Learning at the Edge

Vince Poor

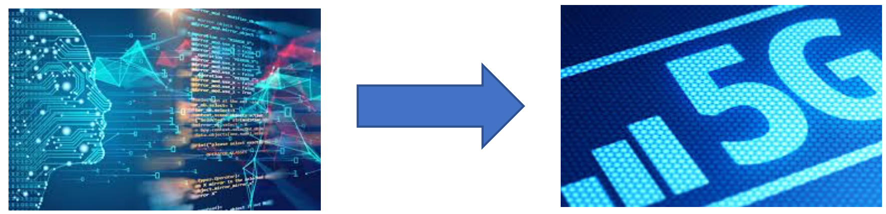
Princeton University

Machine Learning (ML) and Mobile Communications

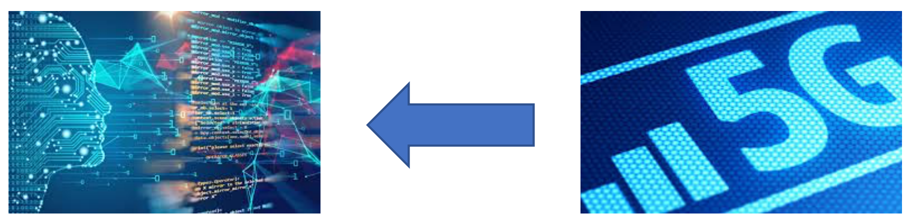
- Two Aspects:
  - Using machine learning to optimize communication networks

Machine Learning (ML) and Mobile Communications

- Two Aspects:



  - Using machine learning to optimize communication networks
  - Learning on mobile devices (the focus of today's talk)

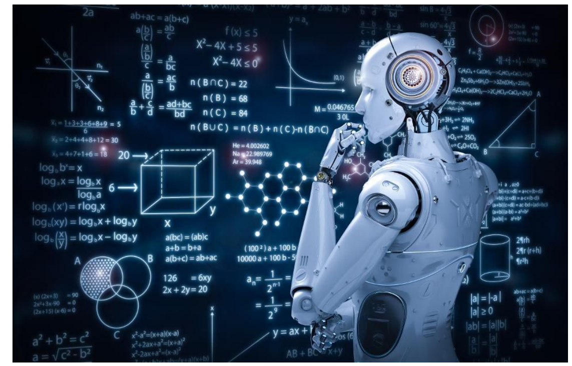


Outline

- Overview and Motivation
- Federated Learning
- Decentralized Learning (Briefly)
- Conclusions

Overview
and
Motivation

Machine Learning (ML): State-of-the-Art



- Tremendous progress in recent years
  - More and more data is available
  - Significant increase in computational power
- "Standard" ML
  - Implemented in a centralized manner (e.g., in a data center/cloud)
  - Full access to the data

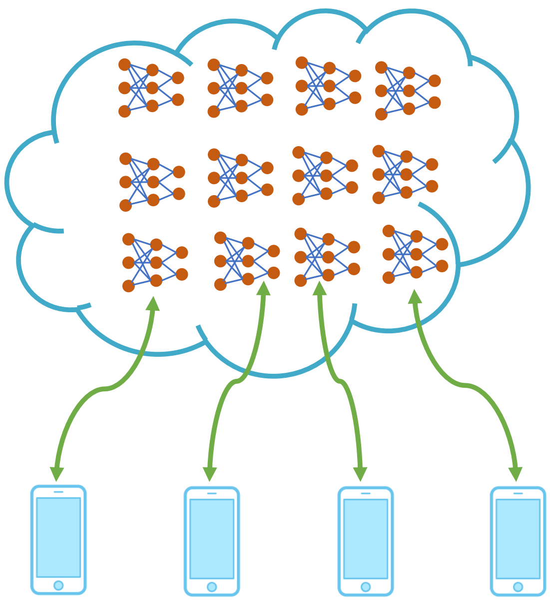
- State-of-the art models (e.g., Deep Neural Networks) run in the cloud
  - Managed and operated by standard software tools (e.g., TensorFlow, etc.)
  - Accelerated by specialized hardware (e.g., Nvidia's GPUs, Google's TPUs)

Machine Learning at the Wireless Edge

- Centralized ML may not be suitable for many emerging applications, e.g.,
  - Tactical networks
  - First responder network
  - Self-driving cars
- What makes these applications/situations different?
  - Data is born at the edge (phones and IoT devices)
  - Limited capacity uplinks
  - Low latency & high reliability
  - Data privacy / security
  - Scalability & locality
- Motivates moving learning closer to the network edge
  - Jointly optimize learning and communication

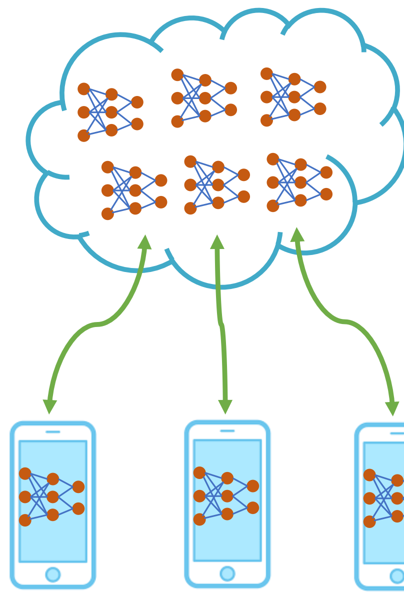
Distributed ML Models

“Standard” ML



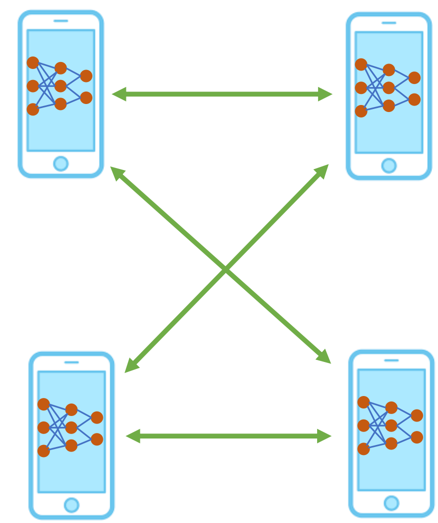
- ML in the cloud with dumb end-user devices
- All data is in the cloud
- Inference and decision making in the cloud
- No data privacy

Federated ML



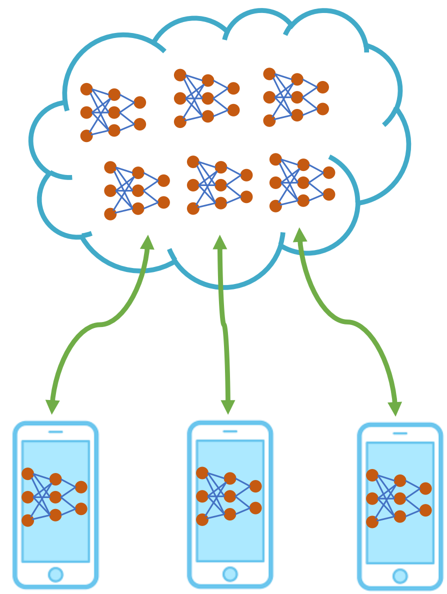
- ML in the cloud + on-user-device ML
- Only part of the data is in the cloud
- Use the cloud but smartly
- Privacy-preserving

Decentralized ML

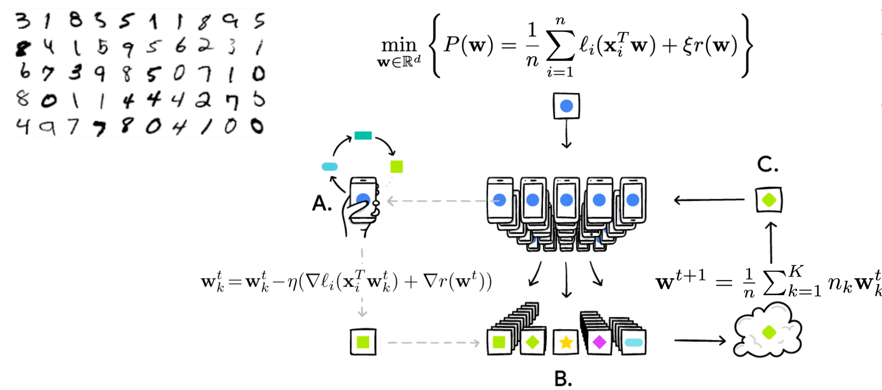


- No infrastructure (e.g., cloud) needed
- Data is fully distributed
- Collaborative intelligence
- Privacy-preserving (sharing models instead of data)

Federated
Learning



Federated Learning: Basic Architecture



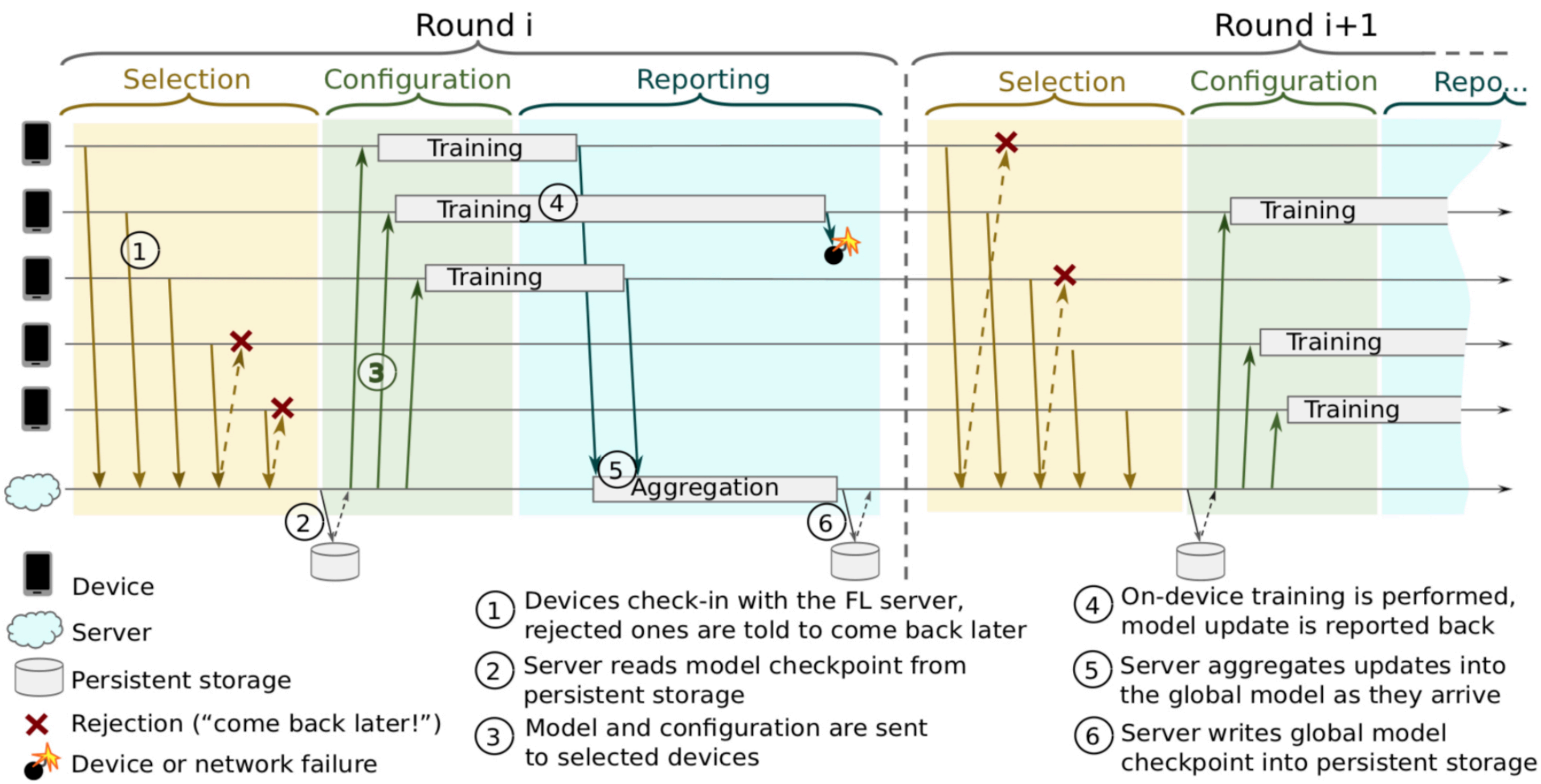
Model	Loss function $\ell_i(\mathbf{x}_i^T\mathbf{w})$
Smooth SVM	$\frac{\xi}{2}\lVert\mathbf{w}\rVert^2 + \frac{1}{2}\max\{0, 1 - y_i\mathbf{w}^T\mathbf{x}_i\}$
Linear regression	$\frac{1}{2}\lVert y_i - \mathbf{w}^T\mathbf{x}_i\rVert^2$
K-means	$\frac{1}{2}\min_{j\in\{1,2,\dots,d\}} \lVert\mathbf{x}_i - (\mathbf{w}^T)_j\rVert^2$

- Key features
  - On-device datasets: end users keep raw data locally
  - On-device training: end-user devices perform training on a shared model
  - Federated computation: an edge node (AP or BS) collects trained weights from end users and updates the shared model (iterated till convergence)

Federated Learning: Issues to Address

- Learning at the edge
  - Communication to the edge node needs to go through wireless channels
  - The communication medium is shared and resource-constrained
    - Only a limited number of end-user devices can be selected in each update round
    - Transmissions are not reliable due to interference
- Questions
  - How should the edge device schedule end-user devices to update trained weights?
  - How does the interference affect the training?

Federated Learning: Evolution in Time



Federated Learning: System Model

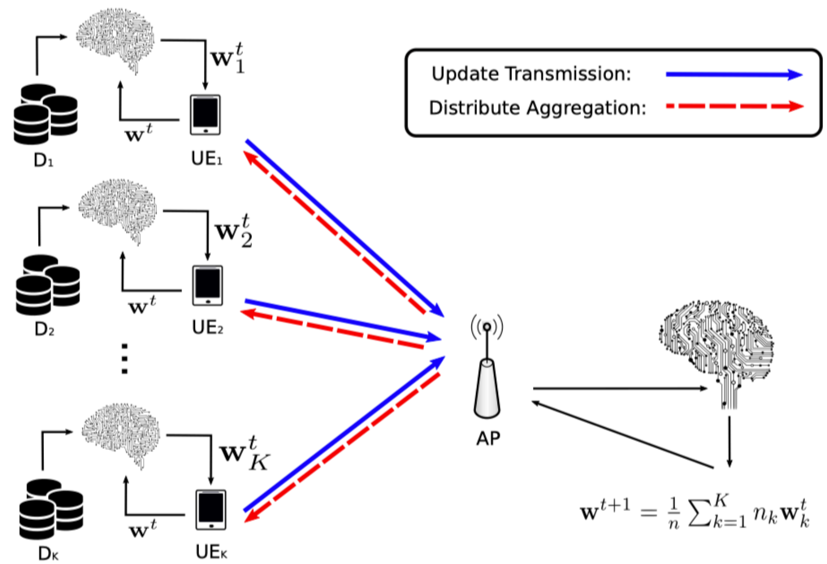
- <u>Mobile edge network</u>
  - APs and UEs capable of computing
  - Each AP has K associated UEs

- <u>Spectrum configuration</u>
  - Spectrum is divided into N subchannels, where $N < K$, and globally reused

Scheduling Mechanisms*

- <u>Scheduling mechanisms</u>
  - <u>Random Scheduling</u>: AP uniformly selects N out of K UEs at random
  - <u>Round Robin</u>: AP groups UEs into $G=K/N$ groups, sequentially selecting each group
  - <u>Proportional Fair</u>: AP selects N out of K UEs with the strongest SNRs:

$$\mathbf{m}^* = \arg\max_{\mathbf{m} \subset \{1,2,\ldots,K\}} \left\{ \frac{\tilde{R}_{m_1}}{\bar{R}_{m_1}}, \ldots, \frac{\tilde{R}_{m_N}}{\bar{R}_{m_N}} \right\}$$

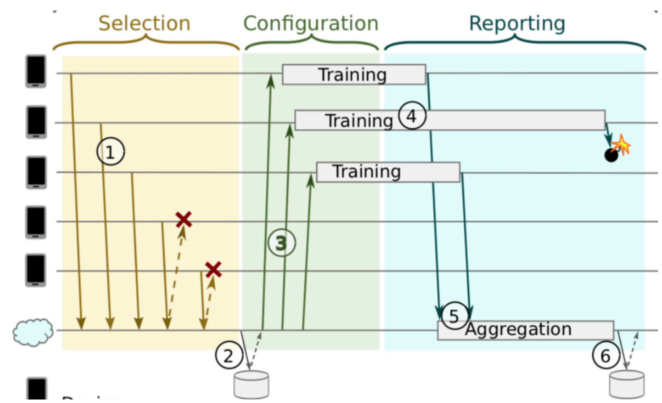


* H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling Policies for Federated Learning in Wireless Networks", *IEEE Trans. Commun.*, to appear.

Performance Metric

- Federated Learning in a mobile edge network
  - The trained update can be successfully received by AP if and only if
    - The UE is selected by the AP, and
    - The received SINR exceeds a decoding threshold:

$$\gamma_{k,t} = \frac{P_{\mathrm{ut}} h_k \|z_k\|^{-\alpha}}{\sum_{z \in \tilde{\Phi}_{\mathrm{u}}^k} P_{\mathrm{ut}} h_z \|z\|^{-\alpha} + \sigma^2} > \theta.$$



- Metric to quantify the effectiveness of training
  - The number of communication rounds required to reach an ε-accurate solution

Convergence Rates of Federated Learning

Theorem 1: *Under RS policy, for any given convergence target ε, choosing the T_{RS} such that*

$$T_{\mathrm{RS}} \geq \frac{\log(\varepsilon/n)}{\log\left(1 - \frac{(1-\beta)/G}{1+\mathcal{V}(\theta,\alpha)}\right)}, \tag{28}$$

we have the expected duality gap satisfies $\mathbb{E}[P(\mathbf{w}(\mathbf{a}^{T_{\mathrm{RS}}})) - D(\mathbf{a}^{T_{\mathrm{RS}}})] < \varepsilon$.

Theorem 2: *Under RR policy, for any given convergence target ε, choosing the T_{RR} such that*

$$T_{\mathrm{RR}} \geq \frac{G\log(\varepsilon/n)}{\log\left(1 - \frac{1-\beta}{1+\mathcal{V}(\theta,\alpha)}\right)}, \tag{31}$$

we have the expected duality gap satisfies $\mathbb{E}[P(\mathbf{w}(\mathbf{a}^{T_{\mathrm{RR}}})) - D(\mathbf{a}^{T_{\mathrm{RR}}})] < \varepsilon$.

Theorem 3: *Under PF policy, for any given convergence target ε, choosing the T_{PF} such that*

$$T_{\mathrm{PF}} \geq \frac{\log(\varepsilon/n)}{\log\left(1-(1-\beta)\sum_{i=1}^{K-N+1}\binom{K-N+1}{i}\frac{(-1)^{i+1}/G}{1+\mathcal{V}(i\theta,\alpha)}\right)}, \tag{33}$$

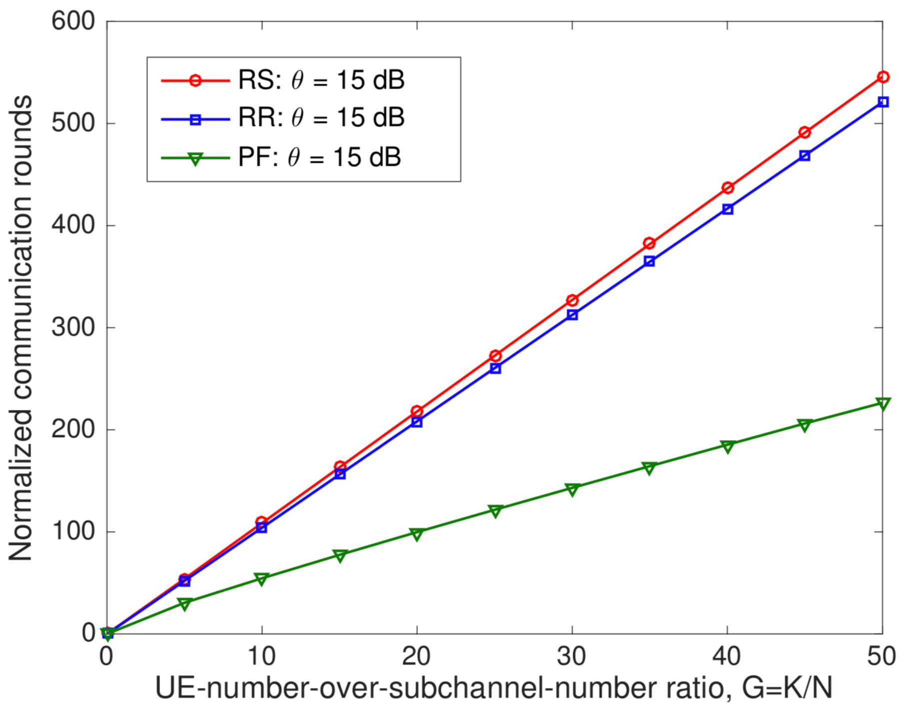
we have the expected duality gap satisfies $\mathbb{E}[P(\mathbf{w}(\mathbf{a}^{T_{\mathrm{PF}}})) - D(\mathbf{a}^{T_{\mathrm{PF}}})] < \varepsilon$.

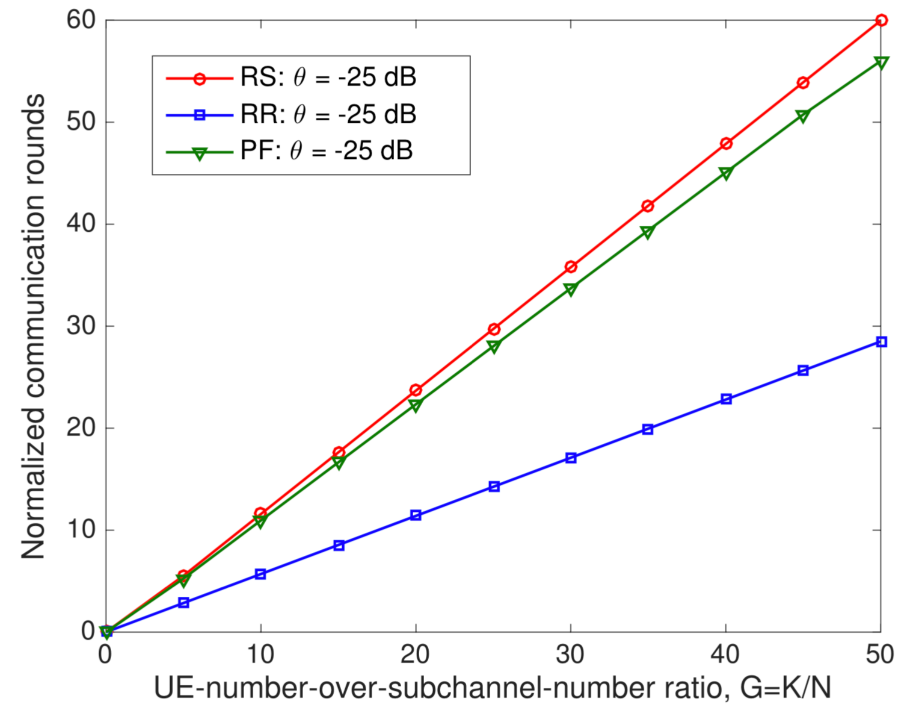
α = path loss exponent
β = precision level at UEs
n = total # exemplars

Numerical Example

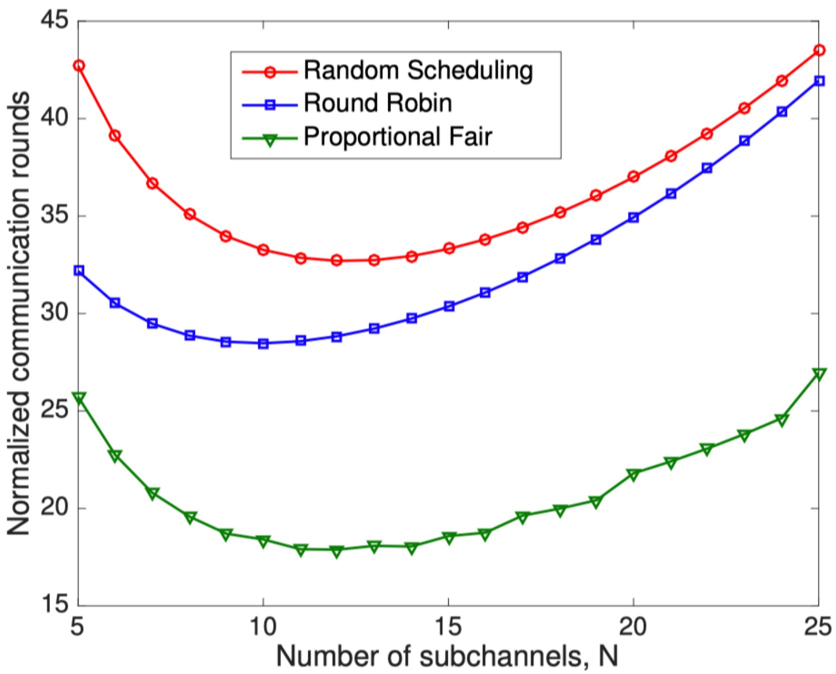
- Each AP has 100 UEs and 20 subchannels
- PF works the best in high SINR condition
- RR works the best in low SINR condition

- High SINR vs low SINR threshold





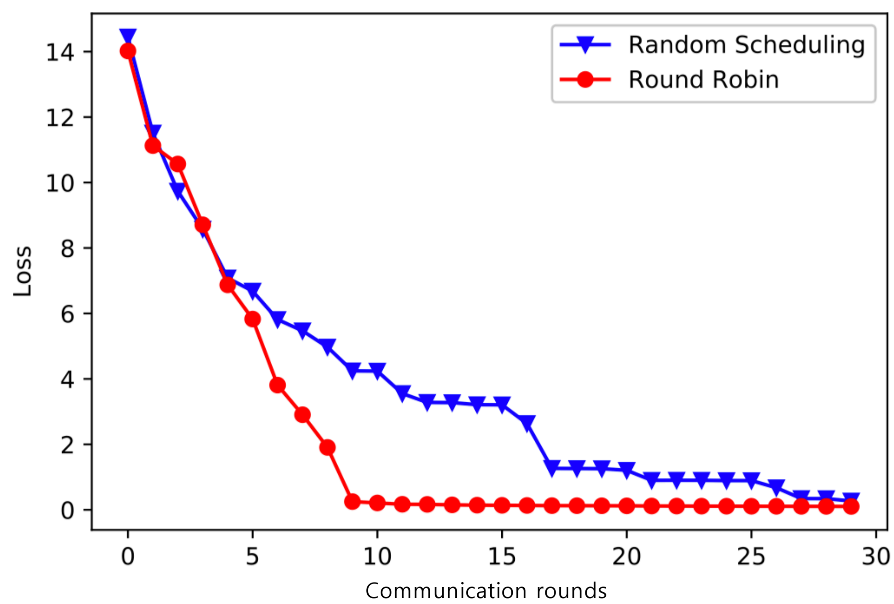
Effect of Channel Bandwidth

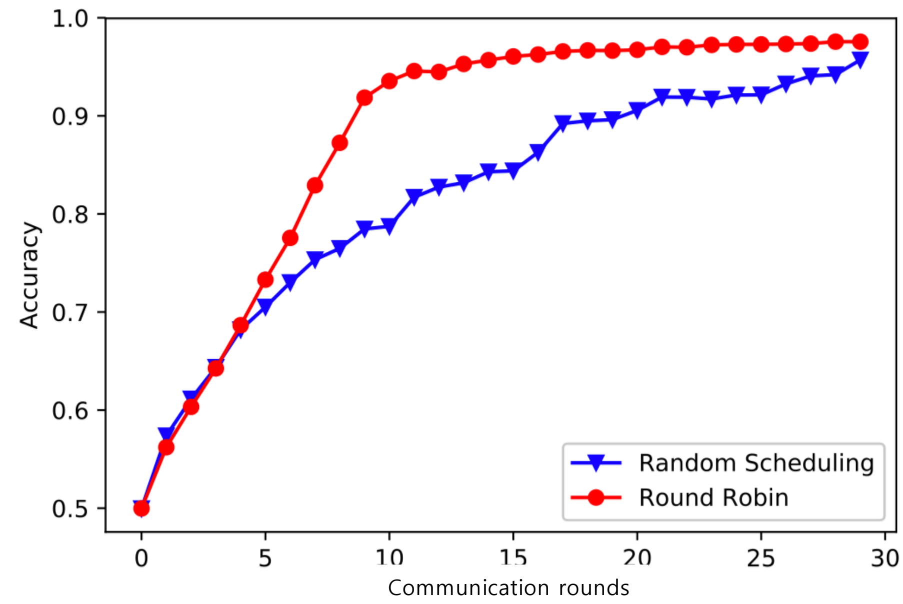


- The total amount of spectrum is fixed
- With more subchannels, more UEs can be selected for update in each communication round, and vice versa
- Increasing the number of subchannels decreases the bandwidth per subchannel
- An optimal number of subchannels exist for each of the three schemes

A Conclusion: Scheduling Protocol Matters

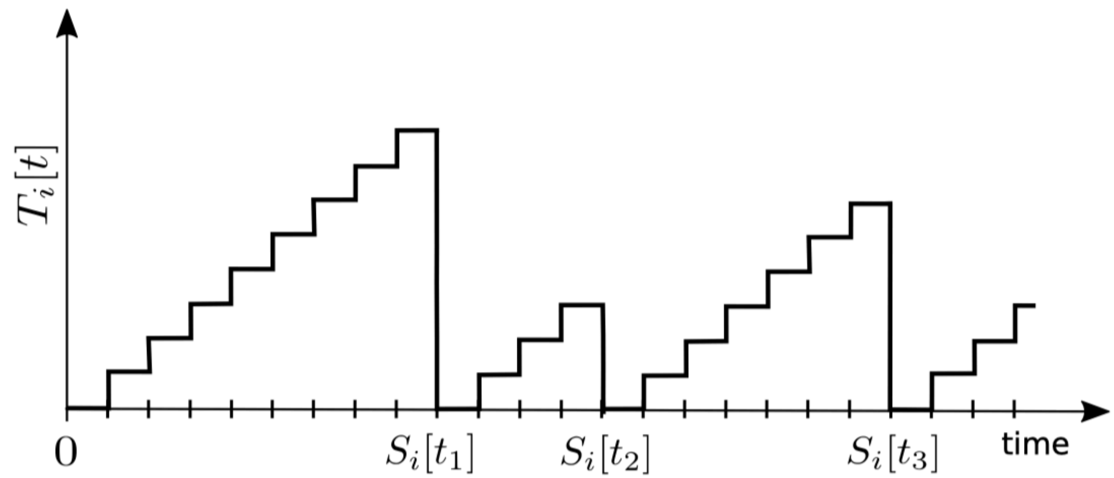
- SVM on MNIST data set
- 10,000 sample points distributed on 100 devices
- Select 20 out of 100 each global aggregation





Can we optimize scheduling?

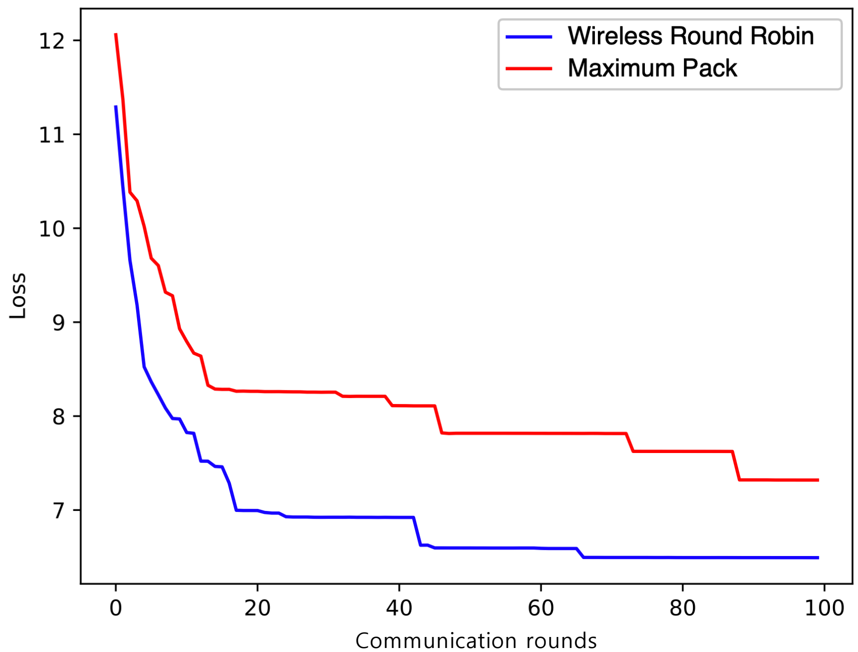
Design Metric: Age of Information

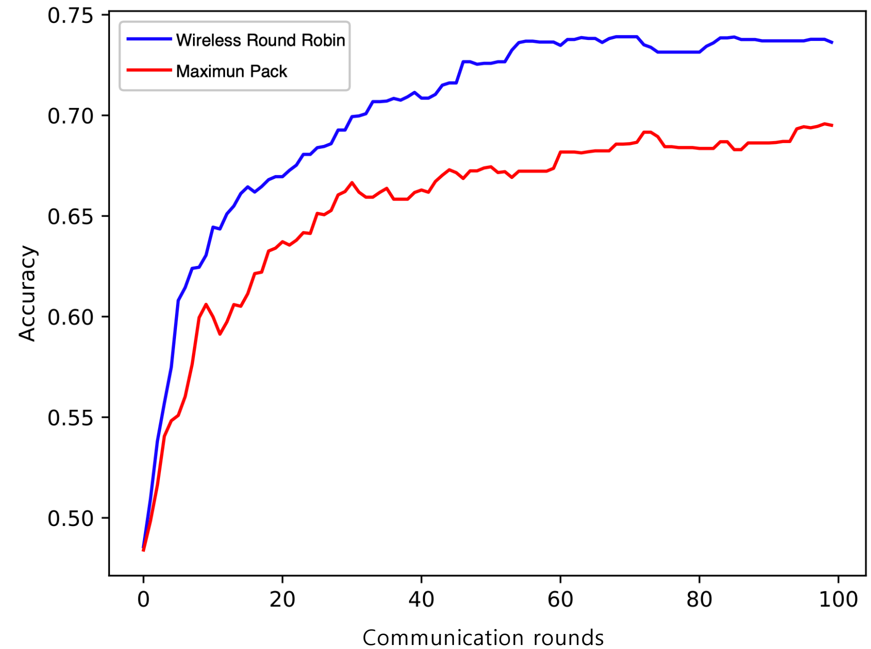


- Metric
  - Age-of-Information (AoI) at a UE i
    - During each communication round, if selected, the AoI drop to 0. Otherwise, the AoI increases by 1: $T_i[t+1] = (T_i[t]+1)(1-S_i[t]), \quad S_i[t] \in \{0,1\}$

Numerical Example
Constrained Minimization of Average AoI*

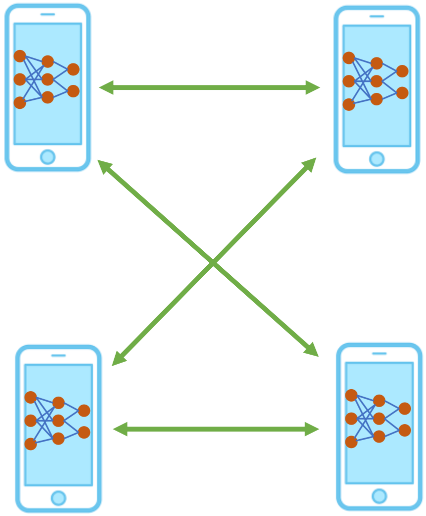
- SVM on MNIST data set
- 10,000 sample points distributed on 100 devices
- Available subchannels: 20





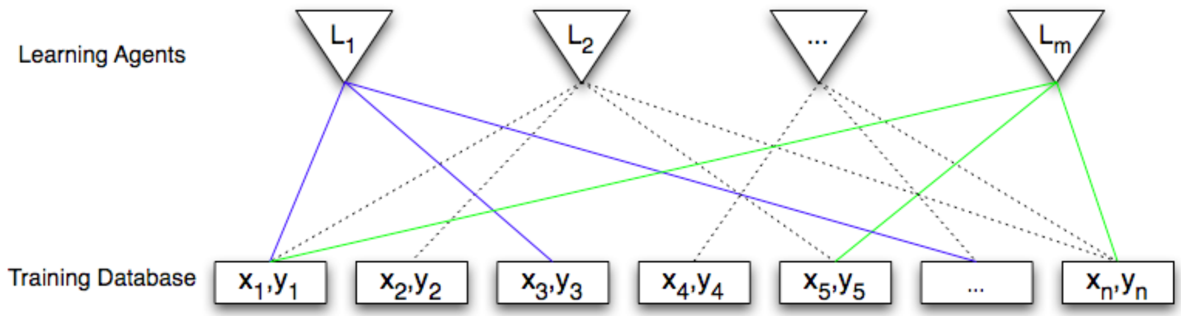
* H. H. Yang, Y. Fu, A. Arafa, T. Q. S. Quek, and H. V. Poor, "Age-Based Scheduling for Federated Learning in Mobile Edge Networks", *Proc. IEEE ICASSP 2020*, to appear.

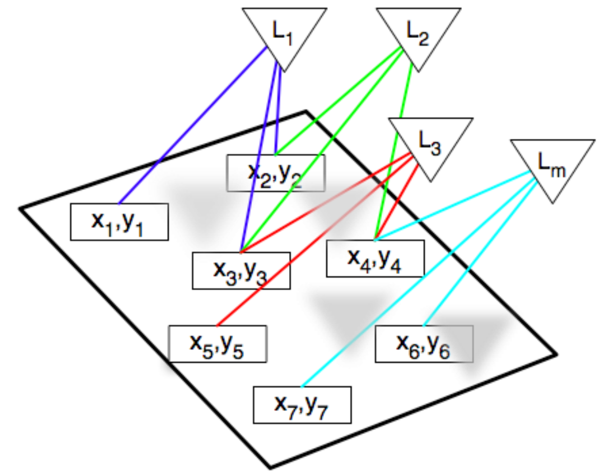
Decentralized Learning (Briefly)



A General Model for Distributed Learning

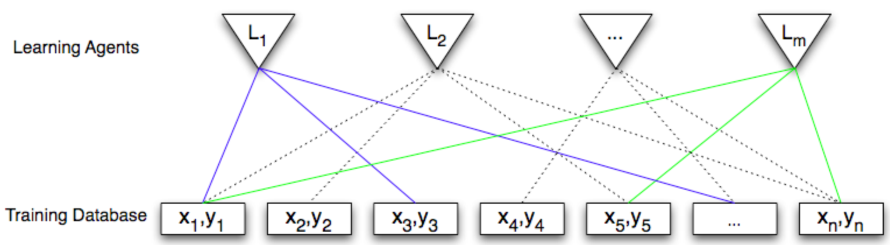
- *m* learning agents (e.g., smart sensors)
- *n* training examples $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$





- Special cases: centralized learning *(m = 1)* & decentralized learning *(m = n)*

Collaboration



$$\hat{f}_1 = \arg\min_{f \in \mathcal{H}_K} \frac{1}{|N_1|} \sum_{j \in N_1} (f(\mathbf{x}_j) - y_j)^2 + \lambda_1 \|f\|^2_{\mathcal{H}_K}$$

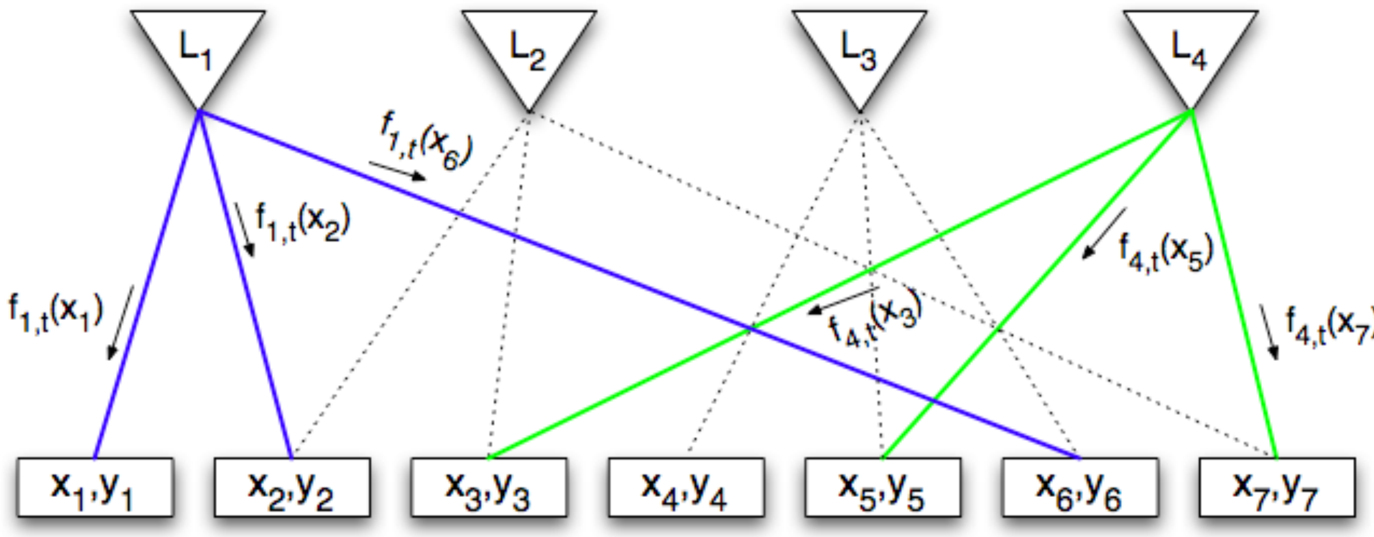
$$\hat{f}_m = \arg\min_{f \in \mathcal{H}_K} \frac{1}{|N_m|} \sum_{j \in N_m} (f(\mathbf{x}_j) - y_j)^2 + \lambda_m \|f\|^2_{\mathcal{H}_K}$$

- Local learning requires only local communication.
- However, it leads to local incoherence, which is undesirable.
- Can agents collaborate to gain coherence, while retaining the efficiency of locality? Yes! *

* J. Predd, S. Kulkarni and H. V. Poor, "A Collaborative Training Algorithm for Distributed Learning," *IEEE Trans. Inf. Theory* **55**(4) 1856-71, 2009.

A Collaborative Algorithm

$$f_{1,t} = \arg\min_{f \in \mathcal{H}_K} \sum_{j \in \{1,2,6\}} (f(\mathbf{x}_j) - y_j)^2 + \lambda_1 \| f - f_{1,t-1} \|^2_{\mathcal{H}_K}$$

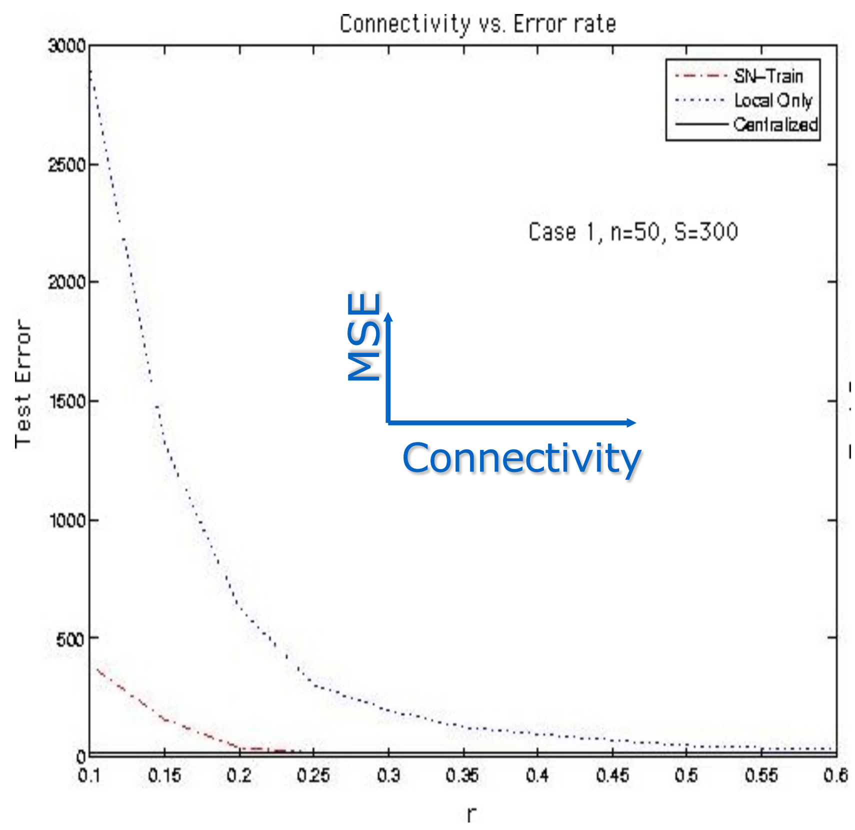


$$f_{4,t} = \arg\min_{f \in \mathcal{H}_K} \sum_{j \in \{3,5,7\}} (f(\mathbf{x}_j) - y_j)^2 + \lambda_4 \| f - f_{4,t-1} \|^2_{\mathcal{H}_K}$$

Converges to a (coherent) relaxation of the global solution.

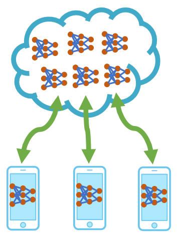
Experiment

- 50 sensors uniform in [-1, 1]
- Sensor i observes $y_i = f(x_i) + n_i$
  - $\{n_i\}$ is i.i.d. $\mathcal{N}(0,1)$
  - regression function f is linear
  - i and j are neighbors: $|x_i - x_j| < r$
- Sensors employ linear kernel

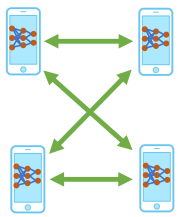


Conclusions

- Mobile networks can be platforms for machine learning
- Federated learning: edge devices (access points) interact with end-user devices to learn common models



- Decentralized learning: end-user devices interact with one another to collaboratively learn models, or actions



Thank You!