Dynamic Resource Optimization for Adaptive Federated Learning empowered by Reconfigurable Intelligent Surfaces

Claudio Battiloro[1] Mattia Merluzzi[2] Paolo Di Lorenzo[1] Sergio Barbarossa[1]

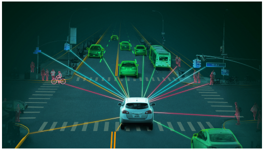
[1] Sapienza University of Rome, Dept. of Information Eng., Elec., and Tlc.
[2]CEA-Leti, Universite Grenoble Alpes

ICASSP, May 7-13 2022 (Virtual), May 22-27 2022 (Onsite)

Federated Learning
Introduction and Applications

- Federated Learning is a technique for training ML models across multiple decentralized edge devices or servers without exchanging local data samples
- FL is a key enabler for Edge Machine Learning, a novel class of cyber-physical systems that exploit the Synergy and Complementarity of Machine Learning and Edge Computing
- Applications: Augmented Reality, Autonomous Driving, Industry 4.0, etc.

(a) AR Visors[1]



(b) Self-Driving Car[2]

(c) Industry 4.0 Concept[3]

[1] https://live.cdn.sms-group-connects.com/fileadmin/_processed_/0/7/csm_20180823_Virtual_and_Augmented_Reality_in_engineering_b9114c3a9e.jpg

[2] https://researchleap.com/wp-content/uploads/2021/12/AI_Drive_Reasoning-002.png

[3] https://batechnology.it/wp-content/uploads/2021/02/4punto0.jpg

RIS-Aided Federated Learning at the Edge

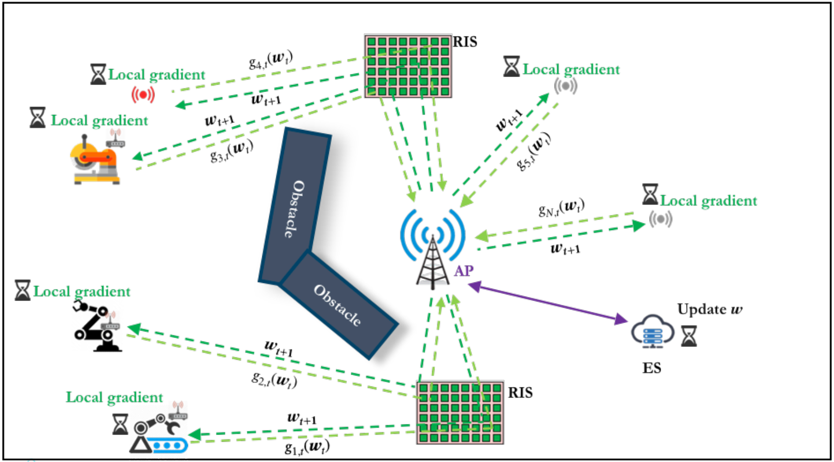
Motivations and state of the art

- Desiderata: Enabling energy-efficient federated learning at the wireless network edge, with latency and learning performance guarantees, in the context of beyond 5G network endowed with Reconfigurable Intelligent Surfaces (RISs).

- Federated Learning (FL):
  - FL seminal papers [Kone15][Kone16]
  - Communication-efficient FL [Kone16] [Ha19] [Wang19]
  - Deep FL [Bren16]
  - Static joint learning and wireless allocation in FL [Chen19] [Tran19]
  - Dynamic user selection for FL [Chen20]
  - FL & RISs [Ni20][Liu21]

- Contribution: Novel dynamic optimization framework for adaptive federated learning in the context of beyond 5G network endowed with RISs, jointly encompassing radio and computation aspects in order to strike the best trade-off between energy, latency, and performance of the FL task.

System Model
Scenario



System Model
Federated Learning Task

- N edge devices and an AP equipped with an edge server
- Consider the learning problem in the unknown model variable $\boldsymbol{w}$

$$\min_{\mathbf{w}} \sum_{i=1}^{N} \mathbb{E}\{J_i(\mathbf{w}; \boldsymbol{x}_i, y_i)\}$$

- At each t, the edge devices compute $\nabla J_i(\mathbf{w}; \boldsymbol{x}_{i,t}, y_{i,t})$ over a batch of data $\mathcal{B}_t$ of size $|\mathcal{B}_t| = B_t$ and upload them to the AP
- The edge server computes $\mathbf{w}_{t+1}$ via *any* gradient-based algorithm and fed it back to the devices. In general:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \mu \cdot f\left(\sum\nolimits_{i \in \mathcal{S}_t} \nabla J_i(\mathbf{w}; \boldsymbol{x}_{i,t}, y_{i,t})\right)$$

System Model

RIS-Enhanced Communications

- K passive RISs with M reflecting elements
- The phase of each element is quantized using b_r bits
- Each element has a complex reflection coefficient:

$$v_{k,l,t} \in \mathcal{R} = \left[0, \left\{e^{j\frac{2n\pi}{2^{b_r}}}\right\}_{n=0}^{2^{b_r}-1}\right], \quad \forall k, l, t$$

- The RIS-aided uplink transmission rate between user i and the AP:

$$R_{i,t} = B_i \log_2\left(1 + \frac{h_{i,t}(\boldsymbol{v}_t)p_{i,t}}{N_0 B_i}\right),$$

where $h_{i,t}(\boldsymbol{v}_t)$ is the RIS-dependent channel coefficient:

$$h_{i,t}(\boldsymbol{v}_t) = \left| h_{i,t}^a + \sum_{k=1}^{K} \boldsymbol{h}_{i,k,t}^T \operatorname{diag}(\boldsymbol{v}_{k,t})\, \boldsymbol{z}_{i,k,t}^a \right|^2$$

System Model

Latency of Training Iterations

- *Local processing time*: $L_{i,t}^{loc} = \dfrac{B_t J_i}{f_{i,t}}$, where f_i^l is the local CPU frequency
- *Uplink communication time*: $L_{i,t}^{u} = \dfrac{m \cdot b_{i,t}}{R_{i,t}}$, where $R_{i,t}$ is the uplink data rate.
- *Remote processing time*: $L_t^s = \dfrac{C|\mathcal{S}_t|}{f_t^s}$, where f^s is the remote frequency of the server.

The overall latency at time t is given by:

$$L_t = \max_{i \in \mathcal{S}_t} \left\{ L_{i,t}^{loc} + L_{i,t}^{u} \right\} + L_t^s$$

System Model
Power Consumption

- *Power spent for local computation*: $p^c_{i,t} = \gamma_l (f_{i,t})^3$
- *Power spent for uplink transmission*: $p_{i,t} = \frac{B_i N_0}{h_{i,t}} \left[\exp\left(\frac{R_{i,t} \ln 2}{B^u_i} \right) - 1 \right]$
- *Power spent for remote computation*: $p^c_{s,t} = \gamma_r (f^s_t)^3$

The overall power consumption at time t is given by:

$$p^{\text{tot}}_t = \sum_{i=1}^{N} \left(p_{i,t} + p^c_{i,t} \right) + p^c_{s,t}$$

$$\min_{\boldsymbol{\Psi}_t} \quad \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left\{p_\tau^{\text{tot}}\right\}$$

subject to $\quad (a) \ \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left\{L_\tau\right\} \leq \overline{\mathrm{L}}; \rightarrow$ Avg. Latency Constraint

$$(b) \ \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{G_\tau\} \geq \overline{\mathrm{G}}; \rightarrow \text{Avg. Learning Performance Constraint}$$

$$(c) \ \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{\alpha_\tau\} = \overline{\alpha}; \rightarrow \text{Avg. Convergence Rate Constraint}$$

$$\left.\begin{array}{l} b_{i,t} \in \mathcal{C}_i, \ \ \forall i \in \mathcal{S}_t, t; \quad R_i^{\min} \leq R_{i,t} \leq R_{i,t}^{\max}, \ \ \forall i \in \mathcal{S}_t, \ t; \\ f_i^{\min} \leq f_{i,t} \leq f_i^{\max}, \ \ \forall i \in \mathcal{S}_t, \ t; \qquad v_{k,l,t} \in \mathcal{R}, \ \ \forall k,l,t; \\ B_t \in \mathcal{B}, \ \ \forall t; \qquad f^{s,\min} \leq f_t^s \leq f^{s,\max}, \ \ \forall t; \end{array}\right\} \mathcal{X}_t$$

where $\boldsymbol{\Psi}_t = [\boldsymbol{v}_t, \{b_{i,t}\}_{i\in\mathcal{S}_t}, \{R_{i,t}\}_{i\in\mathcal{S}_t}, \{f_{i,t}\}_{i\in\mathcal{S}_t}, f_t^s, B_t]$.

Dynamic Resource Allocation for Federated Learning

Online Estimation of Learning Metrics

- Generally no closed-form expression for for G_t and α_t, especially in Deep-Learning (non-convex) settings
- Online estimation in a totally data-driven fashion
- We assume that either the ES is provided with a validation set $\mathcal{T}$ or the agents can sense an additional batch $\mathcal{T}$ of data, compute their local learning perfomance and send it (one scalar) to the server
- $\widehat{G}_t$ and $\widehat{\alpha}_t$ to estimate online G_t and α_t, e.g. for classification:

$$\widehat{G}_t = \frac{1}{|\mathcal{T}|} \sum_{y \in \mathcal{T}} \mathbb{I}(\widehat{y}_t = y), \quad \widehat{\alpha}_t = \frac{1}{\kappa} \sum_{\tau=t-\kappa}^{\kappa-1} (\widehat{G}_\tau - \widehat{G}_{\tau-1})$$

- *Virtual Queues*:
  - Z_t for the Latency inequality constraint:

$$Z_{t+1} = \max\left\{0, Z_t + \epsilon_z \left(L_t - \overline{\mathrm{L}}\right)\right\}$$

  - Q_t for the accuracy inequality constraint

$$Q_{t+1} = \max\left\{0, Q_t + \epsilon_q \left(\overline{\mathrm{G}} - \widehat{G}_t\right)\right\}$$

  - Y_t for the convergence rate equality constraint:

$$Y_{t+1} = \left[Y_t + \epsilon_{y,t} \left(\widehat{\alpha}_t - \overline{\alpha}\right)\right] \cdot \mathbb{I}\left(\widehat{G}_t \leq \overline{\mathrm{G}}\right)$$

- *Drift-plus-penalty function*:

$$\Delta_t^p = \mathbb{E}\left\{\frac{1}{2}(Z_{t+1}^2 + Q_{t+1}^2 + Y_{t+1}^2) - \frac{1}{2}(Z_t^2 + Q_t^2 + Y_t^2) + V \cdot p_t^{\mathrm{tot}} \middle| \, \boldsymbol{\Phi_t}\right\},$$

where $\boldsymbol{\Phi_t} = [Z_t, Q_t, Y_t]$.

- **Step 1:** $\forall t$, observe $\mathbf{\Phi_t}$ and minimize a DPP upper bound instantaneous values:

$$\min_{\mathbf{\Psi}_t \in \mathcal{X}_t} \quad Z_t \widetilde{L}_t - Q_t \widetilde{G}_t - Y_t \widetilde{\alpha}_t + V \cdot p_t^{\text{tot}}$$

Mixed-integer non linear optimization problem, closed-form solutions for any given $\mathcal{S}_t$, $\{b_{i,t}\}_{i \in \mathcal{S}_t}$ and $\boldsymbol{v}_t$, $\rightarrow$ Find $\mathcal{S}_t$, $\{b_{i,t}\}_{i \in \mathcal{S}_t}$ and $\boldsymbol{v}_t$ with the proposed two-stage greedy selection, setting:

$$R_{i,t} = \left[\frac{2B_i}{\ln(2)} W\left(\frac{\ln(2)}{B_i}\sqrt{\frac{Z_t\, m \cdot b_{i,t}\, h_{i,t}(\boldsymbol{v}_t)}{2VN_0}}\right)\right]_{R_i^{\min}}^{R_{i,t}^{\max}}$$

$$f_{i,t} = \left[\left(\frac{Z_t B_t J_i}{3\gamma_i V}\right)^{\frac{1}{4}}\right]_{f_i^{\min}}^{f_i^{\max}} \qquad f_t^r = \left[\left(\frac{Z_t C\,|\mathcal{S}_t|}{3\gamma_s V}\right)^{\frac{1}{4}}\right]_{f^{r,\min}}^{f^{r,\max}}$$

- **Step 2:** Update Z_t, Q_t, Y_t.

Greedy Selection of $\{\boldsymbol{v}_{k,t}\}_{k=1}^{K}$ and $\mathcal{S}_t$

- **Stage 1** $\rightarrow$ Selection of RISs coefficients $\{\boldsymbol{v}_{k,t}\}_{k=1}^{K}$:
  - The method greedily selects $\{\boldsymbol{v}_{k,t}\}_{k=1}^{K}$ to maximize:

$$\Delta^R(\{\boldsymbol{v}_{k,t}\}_{k=1}^{K}) = \sum_{i=1}^{N} \delta_{i,t} \left| h_{i,t}^a + \sum_{k=1}^{K} \boldsymbol{h}_{i,k,t}^T \,\mathrm{diag}(\boldsymbol{v}_{k,t})\, \boldsymbol{z}_{i,k,t}^a \right|^2$$

where $\delta_{i,t} = \dfrac{1/|h_{i,t}^a|^2}{\sum_{i=1}^{N} 1/|h_{i,t}^a|^2}$

  - Polynomial complexity in K, M and $|\mathcal{R}|$

- **Stage 2** $\rightarrow$ Selection of transmitting set $\mathcal{S}_t$:
  - For each $B_t \in \mathcal{B}$, the method starts from $\mathcal{S}_t = \emptyset$ and iteratively selects the most convenient $\{b_{i,t}\}_{i=1}^{N}$ and the corresponding edge resources
  - The method keeps adding devices until the objective decreases
  - Polynomial complexity in N, $\max_i\{|\mathcal{C}_i|\}$, $|\mathcal{B}|$

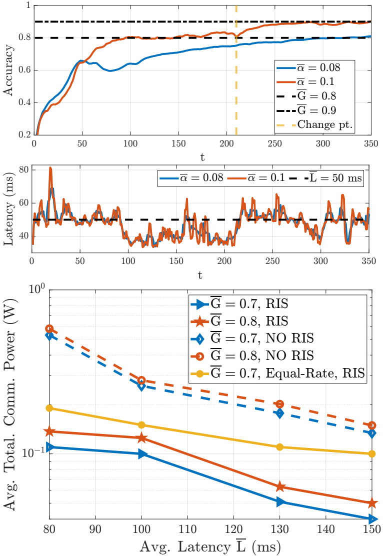
Numerical Results
Simulation Set Up

- $N = 9$ devices and one AP equipped with an edge server
- Classification task on the MNIST dataset (10 classes)
- CNN with 4 convolutional layers ($\sim$100K parameters)
- ADAM Optimizer, learning rate 0.001, forgetting factors $\beta_1 = 0.9$, and $\beta_2 = 0.99$
- One RIS equipped with 1-bit discrete phase shifters
- The channels are generated using the ABG model, using a carrier frequency equal to 6 GHz, with a unit variance Rayleigh fading

Numerical Results

Learning and Trade Off Curves

- The method guarantees the prescribed performance in terms of $\overline{\alpha}$ and $\overline{G}$, within $\overline{L}$
- The method reacts promptly to changes in the accuracy requirement
- Baseline given by an equal-rate policy with all the agents always transmitting
- The tradeoff gets worse imposing a stricter $\overline{G}$ requirement
- Significant gain obtained thanks to the presence of the RIS



Conclusions

- We proposed an online strategy for adaptive federated learning empowered by reconfigurable intelligent surfaces (RISs)
- The strategy dynamically minimizes the power expenditure of the system, while guaranteeing target learning performance and latency constraints in a fully data driven fashion
- The strategy allows the exploration of a new trade-off of communication networks, including power expenditure, delay, and learning performance
- Numerical results on federated training of Deep Neural Networks illustrate the advantages obtained by the proposed strategy and by the usage of RISs