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

Introduction

- The Federated Scenario
- What this paper is about

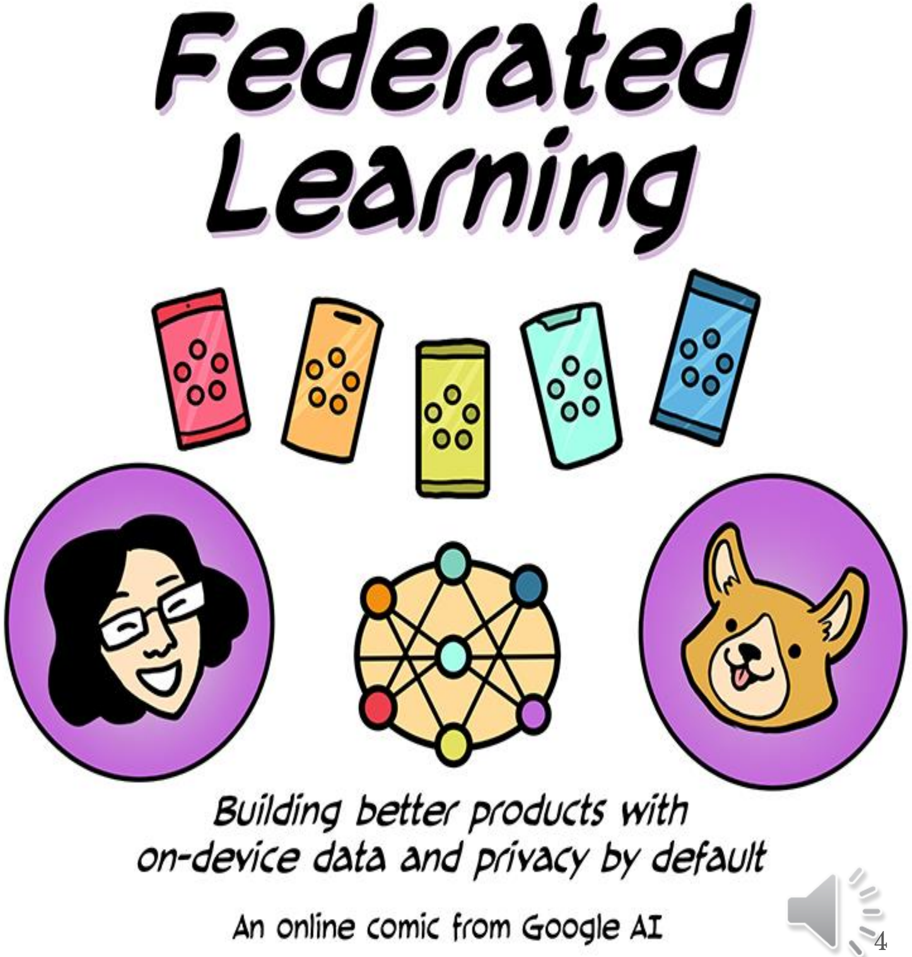

Federated Learning
Building better products with
on-device data and privacy by default
An online comic from Google AI

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

The Federated Scenario

- *Def:*

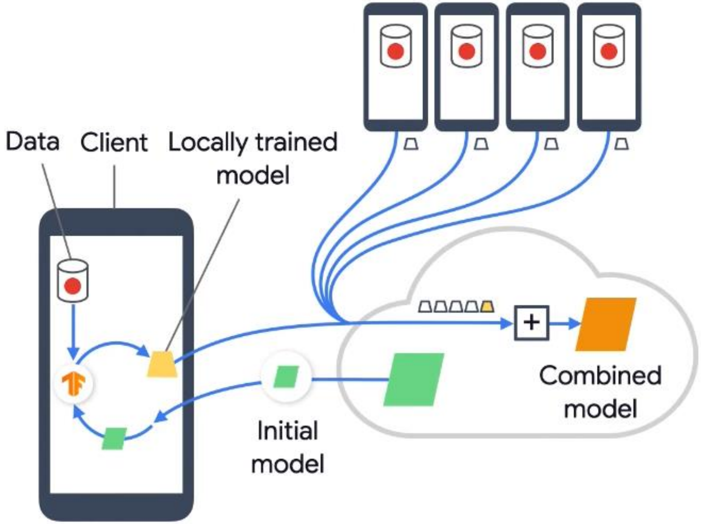
  *"Federated learning is a machine learning setting where **multiple entities** (clients) collaborate in solving a machine learning problem, under the **coordination** of a central server or service provider. Each client's raw **data is stored locally and not exchanged or transferred;** instead, **focused 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

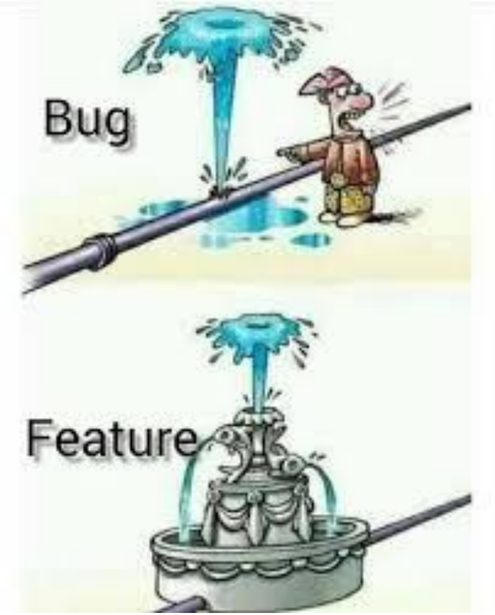
- System challenges:
  - High communication cost;
  - Dealing with stragglers;
  - Robustness to faulty clients (or network problems);

- 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;



Related Work

- 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);

Federated Multi-Task Learning

- General MTL setup
- MOCHA
- Discussion on assumptions

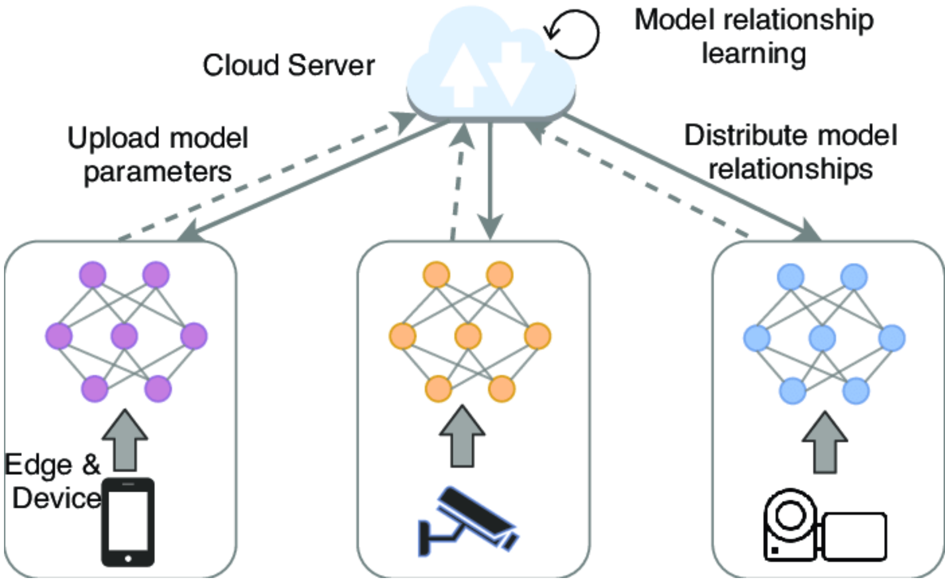


Image taken from: https://www.researchgate.net

General MTL setup

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

- $\boldsymbol{m}$: number of nodes;
- $\boldsymbol{n_t}$: number of examples of *t-th* node;
- $\boldsymbol{l_t}$: 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

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

- 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

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

- Dual problem formulation:

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

- Data-local quadratic subproblems: find updates $\boldsymbol{\Delta\alpha_t}$ to the dual variables in α corresponding to a single node t, and only require accessing data which is available locally, i.e., $\boldsymbol{X_t}$ for node t.

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

How MOCHA avoids stragglers

- The *t-th* node has the flexibility to *approximately* solve its subproblem, by controlling θ_h^t in range [0,1]:
    - $\theta_h^t = 0$ indicates an exact solution;
    - $\theta_h^t = 1$ indicates that node *t* made no progress during iteration *h* (dropped node).

- 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.

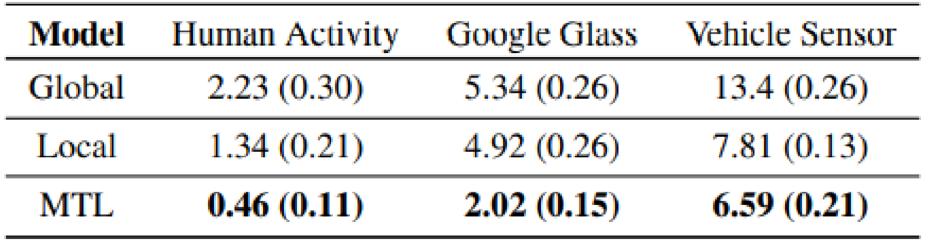
$$\hat{\Theta}_t^h := \mathbb{E}[\theta_t^h | \mathcal{H}_h, \theta_t^h < 1] \leq \Theta_{\max} < 1.$$

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

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

Experiments

- **Take-away message:** for each dataset chosen, multi-task learning significantly outperforms the other models in terms of achieving the lowest average error across tasks.

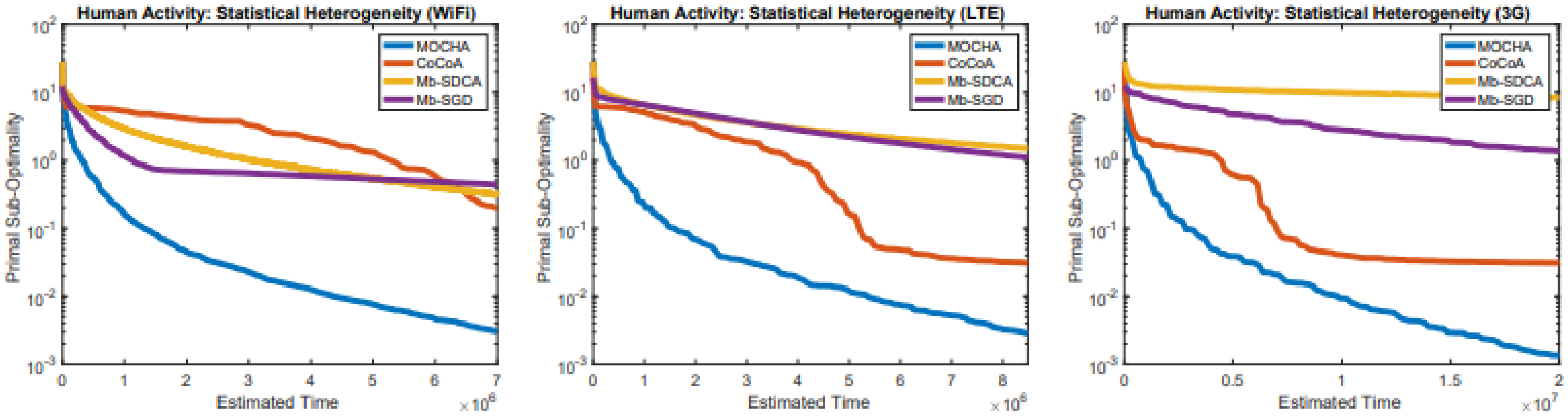


Model	Human Activity	Google Glass	Vehicle Sensor
Global	2.23 (0.30)	5.34 (0.26)	13.4 (0.26)
Local	1.34 (0.21)	4.92 (0.26)	7.81 (0.13)
MTL	**0.46 (0.11)**	**2.02 (0.15)**	**6.59 (0.21)**

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

Experiments – 1/3

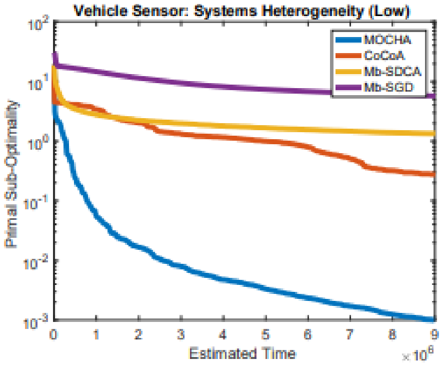
- **Statistical Heterogeneity:** in high communication regimes, MOCHA and COCOA are robust to high communication. COCOA is significantly affected by stragglers—because θ is fixed across nodes and rounds;

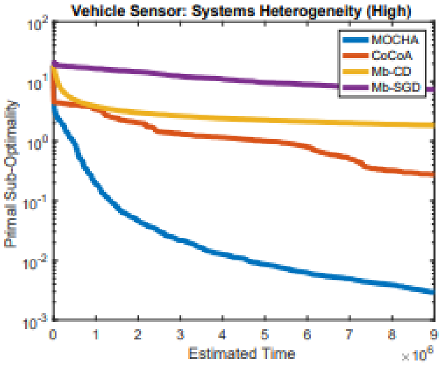


The performance of MOCHA compared to other distributed methods for the W update of (1).

Experiments – 2/3

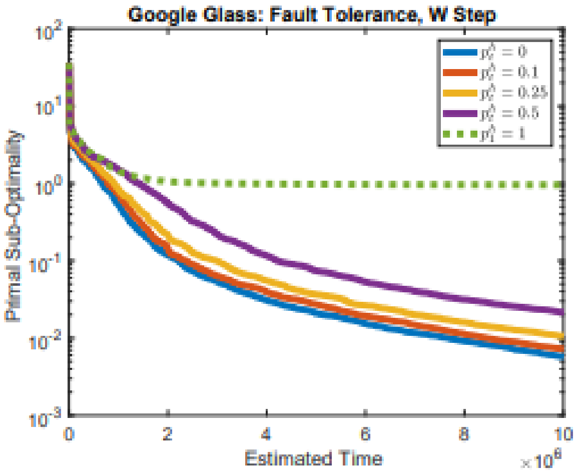
- **System Heterogeneity:** systems heterogeneity is simulated by randomly choosing the number of local iterations wrt the minimum number of local data points:
  - between 10% and 100% for high variability;
  - between 90% and 100% for low variability.

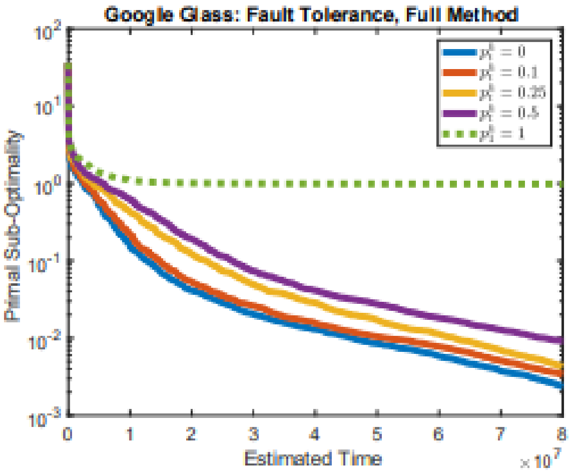




Experiments – 3/3

- **Fault Tolerance:** simulated taking $\theta_h^t = 1$. 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

- A novel method, MOCHA, that generalizes the distributed optimization method COCOA [22, 31] with convergence guarantees;

- MOCHA does not apply to non-convex deep learning models:
  - Future work can explore this approach and “convexified” deep learning models [6, 34, 51, 56] in the context of kernelized federated multi-task learning;
  - See **Clustered Federated Learning**: it is applicable to general non-convex objectives and comes with strong mathematical guarantees on the clustering quality;

Thanks for you attention!

References

- [1] A. Ahmed, A. Das, and A. J. Smola. Scalable hierarchical multitask learning algorithms for conversion optimization in display advertising. In Conference on Web Search and Data Mining, 2014.
- [7] I. M. Baytas, M. Yan, A. K. Jain, and J. Zhou. Asynchronous multi-task learning. In International Conference on Data Mining, 2016.
- [22] M. Jaggi, V. Smith, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan. Communication-Efficient Distributed Dual Coordinate Ascent. In Neural Information Processing Systems, 2014.
- [23] X. Jin, P. Luo, F. Zhuang, J. He, and Q. He. Collaborating between local and global learning for distributed online multiple tasks. In Conference on Information and Knowledge Management, 2015.
- [30] S. Liu, S. J. Pan, and Q. Ho. Distributed multi-task relationship learning. Conference on Knowledge Discovery and Data Mining, 2017.
- [31] C. Ma, V. Smith, M. Jaggi, M. I. Jordan, P. Richtárik, and M. Takáˇc. Adding vs. averaging in distributed primal-dual optimization. In International Conference on Machine Learning, 2015.
- [35] D. Mateos-Núñez and J. Cortés. Distributed optimization for multi-task learning via nuclear-norm approximation. In IFAC Workshop on Distributed Estimation and Control in Networked Systems, 2015.
- [55] J. Wang, M. Kolar, and N. Srebro. Distributed multi-task learning with shared representation. arXiv:1603.02185, 2016.

License

You are free to:

Under the following terms: