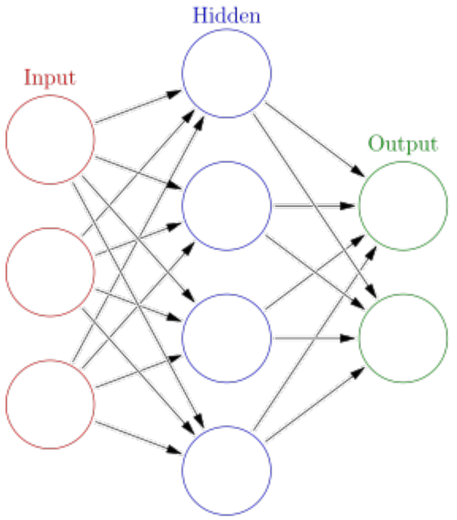


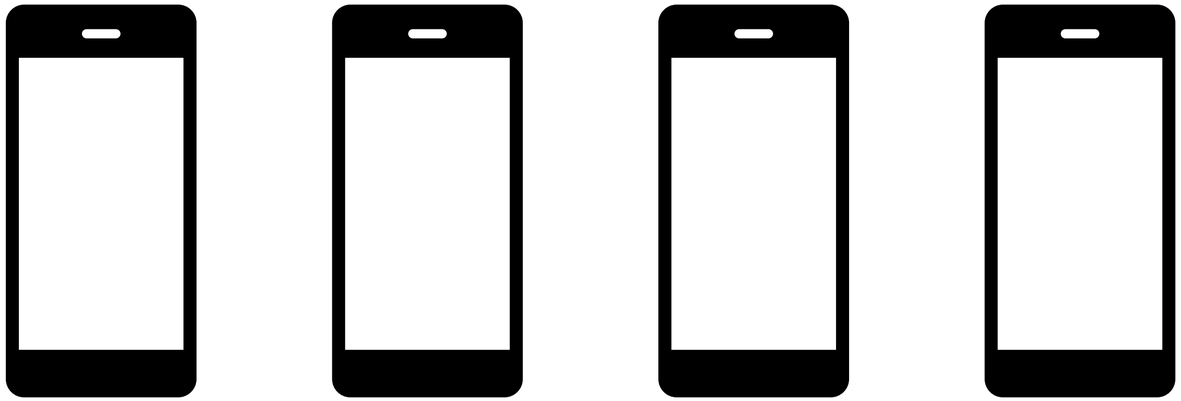

Input
Hidden
Output

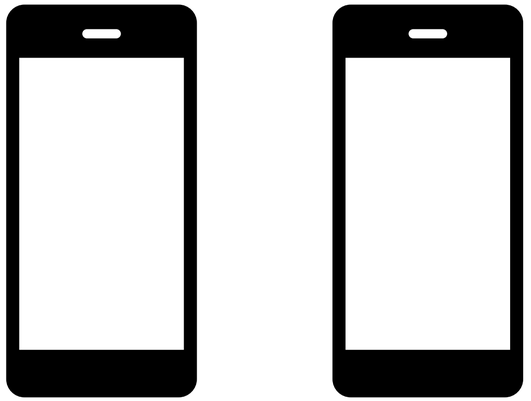


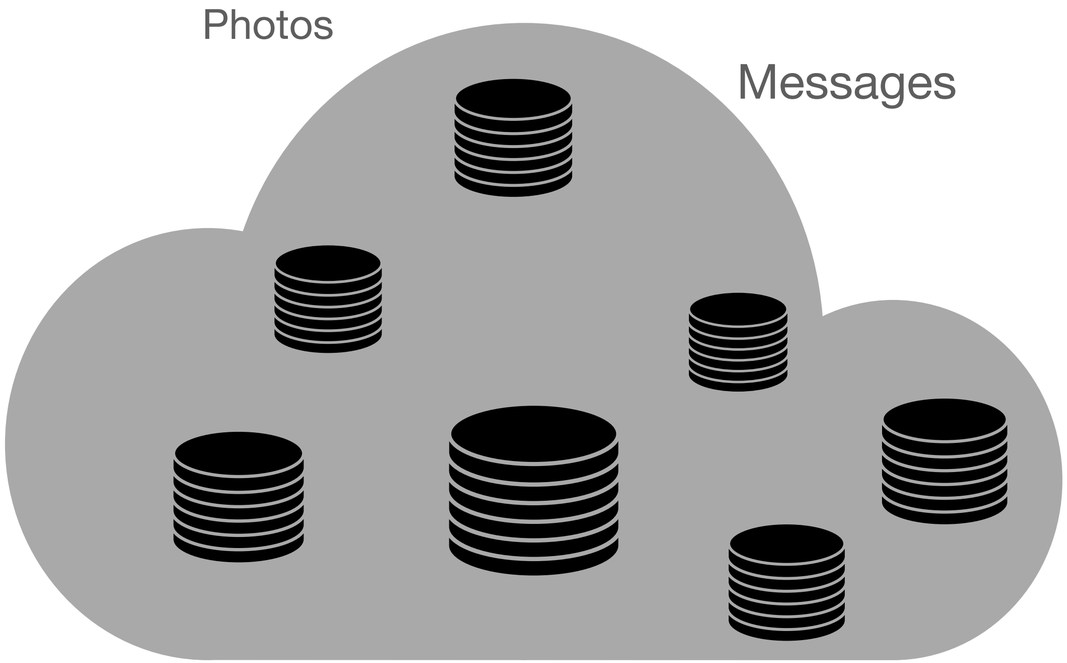


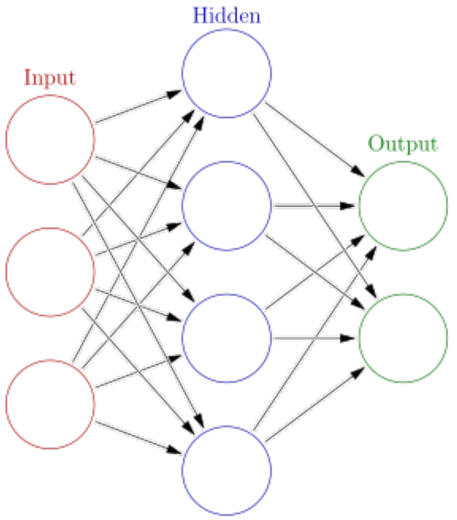

Photos

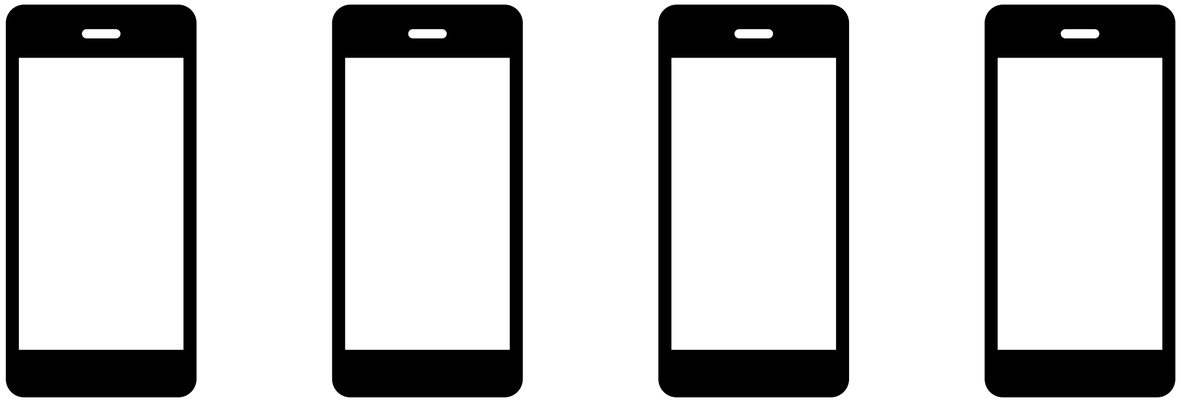

Input
Hidden
Output

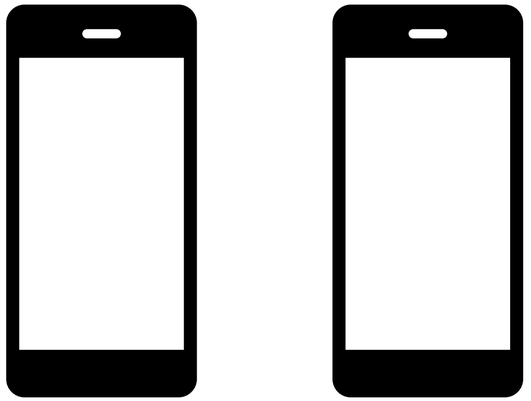


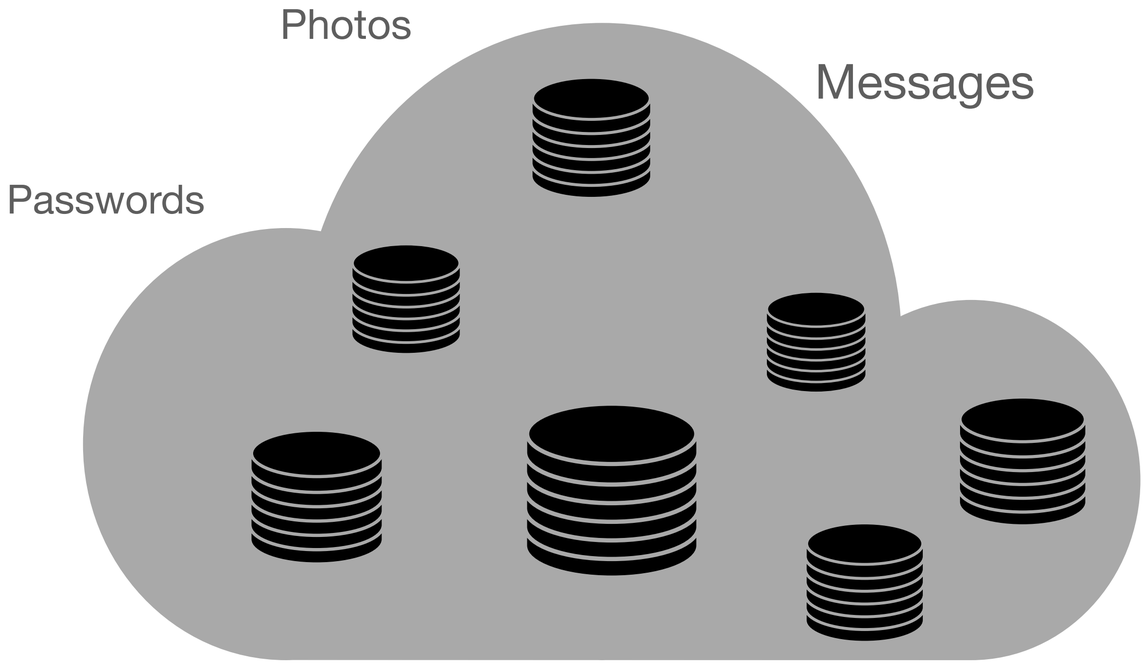


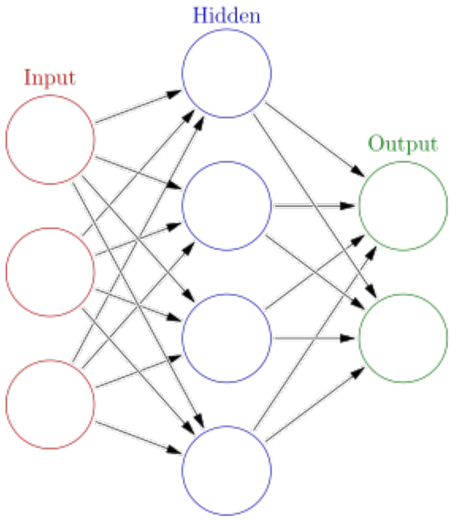

Photos
Messages

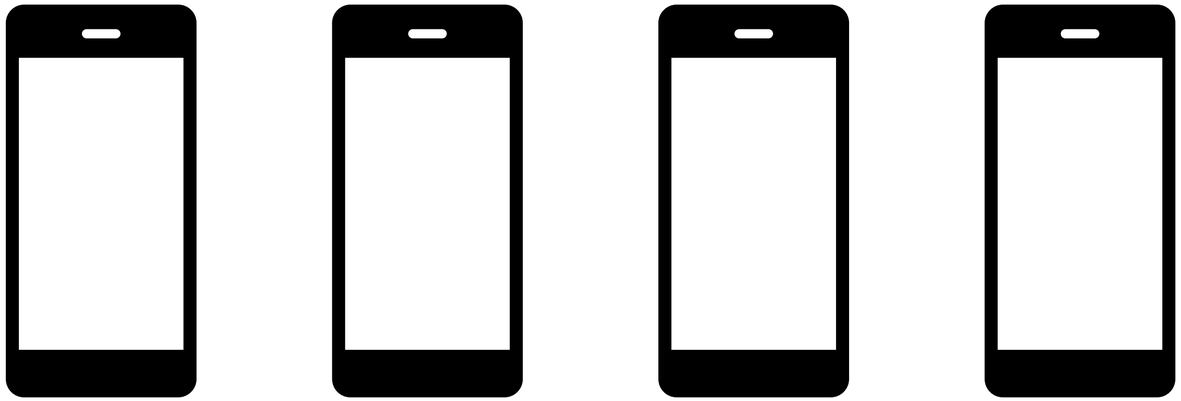

Input
Hidden
Output

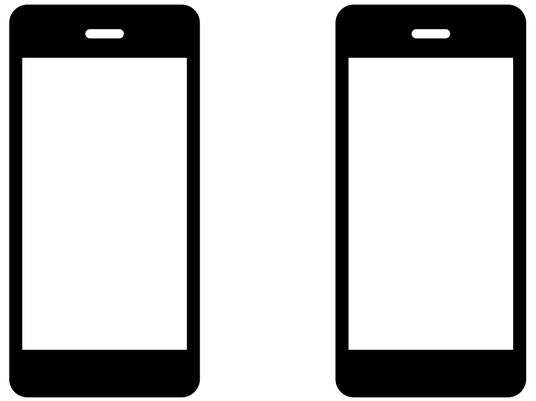


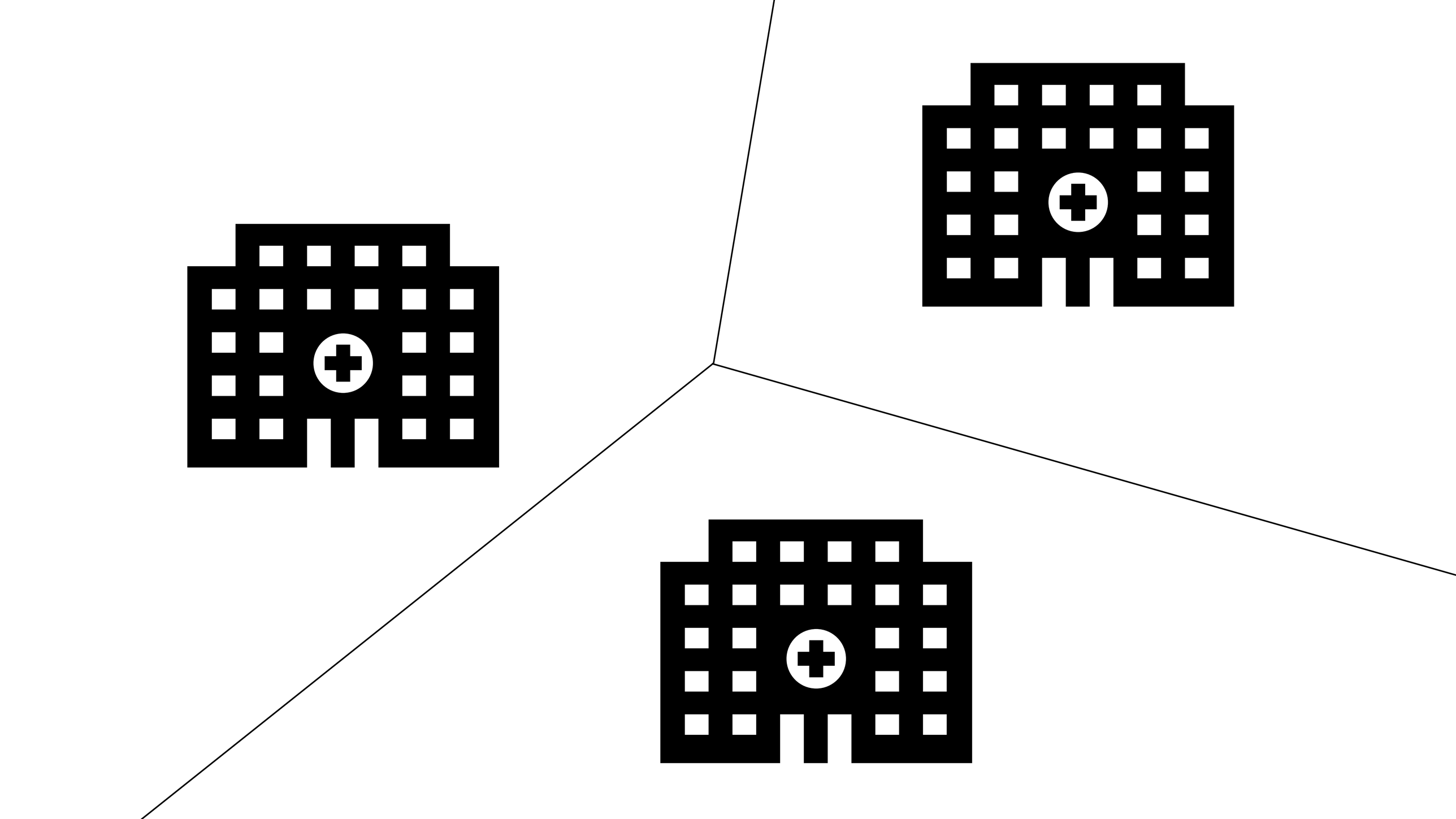


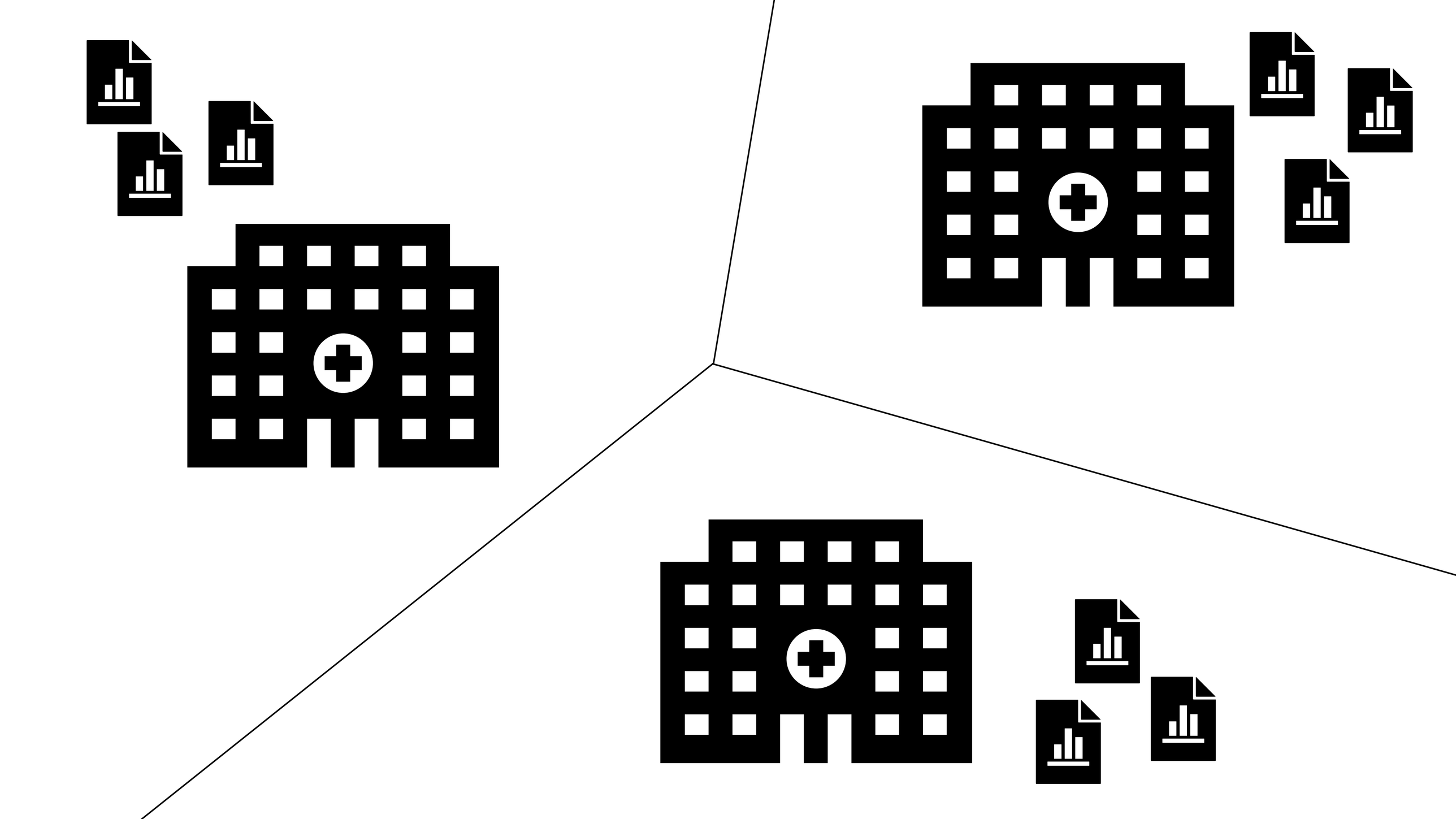

Photos
Messages
Passwords


Hidden
Input
Output





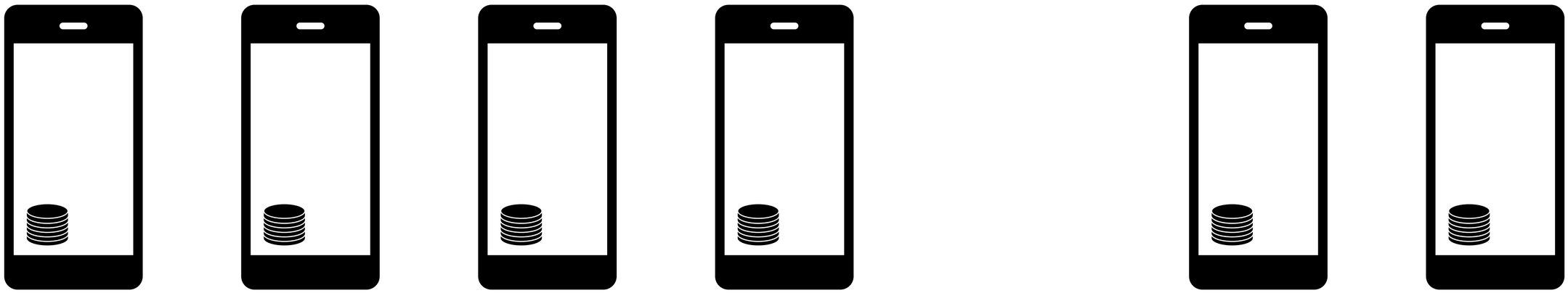




Federated Learning

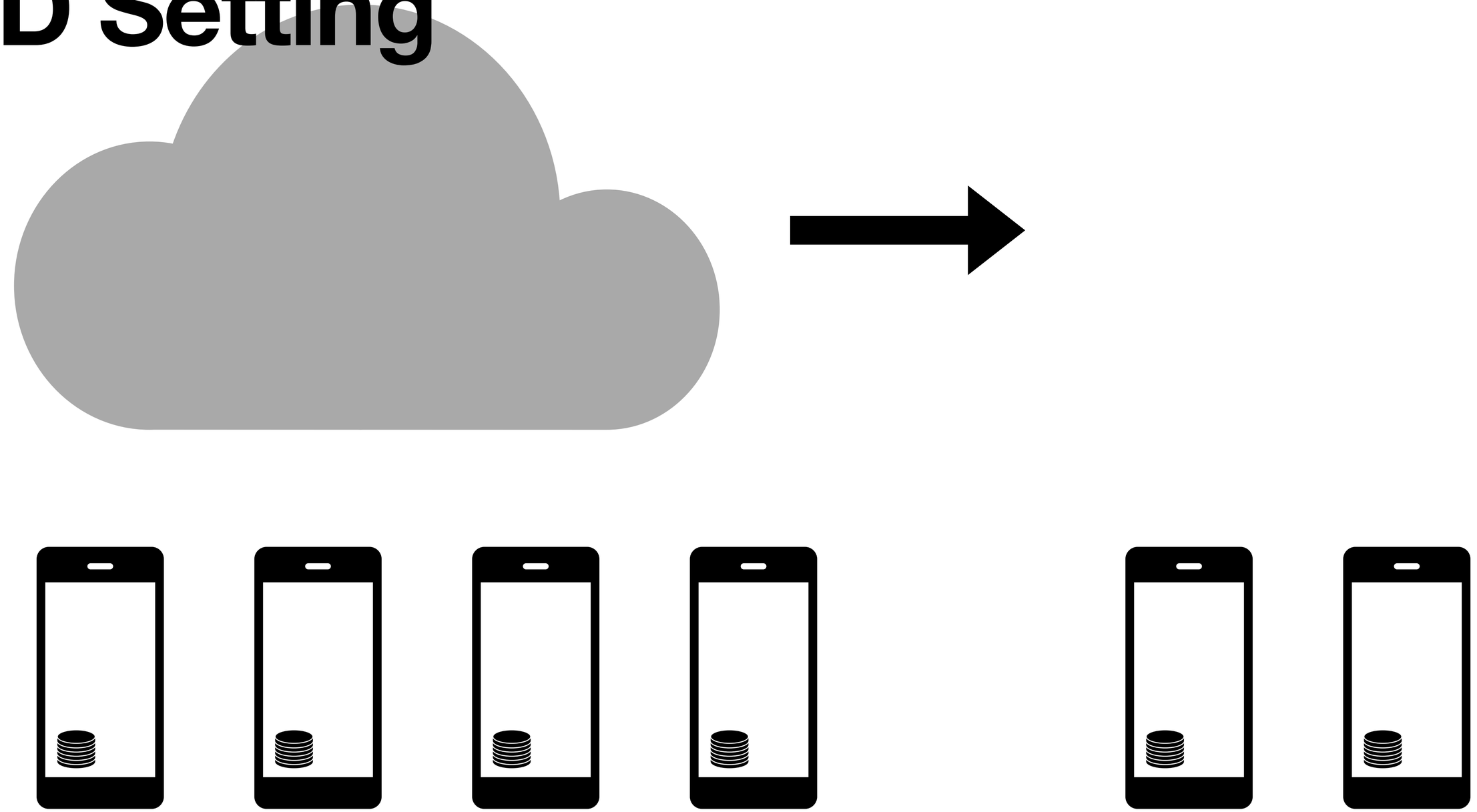
How can we use the data while protecting *Privacy* ?

IID Setting



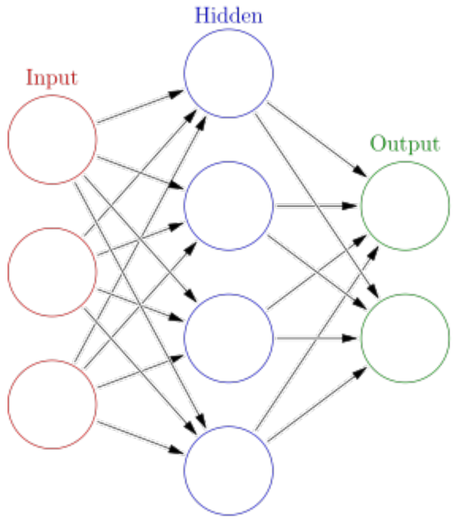
Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

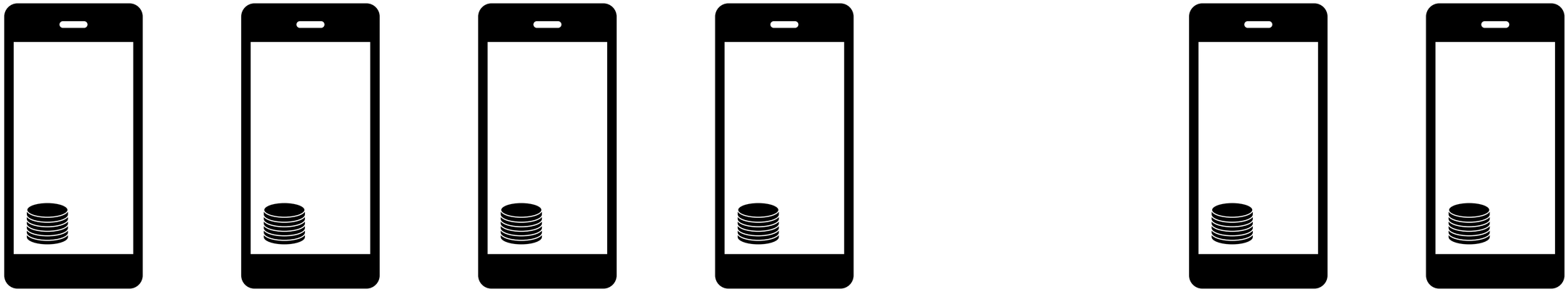
IID Setting



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

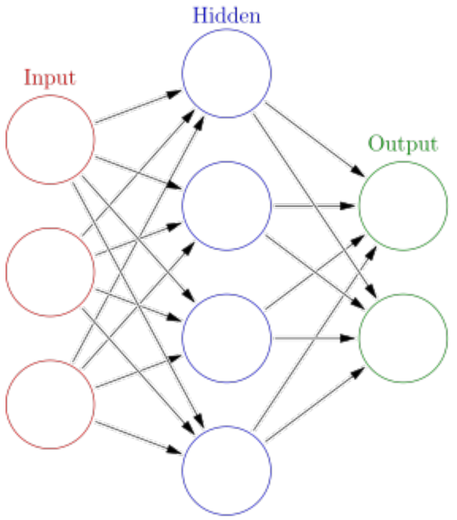
IID Setting

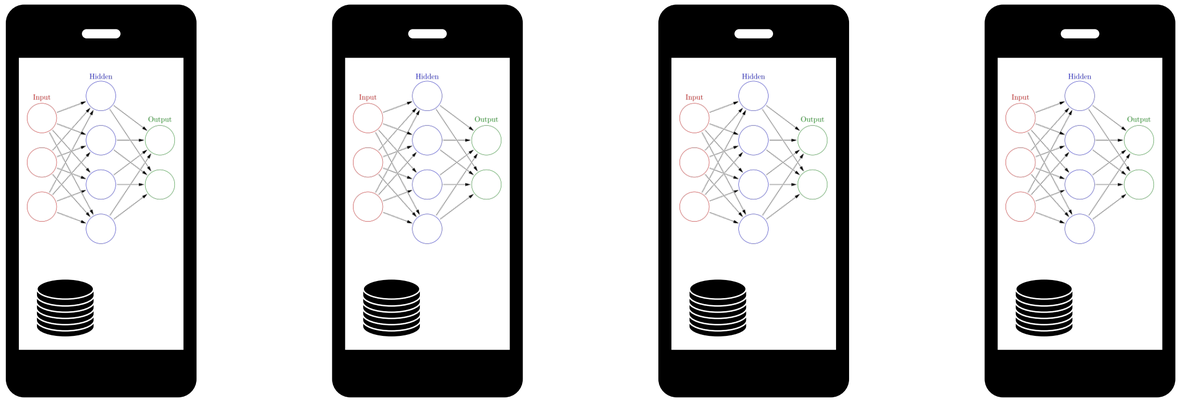


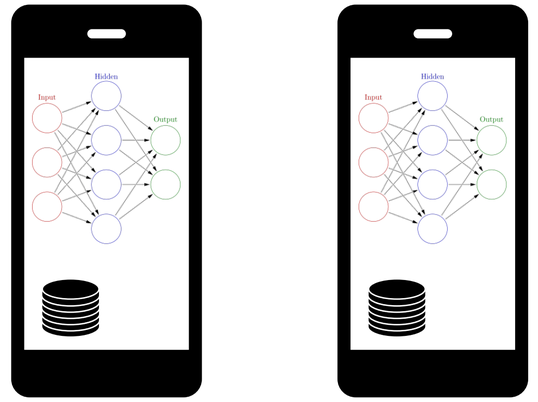


Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

IID Setting

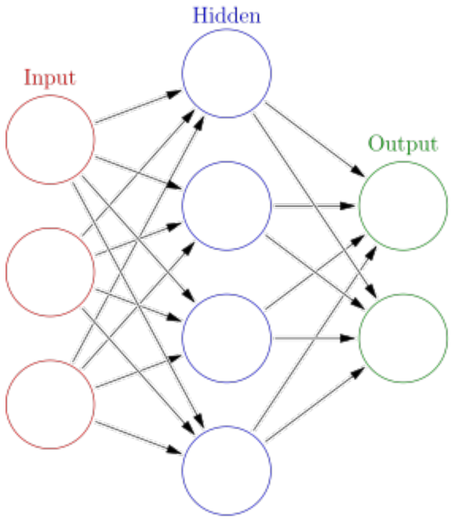


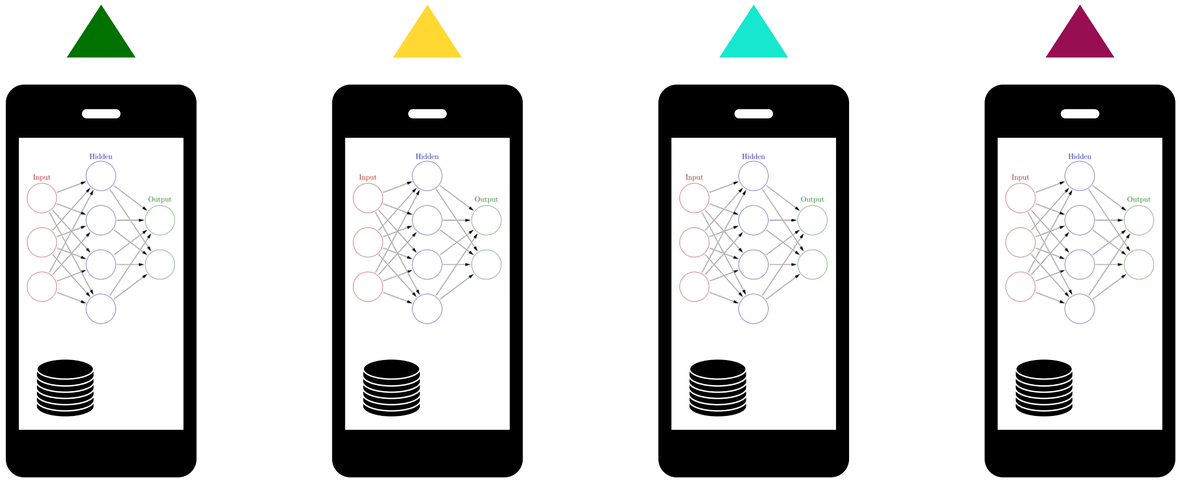


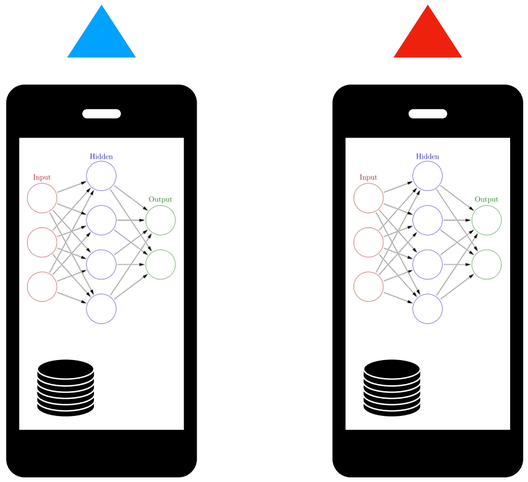


Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

IID Setting

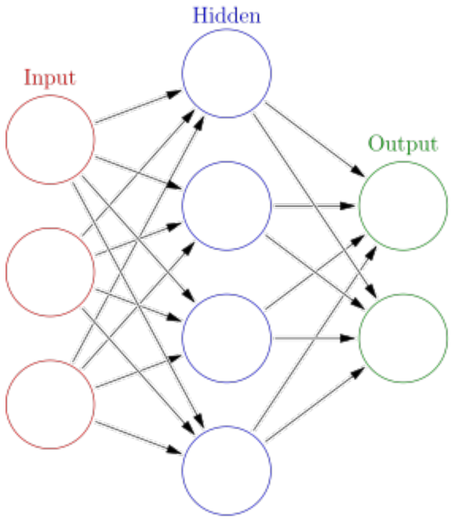


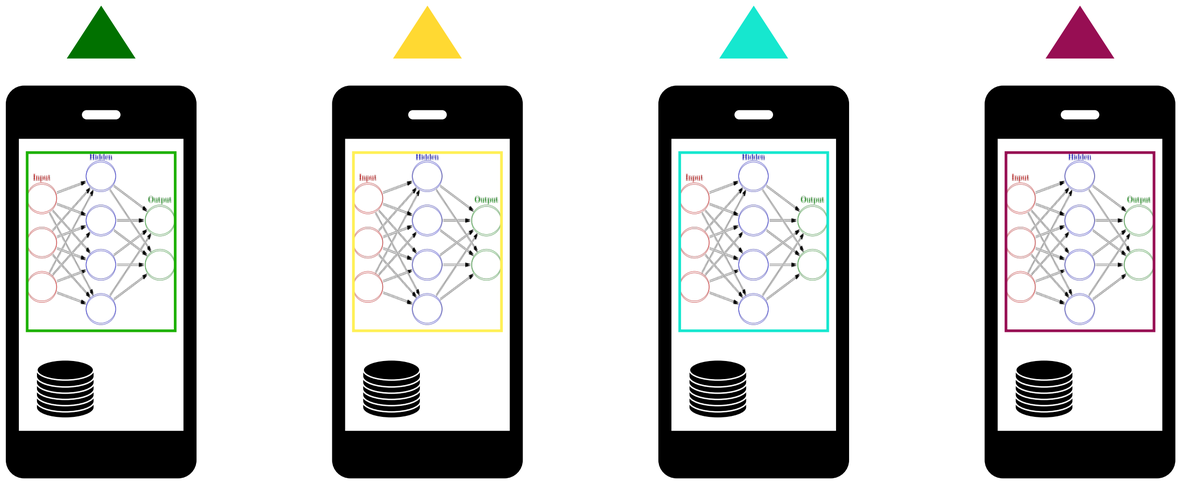


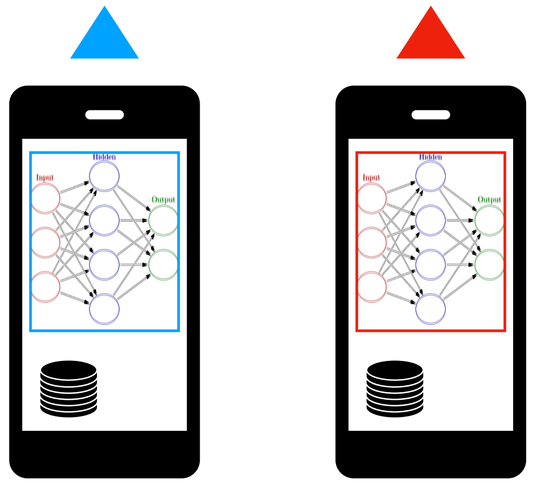


Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

IID Setting

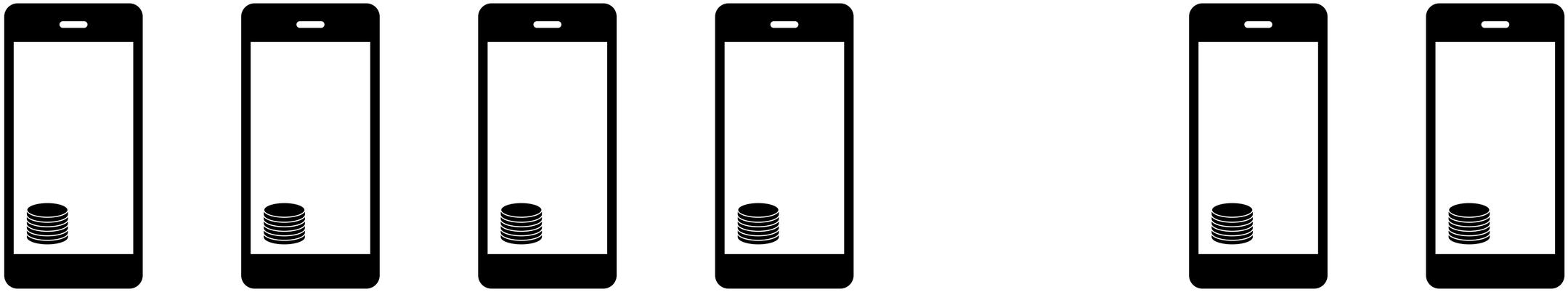






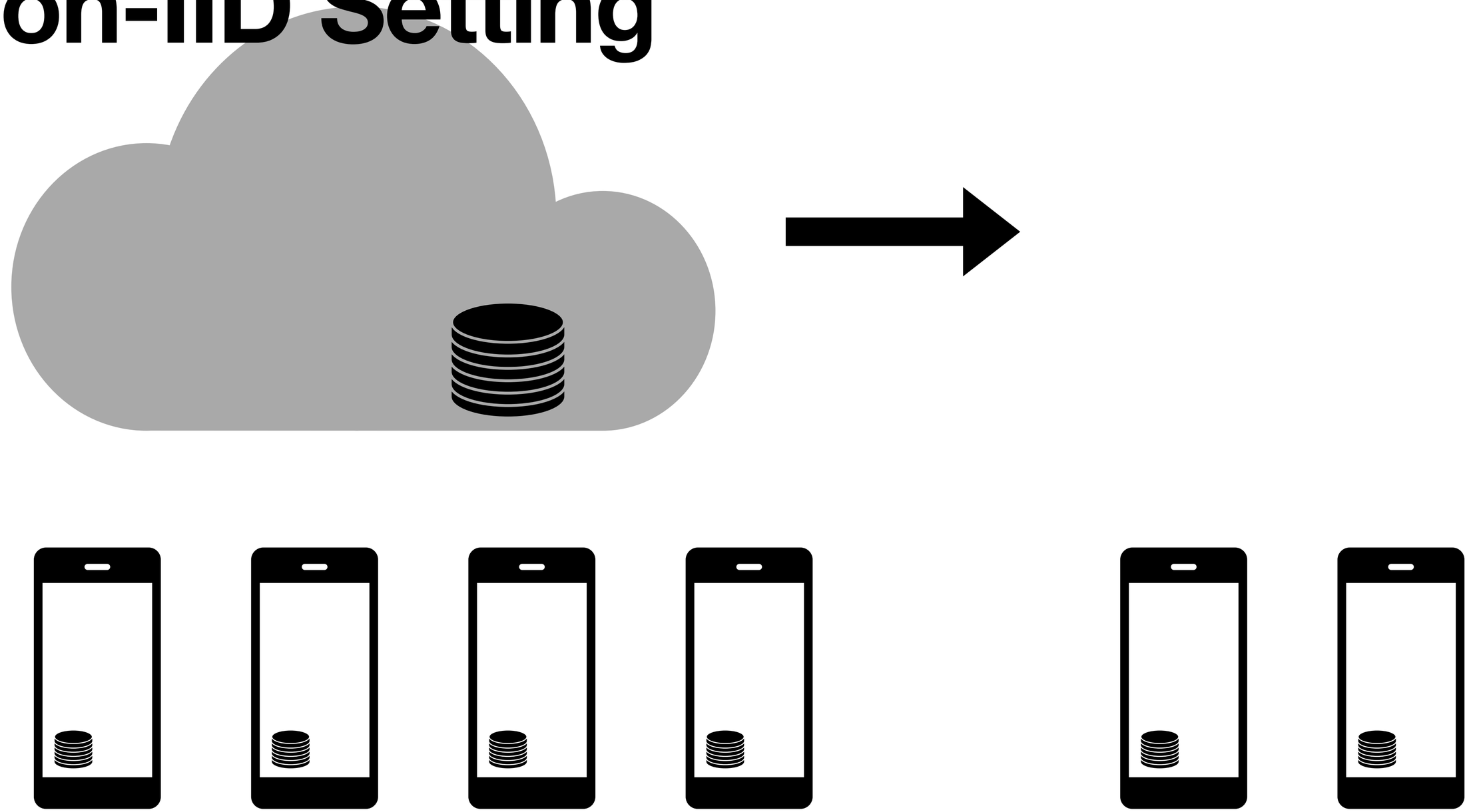
Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

Non-IID Setting



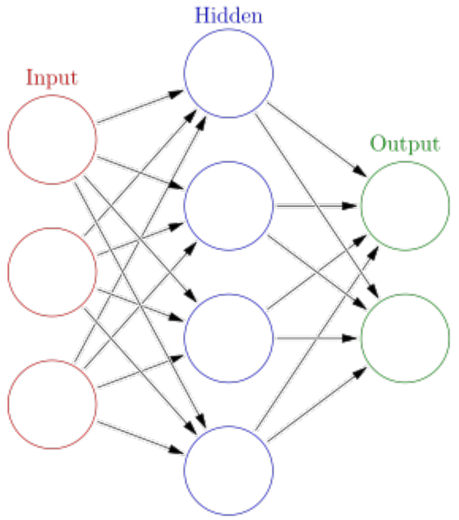
Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

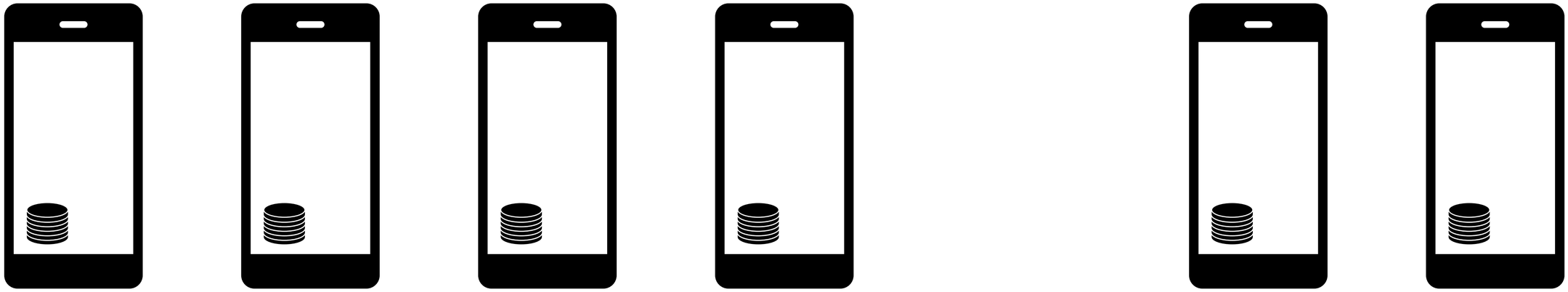
Non-IID Setting



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

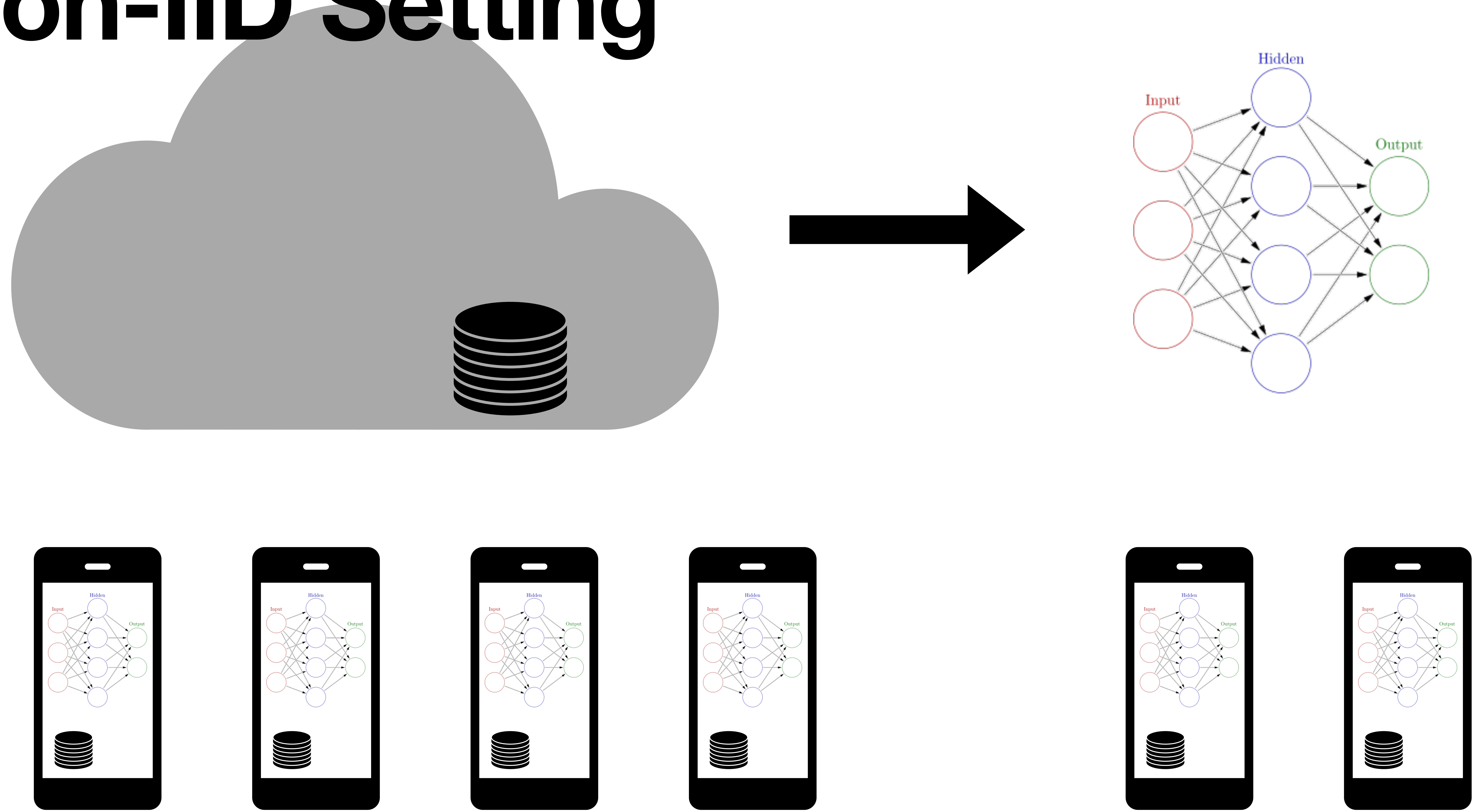
Non-IID Setting





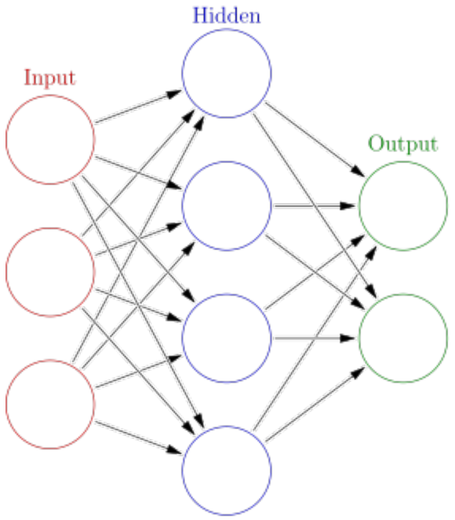
Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

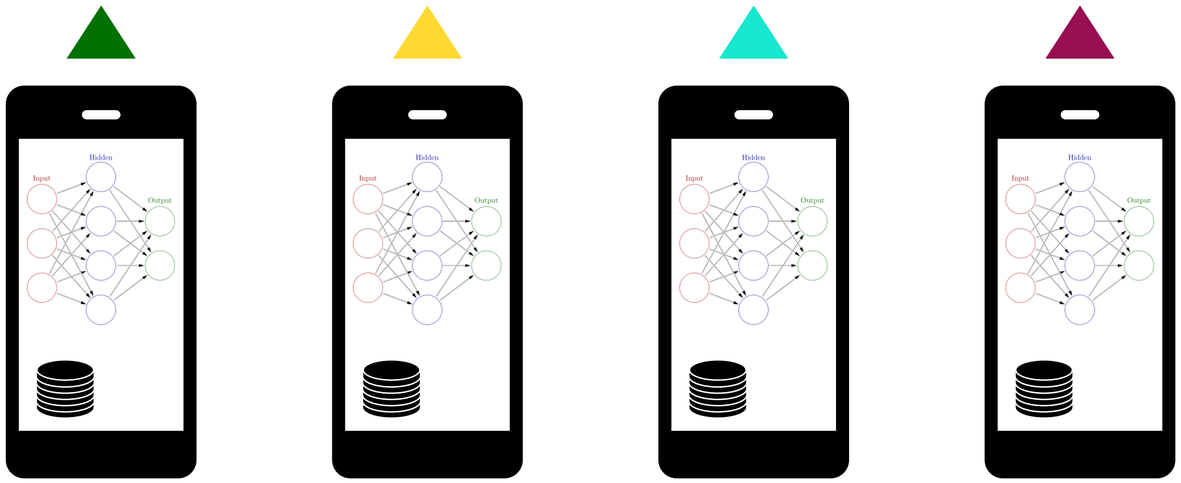
Non-IID Setting

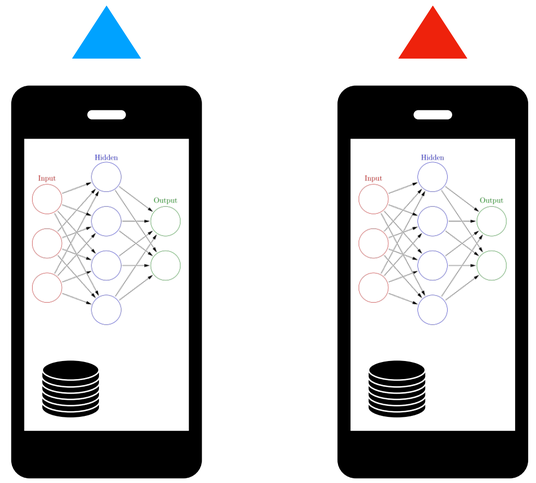


Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

Non-IID Setting

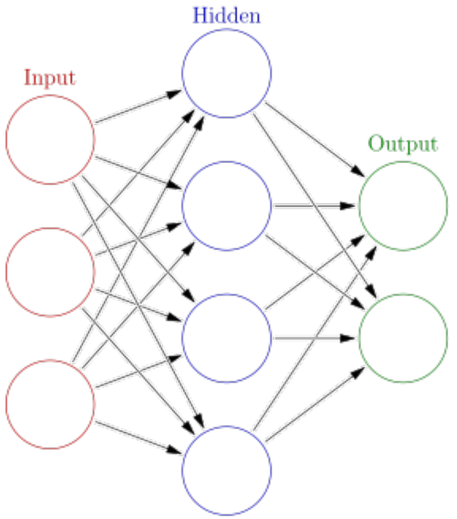


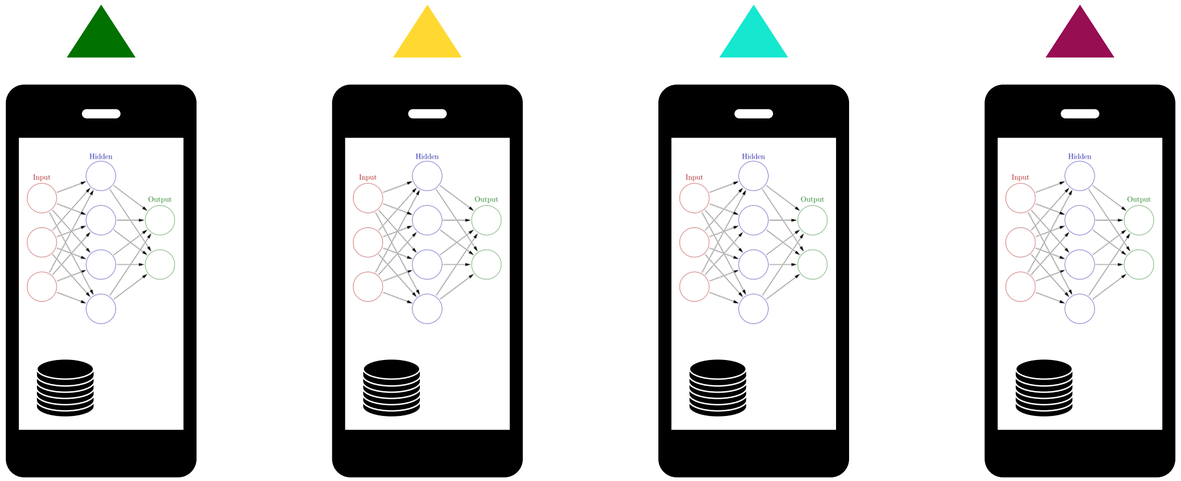


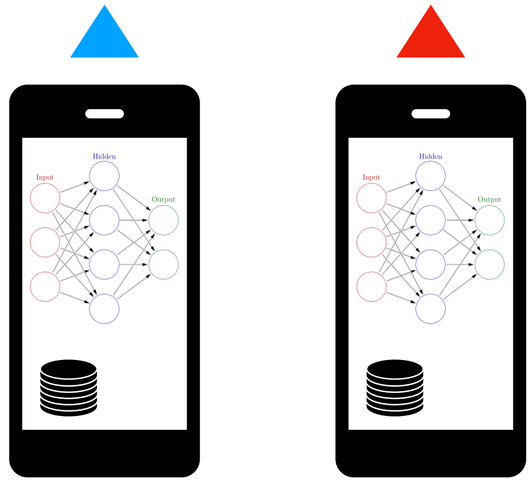


Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

Non-IID Setting

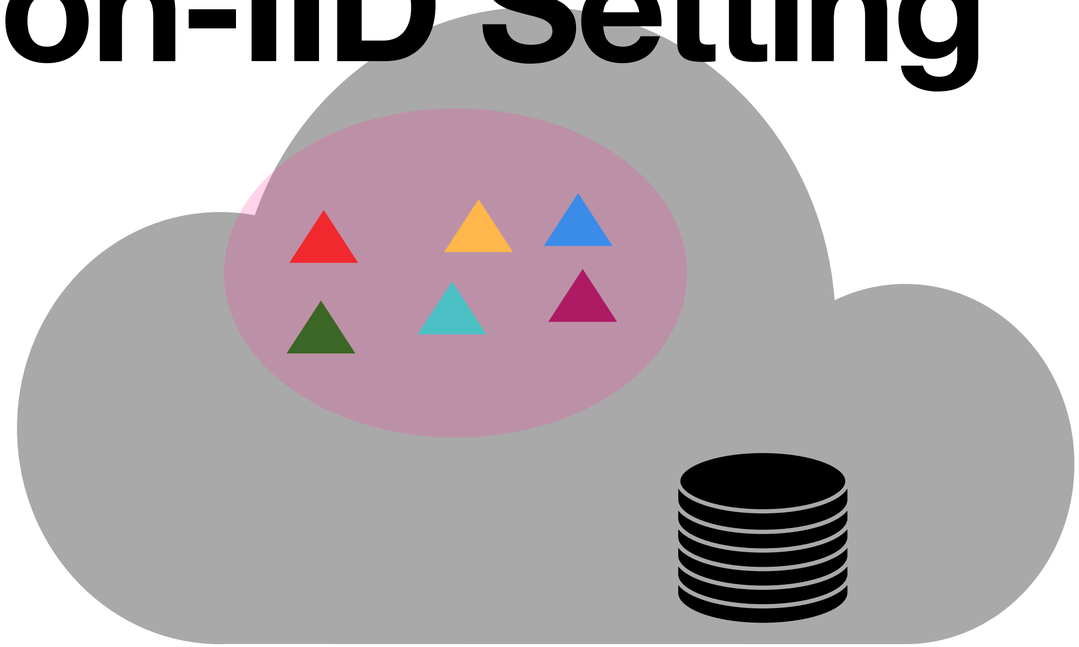


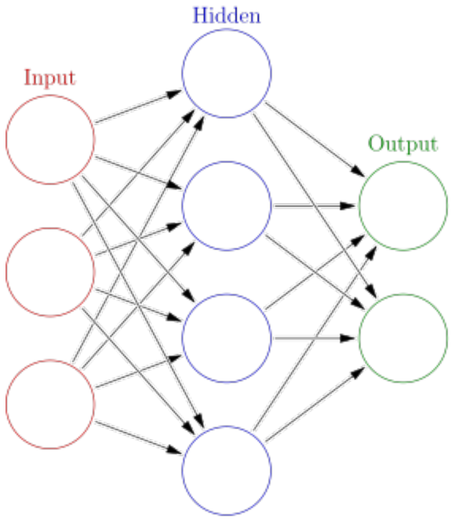


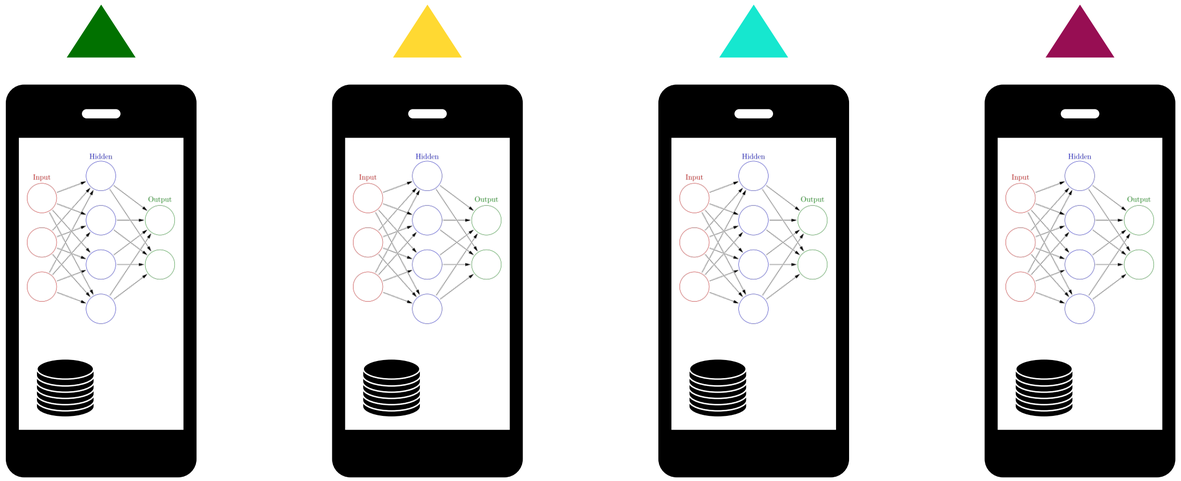


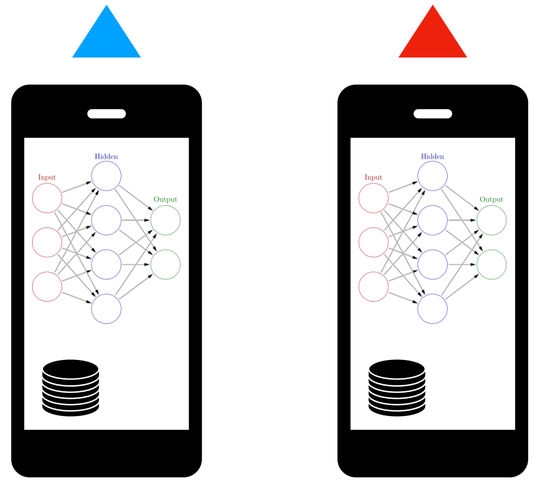
Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

Non-IID Setting



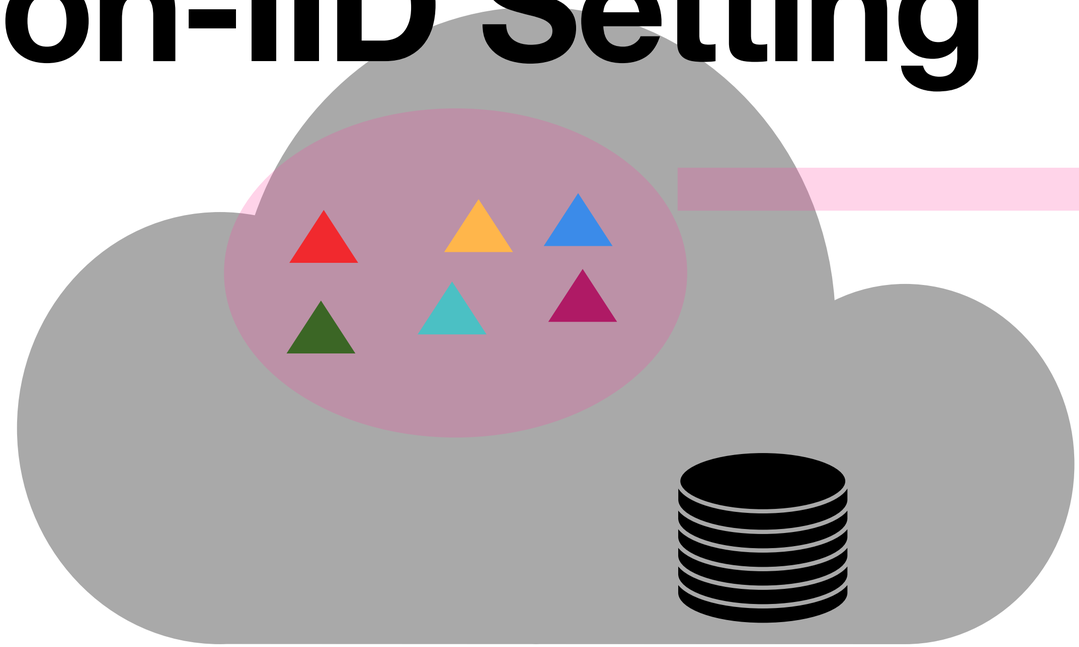


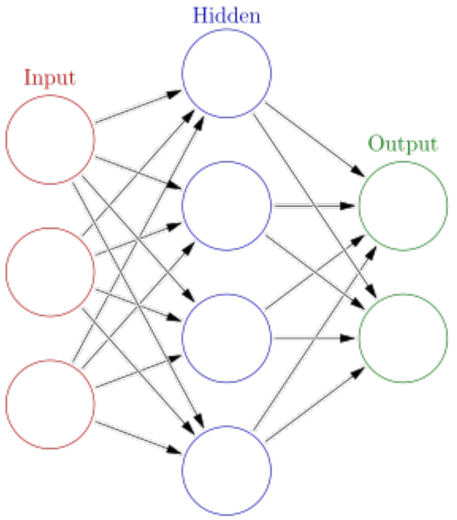


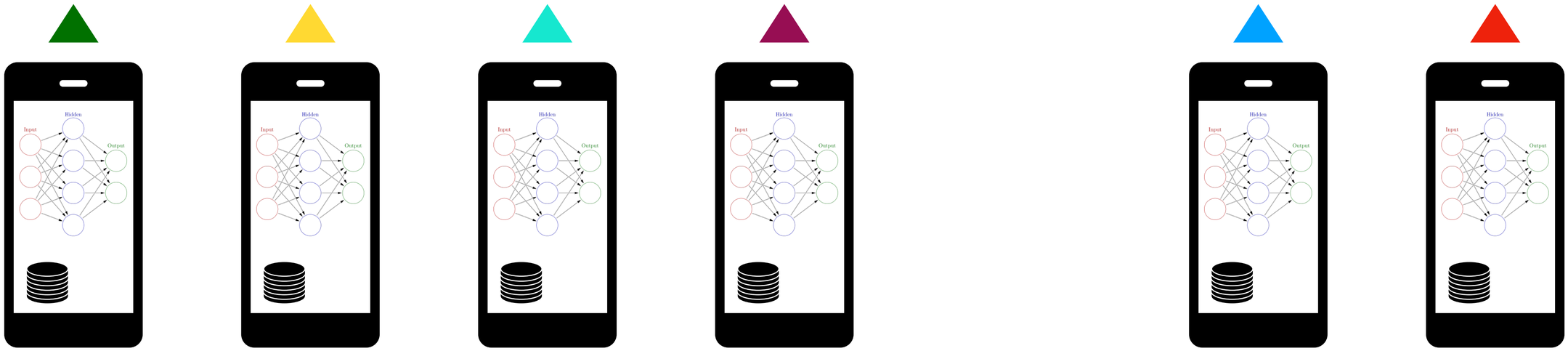


Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

Non-IID Setting

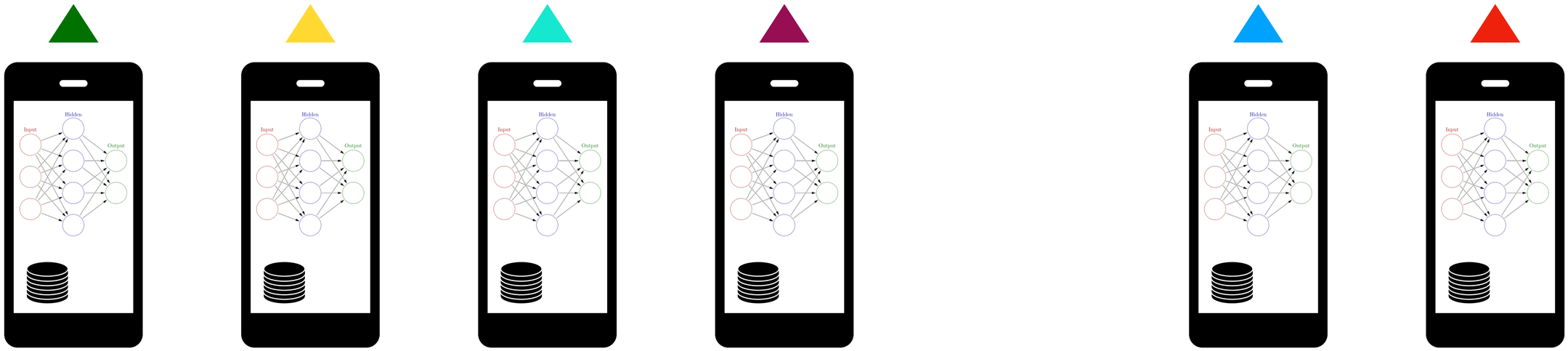






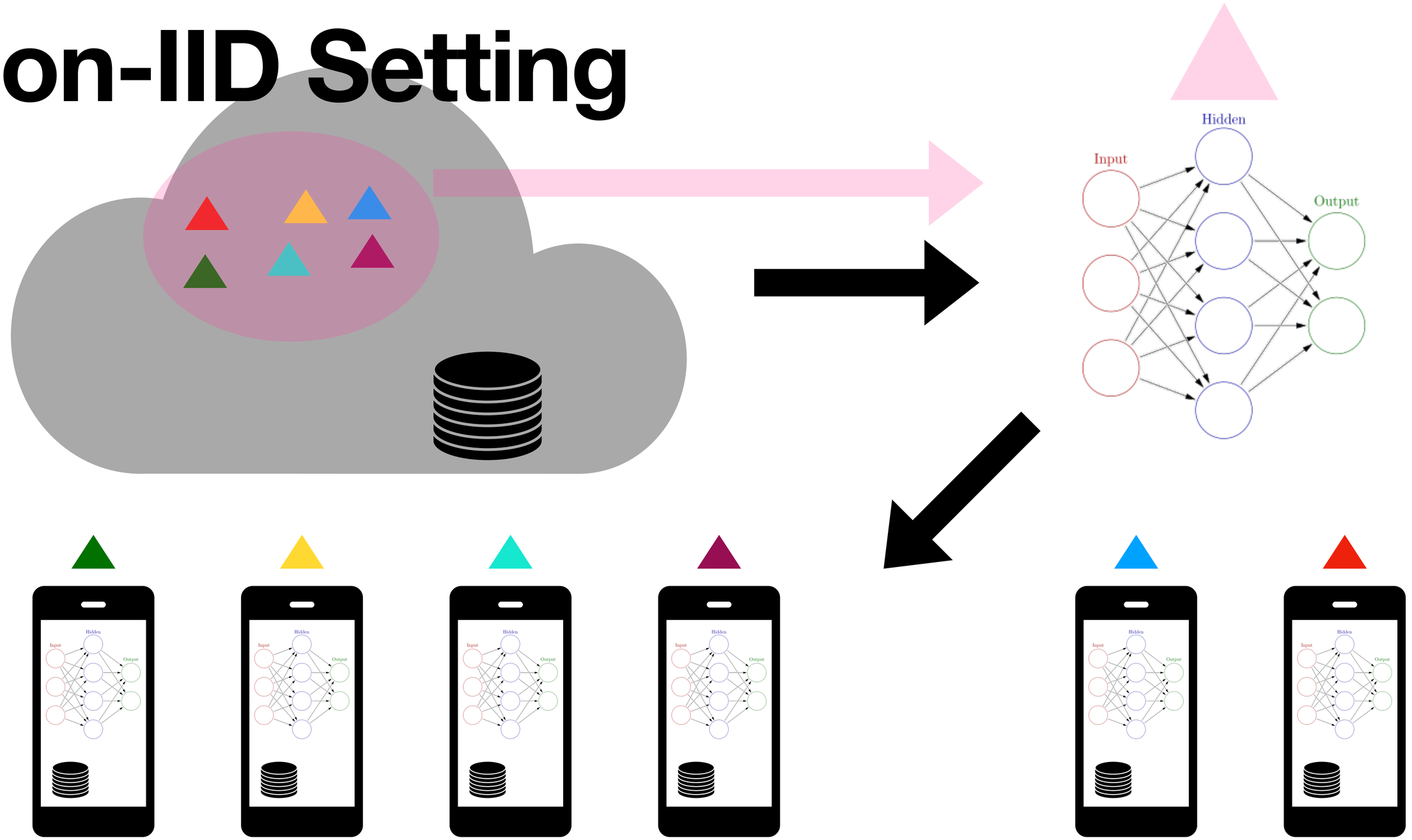
Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

Non-IID Setting



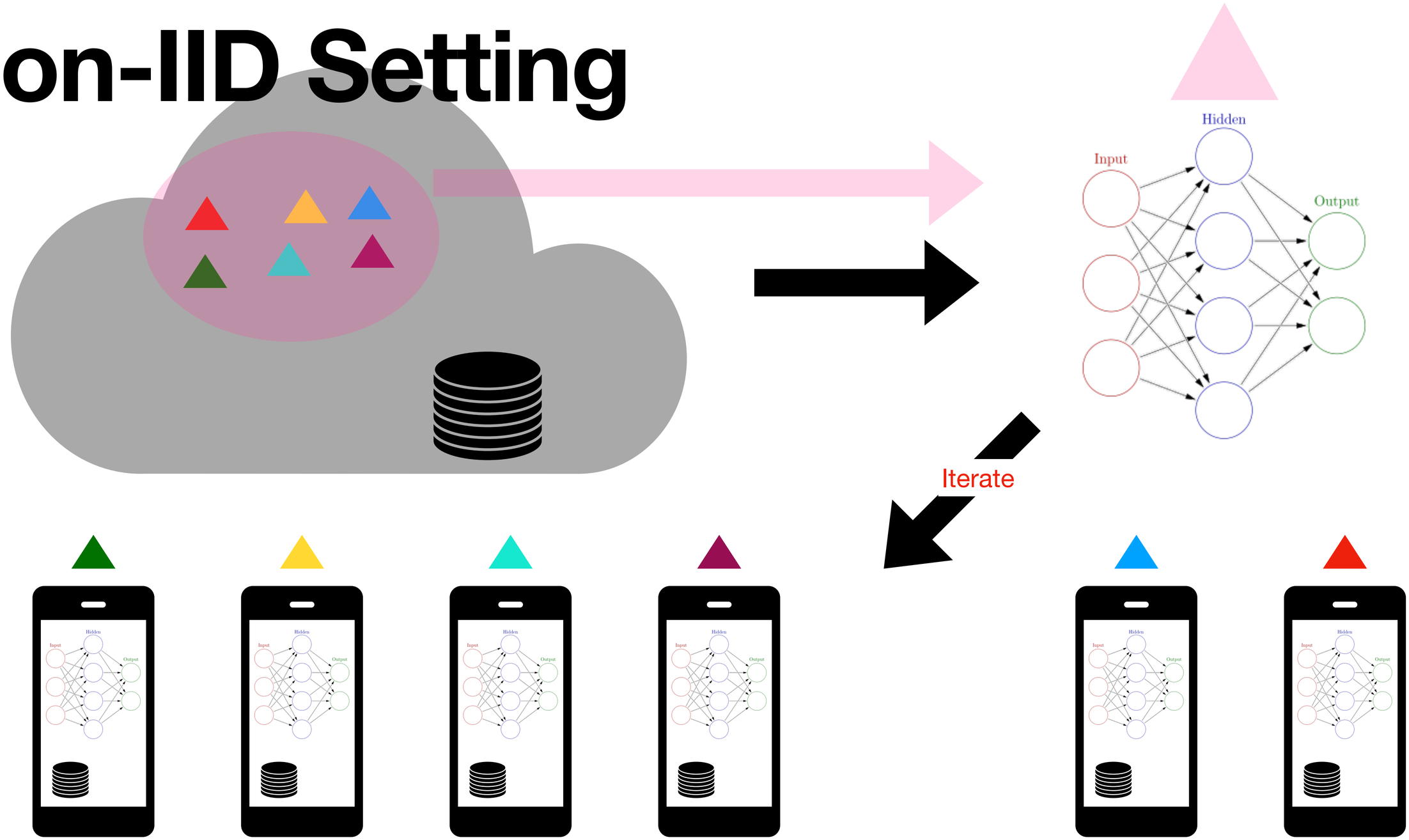
Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

Non-IID Setting



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

Non-IID Setting



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

Federated Optimization

Key properties different from a typical distributed optimization problem

- Non i.i.d
  - Local dataset will not be representative of the population distributio
- Unbalanced
  - Varying amounts of local training data
- Massively distributed
  - The number of clients is much larger than the number of examples per client
- Limited communication
  - Devices are frequently offline or on slow

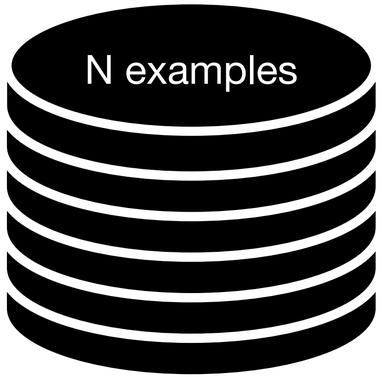
Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

Federated Optimization

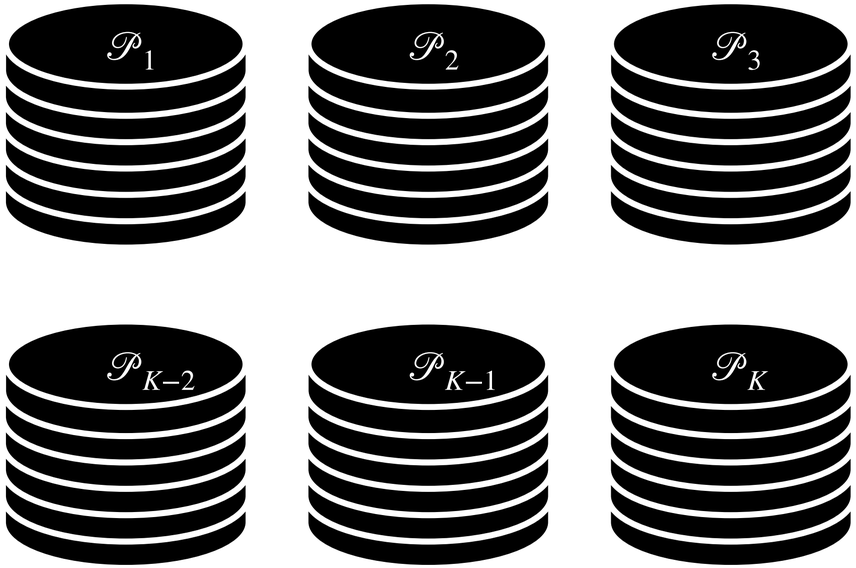
Key properties different from a typical distributed optimization problem

- Non i.i.d
  - Local dataset will not be representative of the population distributio
- Unbalanced
  - Varying amounts of local training data
- Massively distributed
  - The number of clients is much larger than the number of examples per client
- Limited communication
  - Devices are frequently offline or on slow

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



$$\min_{w \in \mathbb{R}^d} f(w) \quad \text{where} \quad f(w) = \frac{1}{N} \sum_{i=1}^{N} f_i(w), \quad f_i(w) = \ell(D_i;\, w)$$



$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w), \quad \text{where} \quad F_k(w) = \frac{1}{n_k} \sum_{i \in \mathcal{P}_k} f_i(w)$$

$$\nabla f(w) = \sum_{k=1}^{K} \frac{n_k}{n} \underbrace{\nabla F_k(w)}_{g_k}, \quad \nabla F_k(w) = \frac{1}{n_k} \sum_{i \in \mathcal{P}_k} \nabla f_i(w)$$

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

FedSGD and FedAVG

FedSGD

$$w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} g_k$$

FedAVG

$$w^k \leftarrow w^k - \eta g_k \qquad\qquad w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$$

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

FedSGD and FedAVG

FedSGD

$$w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} g_k$$

FedAVG

$$w^k \leftarrow w^k - \eta g_k \qquad w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$$

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

FedSGD and FedAVG

FedSGD

$$w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} g_k$$

FedAVG

$$w^k \leftarrow w^k - \eta g_k \qquad w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$$

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**

$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$

$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

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**

$w \leftarrow w - \eta \nabla \ell(w; b)$

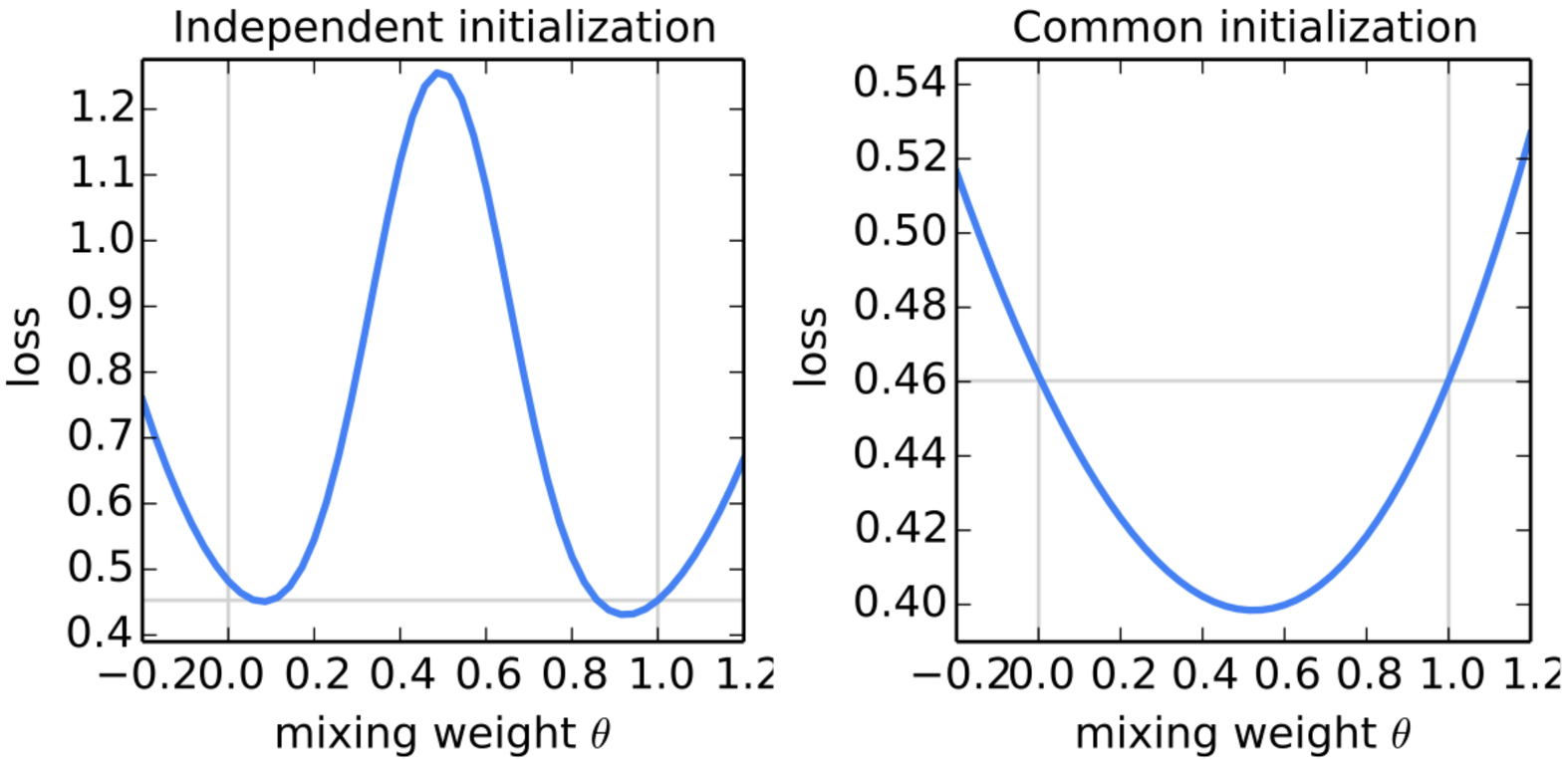
return w to server

C : the fraction of clients that perform computation on each round

B : the local mini batch size

E : the number of training passes

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



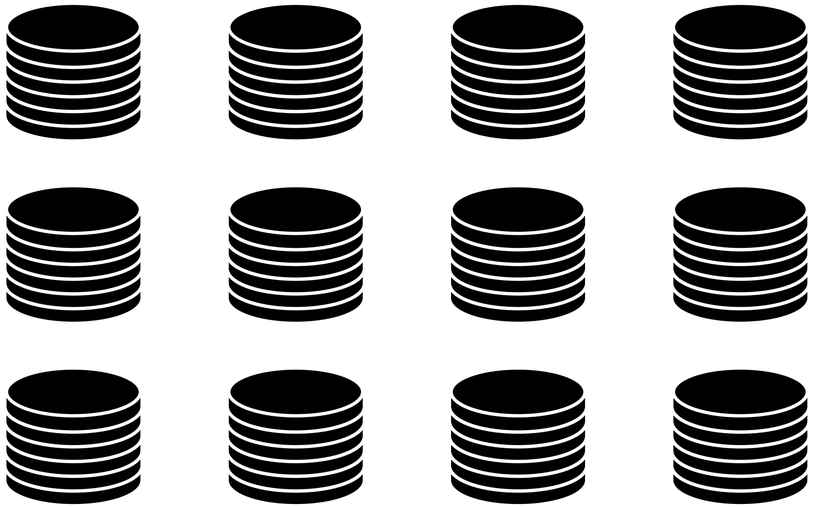
Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTATS, 2017

Experimental Results

Classification task

Image classification task

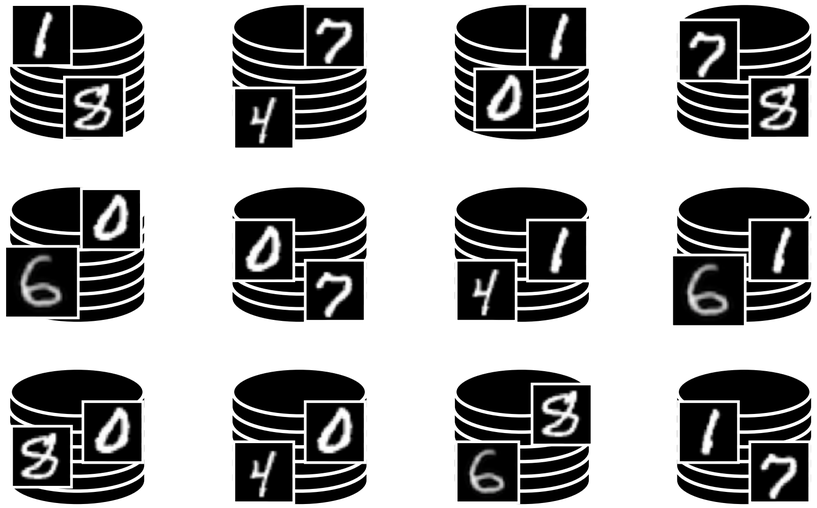
IID



Randomly shuffled

100 clients each receiving 600 examples

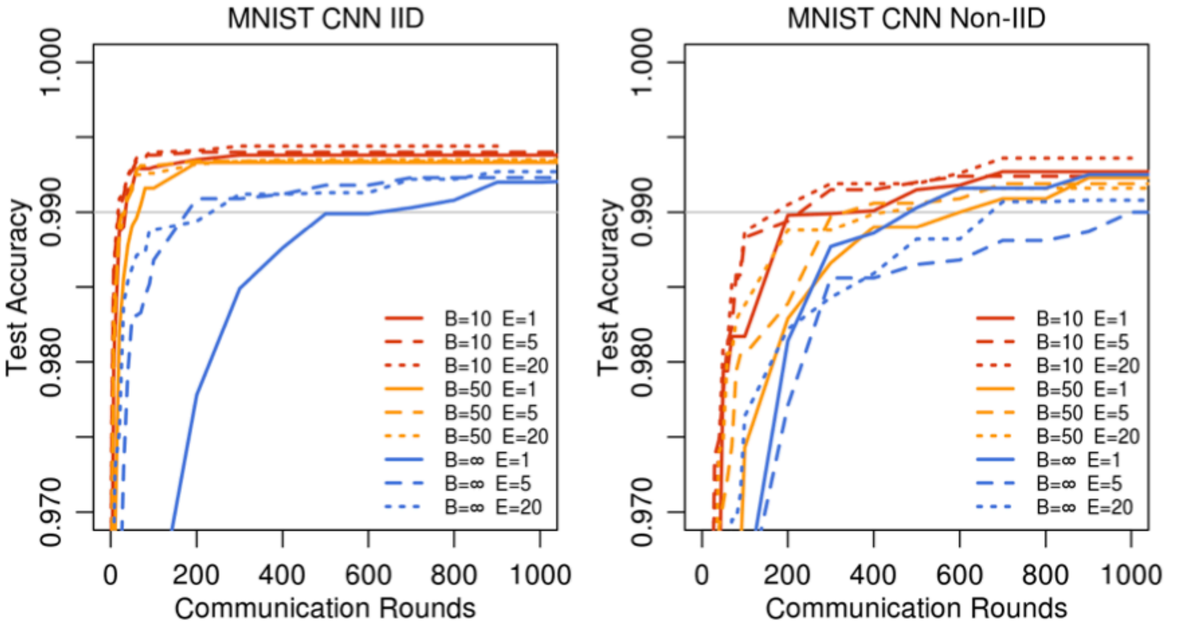
Non-IID



Non-iid partition

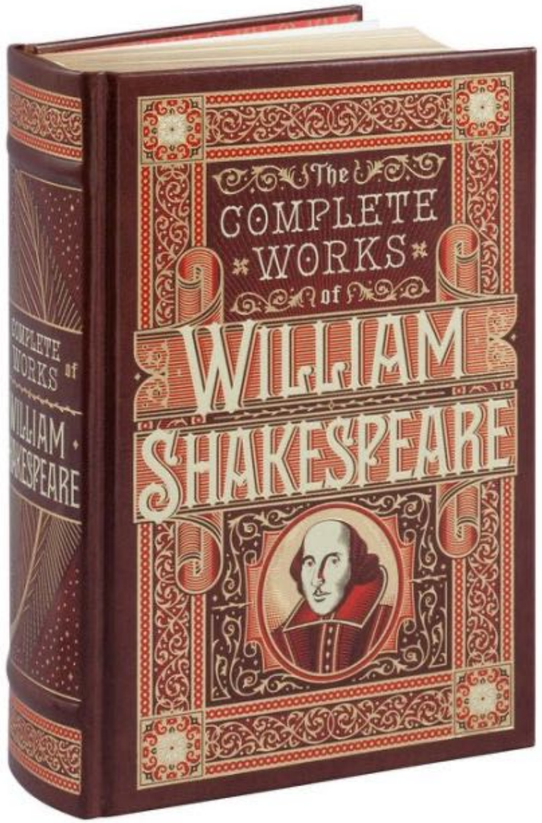
100 clients each receiving 600 examples

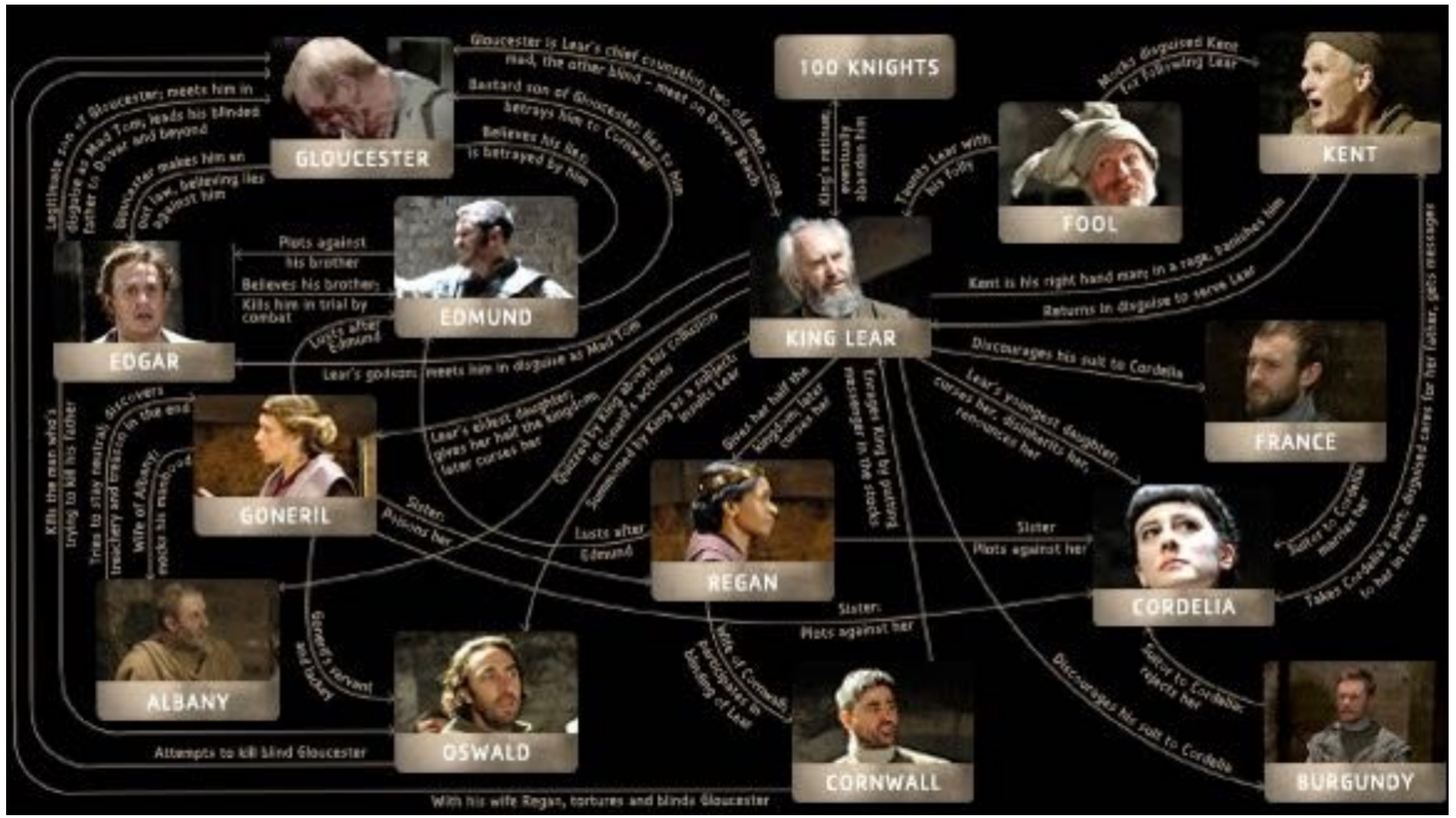
2NN	IID		Non-IID	
C	$B = \infty$	$B = 10$	$B = \infty$	$B = 10$
0.0	1455	316	4278	3275
0.1	1474 (1.0×)	87 (3.6×)	1796 (2.4×)	664 (4.9×)
0.2	1658 (0.9×)	77 (4.1×)	1528 (2.8×)	619 (5.3×)
0.5	— (—)	75 (4.2×)	— (—)	443 (7.4×)
1.0	— (—)	70 (4.5×)	— (—)	380 (8.6×)
CNN, $E = 5$				
0.0	387	50	1181	956
0.1	339 (1.1×)	18 (2.8×)	1100 (1.1×)	206 (4.6×)
0.2	337 (1.1×)	18 (2.8×)	978 (1.2×)	200 (4.8×)
0.5	164 (2.4×)	18 (2.8×)	1067 (1.1×)	261 (3.7×)
1.0	246 (1.6×)	16 (3.1×)	— (—)	97 (9.9×)



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

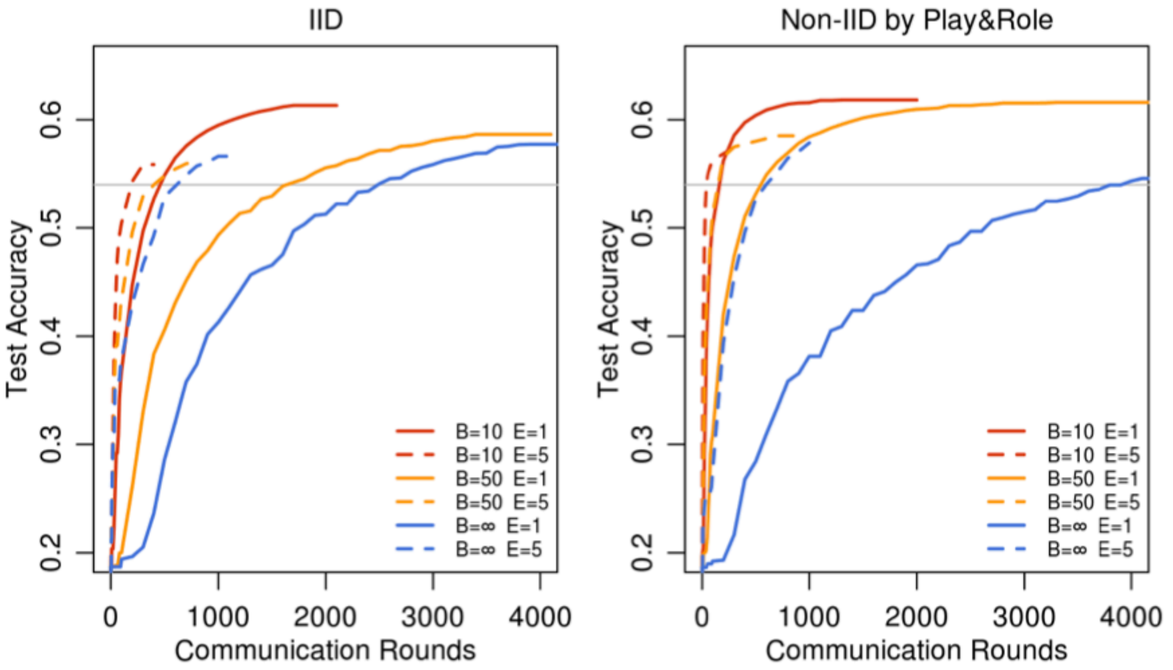
Language modeling





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

MNIST CNN, 99% ACCURACY							
CNN	E	B	u	IID		NON-IID	
FEDSGD	1	∞	1	626		483	
FEDAVG	5	∞	5	179	(3.5×)	1000	(0.5×)
FEDAVG	1	50	12	65	(9.6×)	600	(0.8×)
FEDAVG	20	∞	20	234	(2.7×)	672	(0.7×)
FEDAVG	1	10	60	34	(18.4×)	350	(1.4×)
FEDAVG	5	50	60	29	(21.6×)	334	(1.4×)
FEDAVG	20	50	240	32	(19.6×)	426	(1.1×)
FEDAVG	5	10	300	20	(31.3×)	229	(2.1×)
FEDAVG	20	10	1200	18	(34.8×)	173	(2.8×)
SHAKESPEARE LSTM, 54% ACCURACY							
LSTM	E	B	u	IID		NON-IID	
FEDSGD	1	∞	1.0	2488		3906	
FEDAVG	1	50	1.5	1635	(1.5×)	549	(7.1×)
FEDAVG	5	∞	5.0	613	(4.1×)	597	(6.5×)
FEDAVG	1	10	7.4	460	(5.4×)	164	(23.8×)
FEDAVG	5	50	7.4	401	(6.2×)	152	(25.7×)
FEDAVG	5	10	37.1	192	(13.0×)	41	(95.3×)



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

Statistical Challenges

Each node collects data in a *non-IID* manner $X_t \sim P_t$

The number of data points on each node may vary significantly

Systems Challenges

Large number of nodes causes a significant bottleneck

The storage, computational, and communication capacities of each nodes differ

Multi-Task Learning(MTL) is a natural choice to handle statistical challenges in the federated setting

Federated Multi-Task Learning, NIPS, 2017

Multi-Task Learning

- **MTL approach 1** :
  - A clustered or low-rank structure between the tasks is known *a priori*
- **MTL approach 2** :
  - The task relationships are not known beforehand
  - It can be learned from the data

MTL can address the *statistical challenges* of federated learning

Currently proposed methods for distributed **MTL** don't adequately address the *systems challenges*

Federated Multi-Task Learning, NIPS, 2017

Federated Multi-Task Learning

General Multi-Task Learning Setup

- General formulation :

$$\min_{W,\Omega} \quad \left[\sum_{k=1}^{K} \sum_{i \in \mathcal{P}_k} f_k(w_k^\top x_k^i, y_k^i) + \mathcal{R}(W, \Omega) \right]$$

$\Omega \in \mathbb{R}^{K \times K}$ models *relationships among tasks*. f_k is a convex loss function.

- MTL problems differ based on their assumptions on $\mathcal{R}$

$$\mathcal{R}(W, \Omega) = \lambda_1 \, tr\left(W \Omega W^\top\right) + \lambda_2 \, | \, W |_F^2$$

Federated Multi-Task Learning

MOCHA : A Framework for Federated Multi-Task Learning

$$\min_{W,\Omega} \quad \left[\sum_{k=1}^{K} \sum_{i \in \mathcal{P}_k} f_k(w_k^\top x_k^i, y_k^i) + \mathcal{R}(W, \Omega) \right]$$

$$\mathcal{R}(W, \Omega) = \lambda_1 \, tr\left(W \Omega W^\top\right) + \lambda_2 \, |W|_F^2$$

- ✓ Not jointly convex in W and Ω
- ✓ When fixing Ω, updating W depends on the data X.
- ✓ When fixing W, optimizing for Ω only depends on W, not on X.

Federated Multi-Task Learning, NIPS, 2017

Federated Multi-Task Learning

MOCHA : A Framework for Federated Multi-Task Learning

$$\min_{W,\Omega} \quad \left[\sum_{k=1}^{K} \sum_{i \in \mathcal{P}_k} f_k(w_k^\top x_k^i, y_k^i) + \mathcal{R}(W, \Omega) \right]$$

$$\mathcal{R}(W, \Omega) = \lambda_1 \, tr\left(W \Omega W^\top\right) + \lambda_2 \, |W|_F^2$$

- ✓ Not jointly convex in W and Ω
- ✓ When fixing Ω, updating W depends on the data X.
- ✓ When fixing W, optimizing for Ω only depends on W, not on X.

Federated Multi-Task Learning, NIPS, 2017

Federated Multi-Task Learning

Federated Update of W

Dual Problem

$$\min_{\alpha \in \mathbb{R}^N} \left[D(\alpha) := \sum_{k=1}^{K} \sum_{i \in \mathcal{P}_k} f_k^*(-\alpha_k^i) + \mathcal{R}^*(X\alpha) \right] \qquad X = diag(X_1, \ldots, X_K)$$

Data-local subproblem

$$\min_{\Delta\alpha_k \in \mathbb{R}^N} G_k^{\sigma'}(\Delta\alpha_k;\ v_k, \alpha_k) := \sum_{i \in \mathcal{P}_k} f_k^*(-\alpha_k^i - \Delta\alpha_k^i) + \langle w_k(\alpha), X_k \Delta\alpha_k \rangle + \frac{\alpha}{2} |\, X_k \Delta\alpha_k \,|_{M_k}^2 + c(\alpha)$$

Federated Multi-Task Learning, NIPS, 2017

Federated Multi-Task Learning

Federated Update of W

Algorithm 1 MOCHA: Federated Multi-Task Learning Framework

Input: Data $\mathbf{X}_t$ from $t = 1, \ldots, m$ tasks, stored on one of m nodes, and initial matrix $\boldsymbol{\Omega}_0$
Starting point $\boldsymbol{\alpha}^{(0)} := \mathbf{0} \in \mathbb{R}^n$, $\mathbf{v}^{(0)} := \mathbf{0} \in \mathbb{R}^b$
for iterations $i = 0, 1, \ldots$ **do**
    Set subproblem parameter σ' and number of federated iterations, H_i
    for iterations $h = 0, 1, \cdots, H_i$ **do**
        for tasks $t \in \{1, 2, \ldots, m\}$ **in parallel over** m **nodes do**
            call local solver, returning θ_t^h-approximate solution $\Delta\boldsymbol{\alpha}_t$ of the local subproblem (4)
            update local variables $\boldsymbol{\alpha}_t \leftarrow \boldsymbol{\alpha}_t + \Delta\boldsymbol{\alpha}_t$
            return updates $\Delta\mathbf{v}_t := \mathbf{X}_t\Delta\boldsymbol{\alpha}_t$
        reduce: $\mathbf{v}_t \leftarrow \mathbf{v}_t + \Delta\mathbf{v}_t$
    Update $\boldsymbol{\Omega}$ centrally based on $\mathbf{w}(\boldsymbol{\alpha})$ for latest $\boldsymbol{\alpha}$
Central node computes $\mathbf{w} = \mathbf{w}(\boldsymbol{\alpha})$ based on the lastest $\boldsymbol{\alpha}$
return: $\mathbf{W} := [\mathbf{w}_1, \ldots, \mathbf{w}_m]$

Convergence Analysis

Definition (Per-Node-Per-Iteration-Approximation Parameter)

At each iteration h, we define the accuracy level of the solution calculated by node k to its subproblem as

$$\theta_k^h := \frac{G_k^{\sigma'}(\Delta\alpha_k^{(h)}; v^{(h)}, \alpha_k^{(h)}) - G_k^{\sigma'}(\Delta\alpha_k^{\star}; v^{(h)}, \alpha_k^{(h)})}{G_k^{\sigma'}(\mathbf{0}; v^{(h)}, \alpha_k^{(h)}) - G_k^{\sigma'}(\Delta\alpha_k^{\star}; v^{(h)}, \alpha_k^{(h)})}$$

$\theta_k^h \in [0, 1]$, $\theta_k^h = 1$ means that no updates to the subproblem are made at iteration h

Assumption

Let $\mathscr{H}_h := (\alpha^{(h)}, \ldots, \alpha^{(1)})$ be the *dual vector history* until the beginning of iteration h, and define

$\Theta_k^h := \mathbb{E}[\theta_k^h \,|\, \mathscr{H}_h]$. For all tasks k and all iterations h, we assume $p_k^h := \mathbb{P}(\theta_k^h = 1) \leq p_{max} < 1$ and

$\hat{\Theta}_k^h = \mathbb{E}[\theta_k^h \,|\, \mathscr{H}_h, \theta_k^h < 1] \leq \Theta_{max} < 1.$

Convergence Analysis

Theorem 1

Assume that the losses f_k are $(1/\mu)$-smooth. Then, under Assumptions 1 and 2, there exists a constant $s \in [0, 1]$ such that for any given convergence target ε_D, choosing H such that

$$H \geq \frac{1}{(1 - \bar{\Theta})s} \log \frac{n}{\varepsilon_D}$$

will satisfy $\mathbb{E}[D(\alpha^{(H)}) - D(\alpha^\star)] \leq \varepsilon_D$. Here, $\bar{\Theta} := p_{max} + (1 - p_{max})\Theta_{max} < 1$.

Experimental Results

Google Glass

38 participants wearing Google Glass

Features

- ✓ Accelerometer
- ✓ Magnetometer
- ✓ Gyroscope

Predict between eating and other activities

Federated Multi-Task Learning, NIPS, 2017

Human Activity Recognition

Mobile phone from 30 individuals.

Features

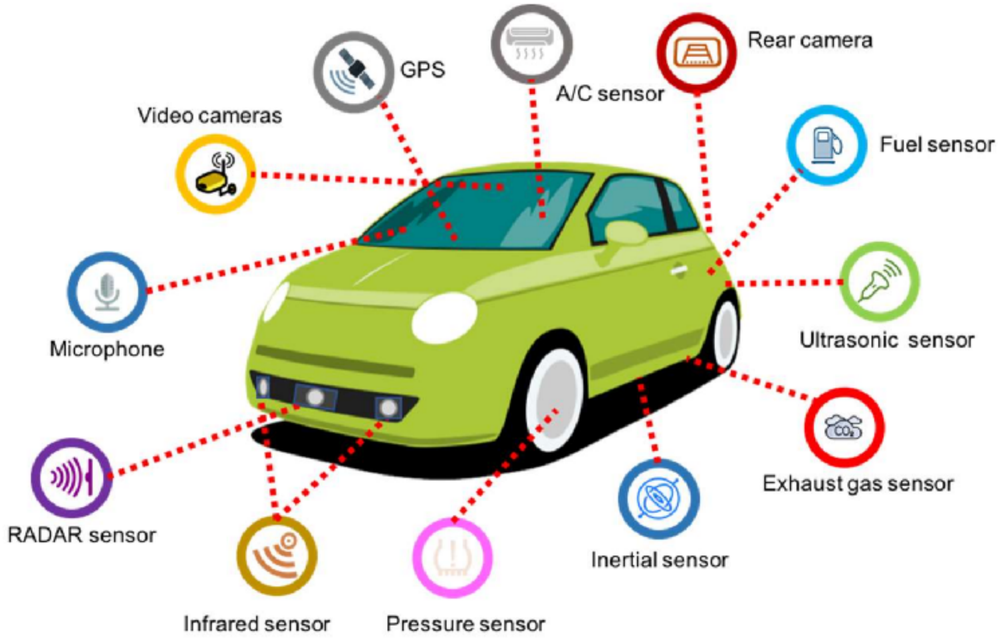
✓ Accelerometer

✓ Gyroscope

Predict between sitting and other activities

Federated Multi-Task Learning, NIPS, 2017

Vehicle Sensor



23 sensors for the aim of segmentation.

Features

- ✔ Acoustic feature $(\times 50)$
- ✔ Seismic feature $(\times 50)$

Predict between AAV-type and DW-type vehicles

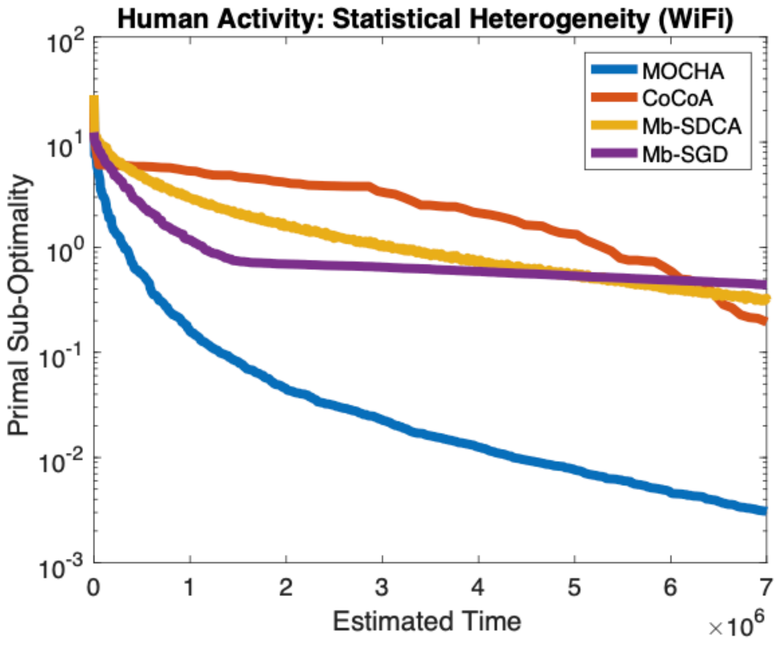
Federated Multi-Task Learning, NIPS, 2017

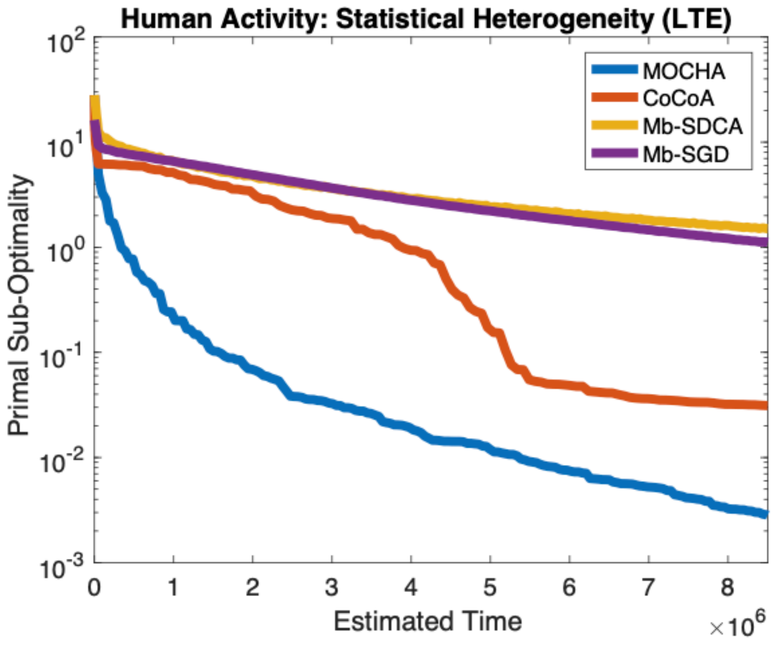
Simulations

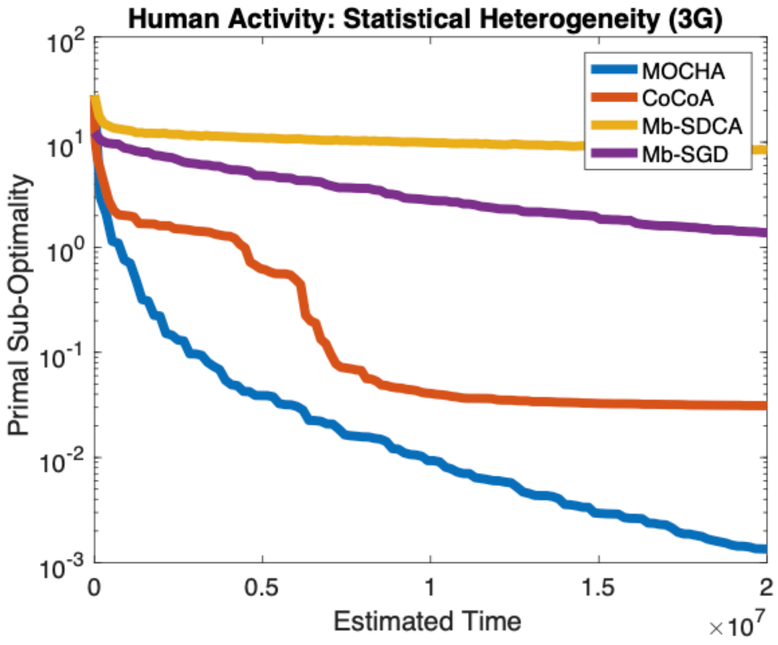
Multi-Task Learning for the Federated Setting

Table 1: Average prediction error: Means and standard errors over 10 random shuffles.

Model	Human Activity	Google Glass	Vehicle Sensor
Global	2.23 (0.30)	5.34 (0.26)	13.4 (0.26)
Local	1.34 (0.21)	4.92 (0.26)	7.81 (0.13)
MTL	**0.46 (0.11)**	**2.02 (0.15)**	**6.59 (0.21)**


Human Activity: Statistical Heterogeneity (WiFi)
MOCHA
CoCoA
Mb-SDCA
Mb-SGD
Primal Sub-Optimality
Estimated Time
×10^6

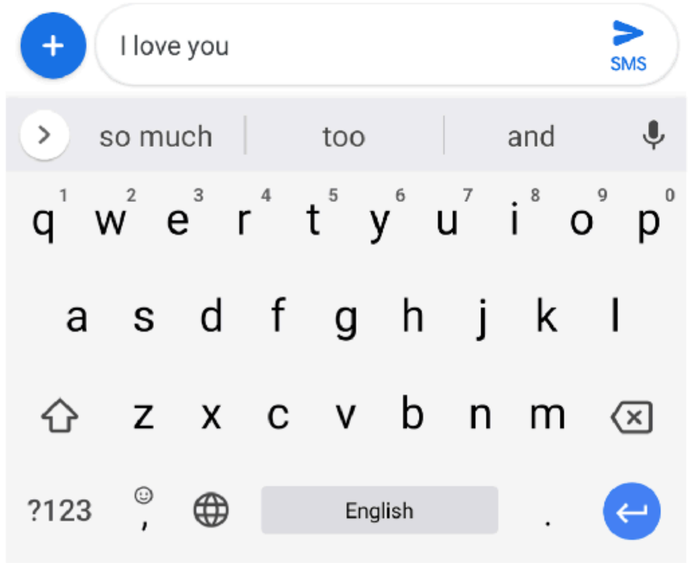

Human Activity: Statistical Heterogeneity (LTE)
MOCHA
CoCoA
Mb-SDCA
Mb-SGD
Primal Sub-Optimality
Estimated Time
×10^6

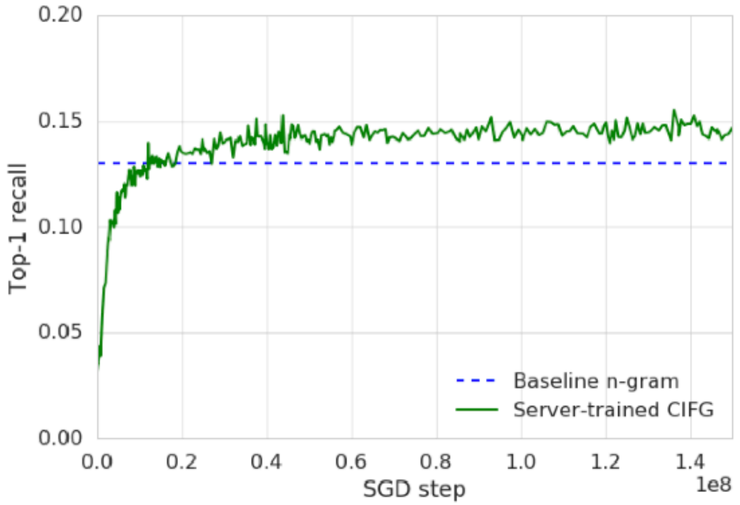

Human Activity: Statistical Heterogeneity (3G)
MOCHA
CoCoA
Mb-SDCA
Mb-SGD
Primal Sub-Optimality
Estimated Time
×10^7

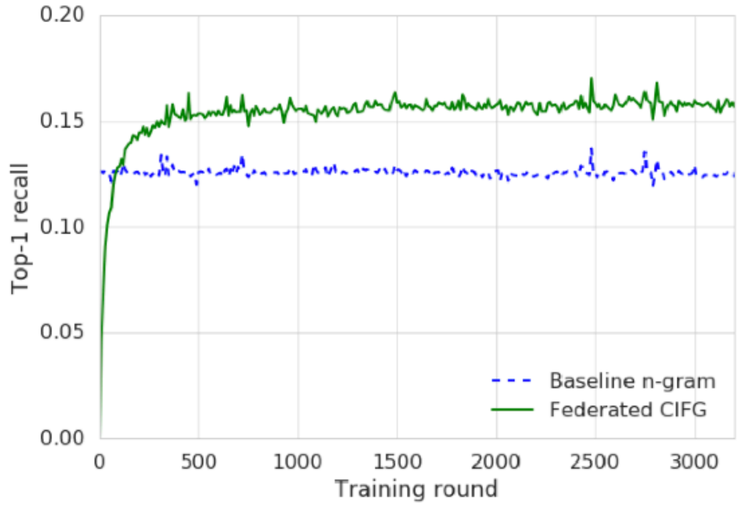
Application & Deployment

TensorFlow Federated

Gboard : language modeling



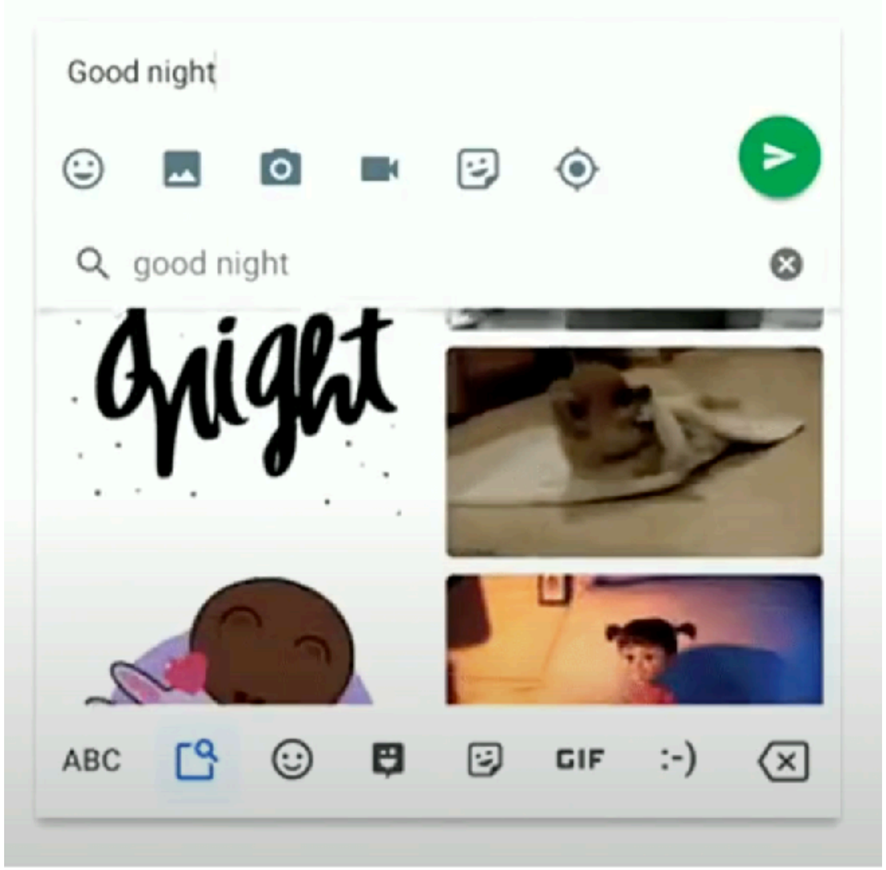




, Google, 2018

- ✓ Better next-word prediction accuracy : +24%
- ✓ More useful prediction strip : +10% more clicks

Gboard : emoji prediction



- ✓ 7% more accurate emoji prediction
- ✓ More useful prediction strip : +4% more clicks
- ✓ 11% more users share emojis!

TensorFlow Federated

https://www.tensorflow.org/federated

PySyft

https://www.openmined.org