Politecnico di Torino

Federated Multi-Task Learning

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, Ameet Talwalkar

31° Conference on Neural Information Processing Systems (NIPS 2017)

Riccardo Zaccone
s269240@studenti.polito.it

MLDL
A.A 2020/2021

How to use this presentation

- Mainly as presentation aid, but enriched with other resources to be used as learning tool;
- GitHub repository:
    - A copy of the original paper, with my notes;
    - The .pptx and .pdf file of the presentation submitted for the course and used to explain the paper to the class;
    - The .mp4 of the final paper presentation video submitted for evaluation;
    - Cited papers, with notes and references;

Outline

- Introduction
  - The federated scenario
  - What this paper is about
- Related work
- Federated Multi-Task Learning
  - Theory;
  - Experiments;
- Conclusion and Citing works

Introduction

- The Federated Scenario
- What this paper is about

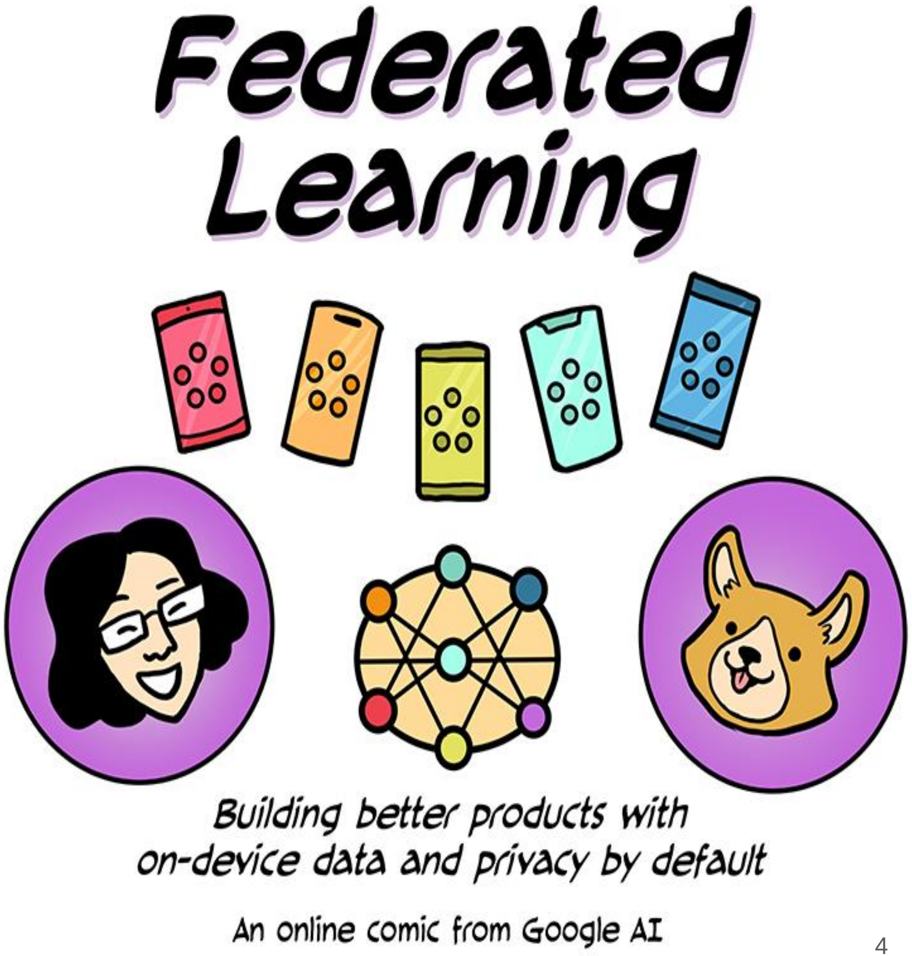


Image taken from: https://federated.withgoogle.com/

The Federated Scenario

- *Def:*

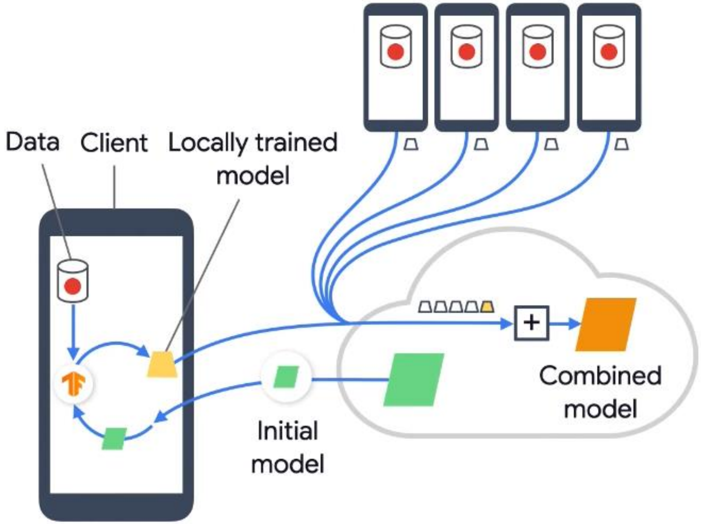
Federated learning is a machine learning setting where **multiple entities** (clients) collaborate in solving a machine learning problem, under the **coordination** of a central server or service provider. Each client's raw **data is stored locally and not exchanged or transferred;** instead, **focused updates** intended for immediate aggregation are used to achieve the learning objective.

- Key points:
  - Entities involved are **heterogeneous**, and they cannot be controlled by the server: it is «**federated**», not «**distributed**»!
  - Entities can have data with different underlying distribution **(IIDness)**;
  - **Privacy** is paramount.

The Federated Scenario

- System challenges:
  - Large number of computationally heterogeneous nodes;
  - Dealing with stragglers;
  - Robustness to faulty clients (or network problems);

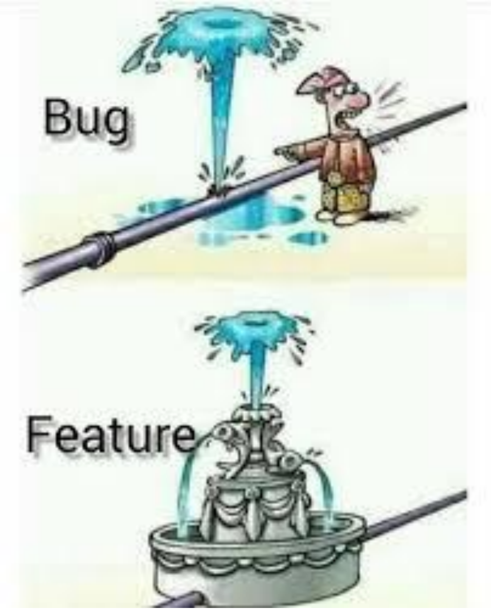
- Statistical challenges:
  - Non-IDDness of local datasets (global model scenario);
  - Capturing the relationships among nodes and their associated distribution (multi-task scenario).



What this paper is about

- Is learning a single global model the best possible?
  - Non IID-ness could be itself information to be learnt, a.k.a similar devices can have similar data distributions;

- How to learn multi-task in the federated scenario?
  - Taking into account high communication cost, stragglers and fault tolerance;
  - While preserving privacy.



Related Work

- **Multi-Task Learning:** the goal is to learn models for multiple related tasks simultaneously. Currently proposed methods for distributed MTL (discussed below) do not adequately address the systems challenges associated with federated learning.
- **Distributed Multi-Task Learning:**
  - [1, 35, 54, 55]: the proposed methods do not allow for flexibility of communication versus computation;
  - [23] and [7] allow for asynchronous updates to help mitigate stragglers, but do not address fault tolerance;
  - [30] does not explore the federated setting, (assumption that the same amount of work is done locally on each node is prohibitive in federated settings);

Points of strenght of this paper

- A novel method, MOCHA, that generalizes the distributed optimization method COCOA [22, 31] in order to address systems challenges associated with network size and node heterogeneity;
- Convergence guarantees for MOCHA that carefully consider the unique systems challenges and provide insight into practical performance;

Federated Multi-Task Learning

- General MTL setup
- MOCHA
- Discussion on assumptions

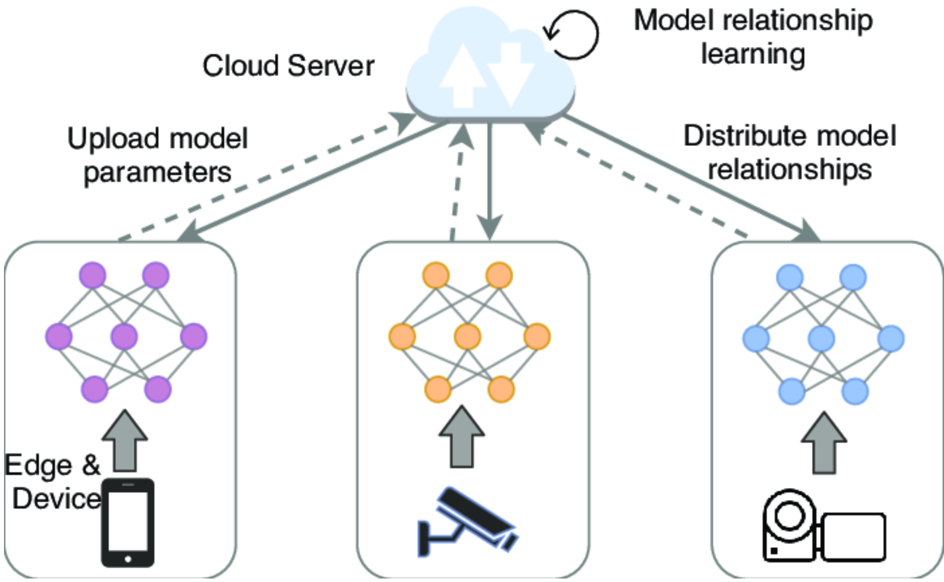


Image taken from: https://www.researchgate.net

General MTL setup

$$\min_{\mathbf{W},\mathbf{\Omega}} \left\{ \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \mathbf{x}_t^i, y_t^i) + \mathcal{R}(\mathbf{W}, \mathbf{\Omega}) \right\}, \quad (1)$$

- **m**: number of nodes;
- **nt**: number of examples of t-th node;
- **lt**: convex loss function of the t-th node;
- **W**: a dxm matrix whose t-th column is the weight vector for the t-th task
- **Ω**: a mxm matrix that models relationships among tasks (nodes);
- **R(W, Ω)**: promotes some suitable structure among the tasks

General MTL setup - observations

$$\min_{\mathbf{W},\boldsymbol{\Omega}} \left\{ \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \mathbf{x}_t^i, y_t^i) + \mathcal{R}(\mathbf{W}, \boldsymbol{\Omega}) \right\}, \tag{1}$$

- The objective function in general is not convex in **W** and **Ω;**
- When fixing **Ω**, updating **W** depends on both the data X, which is distributed across the nodes, and the structure **Ω**, which is known centrally;
- When fixing **W**, optimizing for **Ω** only depends on **W** and not on the data **X.**

- =>solving for **Ω** is not dependent on data, can be computed centrally, so the method focuses on techniques for updating **W**;

MOCHA

$$\min_{\mathbf{W},\boldsymbol{\Omega}} \left\{ \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t(\mathbf{w}_t^T \mathbf{x}_t^i, y_t^i) + \mathcal{R}(\mathbf{W}, \boldsymbol{\Omega}) \right\}, \quad (1)$$

- Dual problem formulation:

$$\min_{\boldsymbol{\alpha}} \left\{ \mathcal{D}(\boldsymbol{\alpha}) := \sum_{t=1}^{m} \sum_{i=1}^{n_t} \ell_t^*(-\boldsymbol{\alpha}_t^i) + \mathcal{R}^*(\mathbf{X}\boldsymbol{\alpha}) \right\}, \quad (3)$$

- Data-local quadratic subproblems: find updates **Δαt** to the dual variables in α corresponding to a single node t, and only require accessing data which is available locally, i.e., **Xt** for node t.

$$\min_{\Delta\boldsymbol{\alpha}_t} \mathcal{G}_t^{\sigma'}(\Delta\boldsymbol{\alpha}_t; \mathbf{v}_t, \boldsymbol{\alpha}_t) := \sum_{i=1}^{n_t} \ell_t^*(-\boldsymbol{\alpha}_t^i - \Delta\boldsymbol{\alpha}_t^i) + \langle \mathbf{w}_t(\boldsymbol{\alpha}), \mathbf{X}_t \Delta\boldsymbol{\alpha}_t \rangle + \frac{\sigma'}{2} \|\mathbf{X}_t \Delta\boldsymbol{\alpha}_t\|_{\mathbf{M}_t}^2 + c(\boldsymbol{\alpha}), \quad (4)$$

$$\theta_t^h := \frac{\mathcal{G}_t^{\sigma'}(\Delta\boldsymbol{\alpha}_t^{(h)}; \mathbf{v}^{(h)}, \boldsymbol{\alpha}_t^{(h)}) - \mathcal{G}_t^{\sigma'}(\Delta\boldsymbol{\alpha}_t^\star; \mathbf{v}^{(h)}, \boldsymbol{\alpha}_t^{(h)})}{\mathcal{G}_t^{\sigma'}(\mathbf{0}; \mathbf{v}^{(h)}, \boldsymbol{\alpha}_t^{(h)}) - \mathcal{G}_t^{\sigma'}(\Delta\boldsymbol{\alpha}_t^\star; \mathbf{v}^{(h)}, \boldsymbol{\alpha}_t^{(h)})},$$

MOCHA

- To avoid stragglers, MOCHA provides the t-th node with the flexibility to approximately solve its subproblem, where the quality of the approximation is controlled by a per-node parameter θ_h^t .

- θ_h^t ranges from zero to one, where $\theta_h^t = 0$ indicates an exact solution and $\theta_h^t = 1$ indicates that node t made no progress during iteration h (dropped node).
- MOCHA mitigates stragglers by enabling the t-th node to define its own θ_h^t.
- This new degree of freedom also pose new challenges in providing convergence guarantees for MOCHA

Discussion on assumptions

- Convergence guarantees come at a price:
  - Must assume a non-zero probability of a node sending a result, in any iteration;

$$\mathbb{P}[\theta_t^h = 1] \leq p_{\max} < 1$$

  - The quality of the returned result is, on average, better than the previous iterate.

$$\hat{\Theta}_t^h := \mathbb{E}[\theta_t^h | \mathcal{H}_h, \theta_t^h < 1] \leq \Theta_{\max} < 1.$$

- Having the dual vector history until the beginning of iteration h:

$$\mathcal{H}_h := (\boldsymbol{\alpha}^{(h)}, \boldsymbol{\alpha}^{(h-1)}, \cdots, \boldsymbol{\alpha}^{(1)})$$

MOCHA and COCOA

- Prior work of COCOA may suffer from severe straggler effects in federated settings, as it requires a fixed $\theta_h^t = \theta$ across all nodes and all iterations while still maintaining synchronous updates, and it does not allow for the case of dropped nodes ($\theta := 1$);
- explore the differences and connections between asynchronous methods and approximation-based, synchronous methods like MOCHA in future work.

Experiments

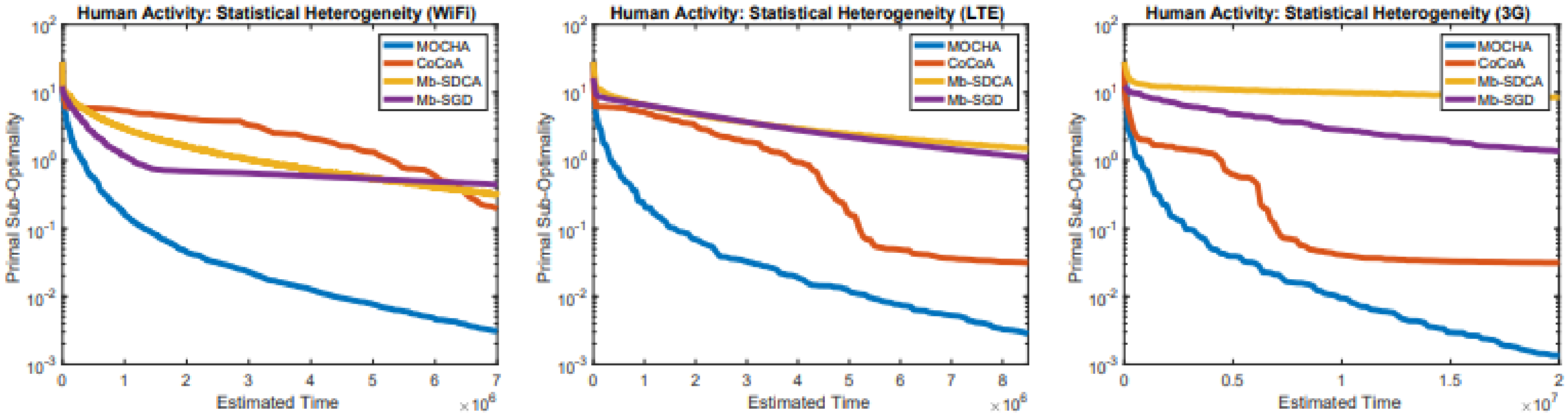
- For each dataset, multi-task learning significantly outperforms the other models in terms of achieving the lowest average error across tasks.

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)**

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

Experiments – 1/3

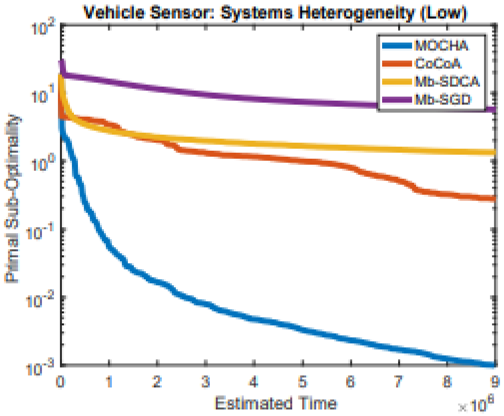
- **Statistical Heterogeneity:** while the performance degrades for mini-batch methods in high communication regimes, MOCHA and COCOA are robust to high communication. However, COCOA is significantly affected by stragglers—because θ is fixed across nodes and rounds, difficult subproblems adversely impact convergence.

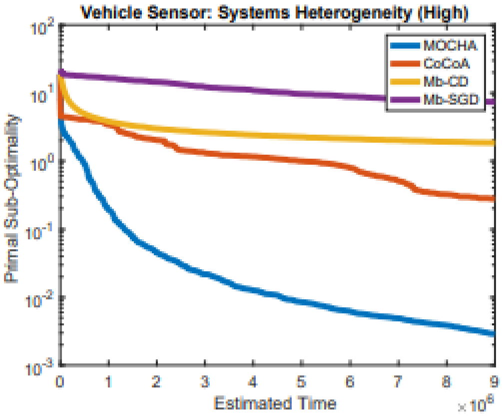


The performance of MOCHA compared to other distributed methods for the W update of (1).

Experiments – 2/3

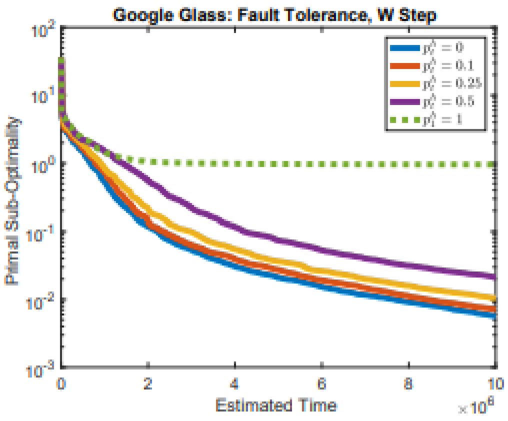
- **System Heterogeneity:** systems heterogeneity is simulated by randomly choosing the number of local iterations for MOCHA or the mini-batch size for mini-batch methods, between 10% and 100% of the minimum number of local data points for high variability environments, to between 90% and 100% for low variability.

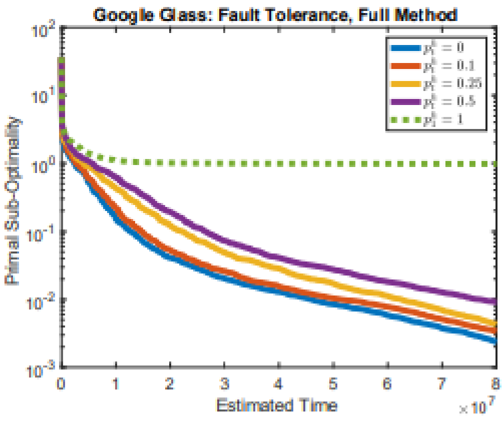




Experiments – 3/3

- **Fault Tolerance:** In MOCHA this is simulated taking $\theta_h^t := 1$. the performance of MOCHA is robust to relatively high values of p_h^t; however, if one of the nodes never sends updates (i.e., $p_1^h := 1$ for all h, green dotted line), the method does not converge to the correct solution.





Conclusion & Citing work

- MOCHA does not apply to non-convex deep learning models:
  - The method relies on alternating bi-convex optimization and is thus only applicable to convex objective functions and limited in its ability to scale to massive client populations.
  - Future work can explore this approach and “convexified” deep learning models [6, 34, 51, 56] in the context of kernelized federated multi-task learning.
- Future works:
  - **Clustered Federated Learning**: it is applicable to general non-convex objectives (in particular deep neural networks) and comes with strong mathematical guarantees on the clustering quality;

Thanks for you attention!

License