PhD Open Days

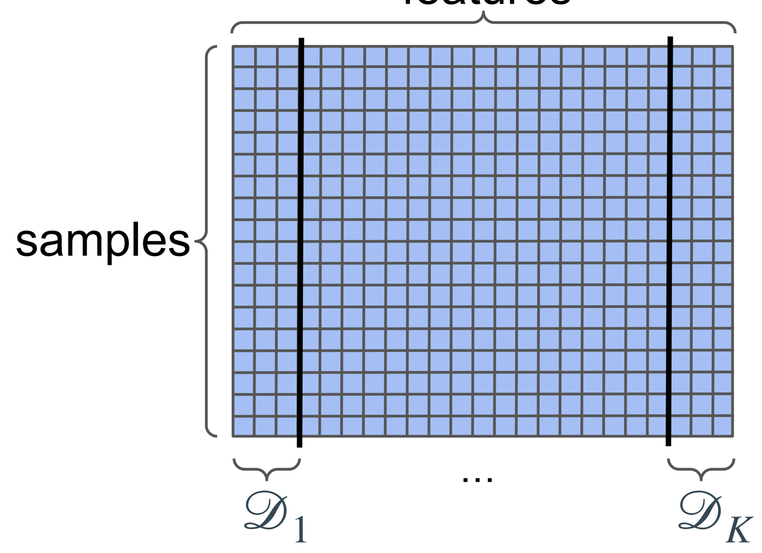
Error Feedback Compressed Vertical Federated Learning

Carnegie Mellon Portugal Double Degree Program

Pedro Valdeira, pvaldeira@cmu.edu

Background on Vertical Federated Learning

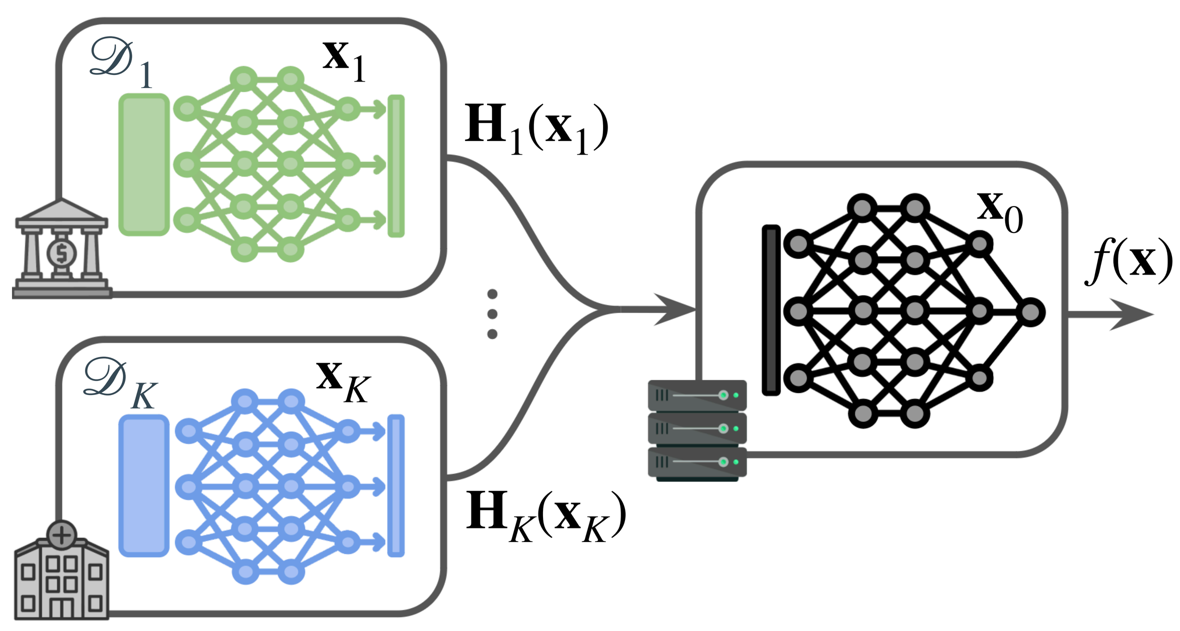
Dataset $\mathscr{D}$ partitioned across K clients,



who want to collaborate to learn the parameters $\mathbf{x} := (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_K)$ of a model:

$$f(\mathbf{x}) := \phi(\mathbf{x}_0, \mathbf{H}_1(\mathbf{x}_1), \dots, \mathbf{H}_K(\mathbf{x}_K)),$$

where $\mathbf{H}_k(\mathbf{x}_k) := \mathbf{H}_k(\mathbf{x}_k; \mathscr{D}_k)$, without sharing local data.



Communication is often the bottleneck, so communication efficiency is a major challenge.

Prior art

Algorithm : Compressed Vertical Federated Learning [1]

Input : initial point $\boldsymbol{x}^0$, step-size η

for $t = 0, \dots, T-1$ **do**
- Server samples $\mathcal{B}^t \subseteq [N]$ and sends $\mathcal{B}^t$ and $\boldsymbol{x}_0^t$ to all clients
- **for** $k \in [K]_0$ **in parallel do**
  - Compute $\boldsymbol{x}_k^{t+1}$ at k using coordinate descent
  - **if** $k > 0$ **then**
    - Client k computes and sends $\mathcal{C}(\boldsymbol{H}_{k\mathcal{B}^t}(\boldsymbol{x}_k^{t+1}))$ to the server

We say $\mathscr{C}\colon \mathbb{R}^d \to \mathbb{R}^d$ is a (biased) compressor if there exists $\alpha \in (0,1]$ such that

$$\mathbb{E}\|\mathscr{C}(\mathbf{x}) - \mathbf{x}\|^2 \le (1-\alpha)\|\mathbf{x}\|^2, \quad \forall \mathbf{x} \in \mathbb{R}^d.$$

This includes top-k sparsification, quantization,…

Problem of direct compression: compression error does not vanish unless $\alpha \to 1$, requiring a vanishing stepsize to converge at a $\mathscr{O}\left(1/\sqrt{T}\right)$ rate.

EFVFL (our method)

Algorithm : Error Feedback Compressed Vertical Federated Learning

Input : initial point $\boldsymbol{x}^0$, step-size η, initial estimates $\{\boldsymbol{G}_k^0 = \mathcal{C}(\boldsymbol{H}_k(\boldsymbol{x}_k^0))\}$

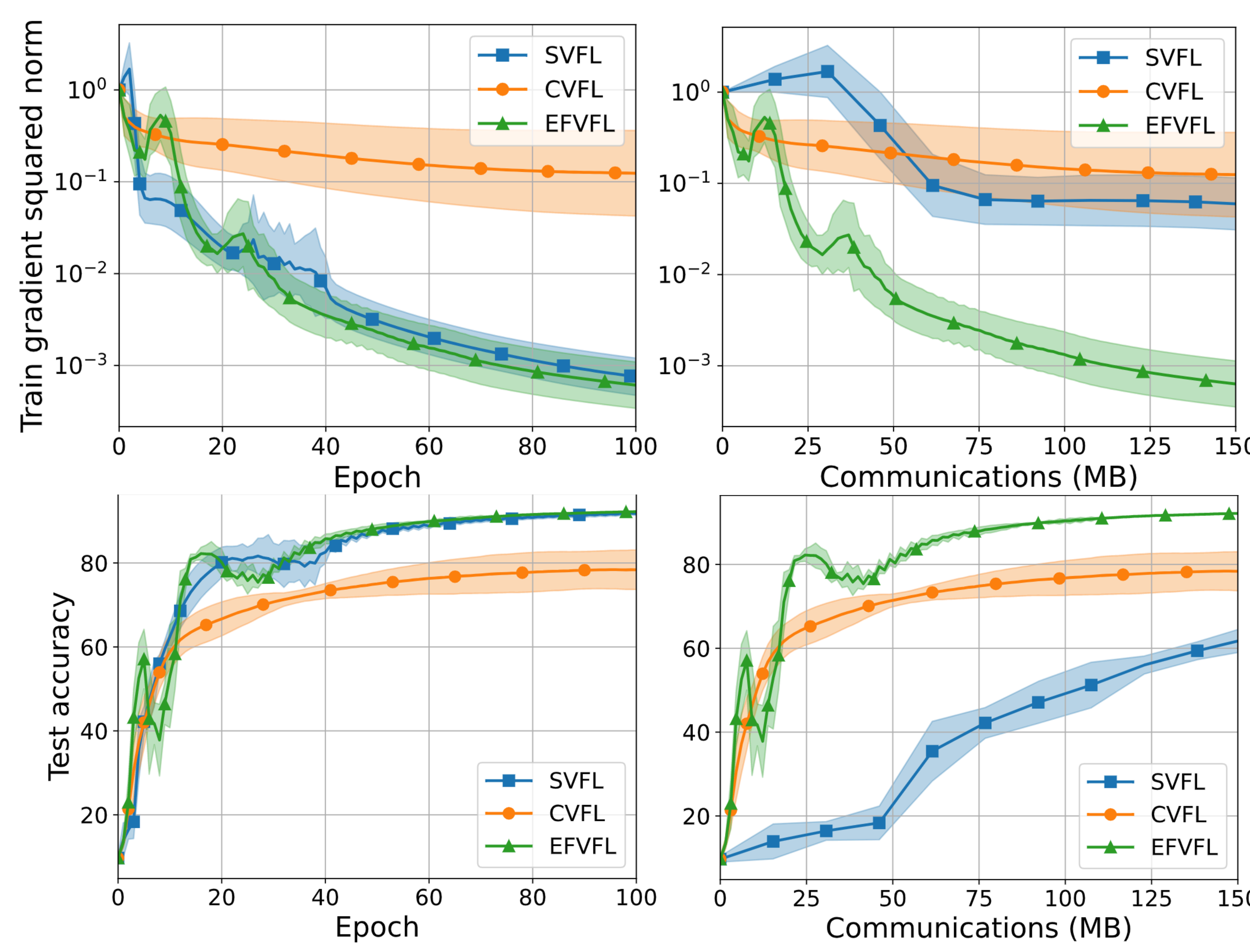
for $t = 0, \dots, T-1$ **do**
- Server samples $\mathcal{B}^t \subseteq [N]$ and sends $\mathcal{B}^t$, $\{\boldsymbol{G}_{k\mathcal{B}^t}^t\}$, and $\boldsymbol{x}_0^t$ to all clients
- **for** $k \in [K]_0$ **in parallel do**
  - Compute $\boldsymbol{x}_k^{t+1}$ at k using a coordinate descent step
  - **if** $k > 0$ **then**
    - Client k computes $\boldsymbol{C}_k^t = \mathcal{C}(\boldsymbol{H}_{k\mathcal{B}^t}(\boldsymbol{x}_k^{t+1}) - \boldsymbol{G}_{k\mathcal{B}^t}^t)$ and sends it to the server
    - Update $\boldsymbol{G}_{k\mathcal{B}^t}^{t+1} = \boldsymbol{G}_{k\mathcal{B}^t}^t + \boldsymbol{C}_k^t$ and $\boldsymbol{G}_{ki}^{t+1} = \boldsymbol{G}_{ki}^t$ for $i \notin \mathcal{B}^t$ at client k and at the server

Convergence guarantees:

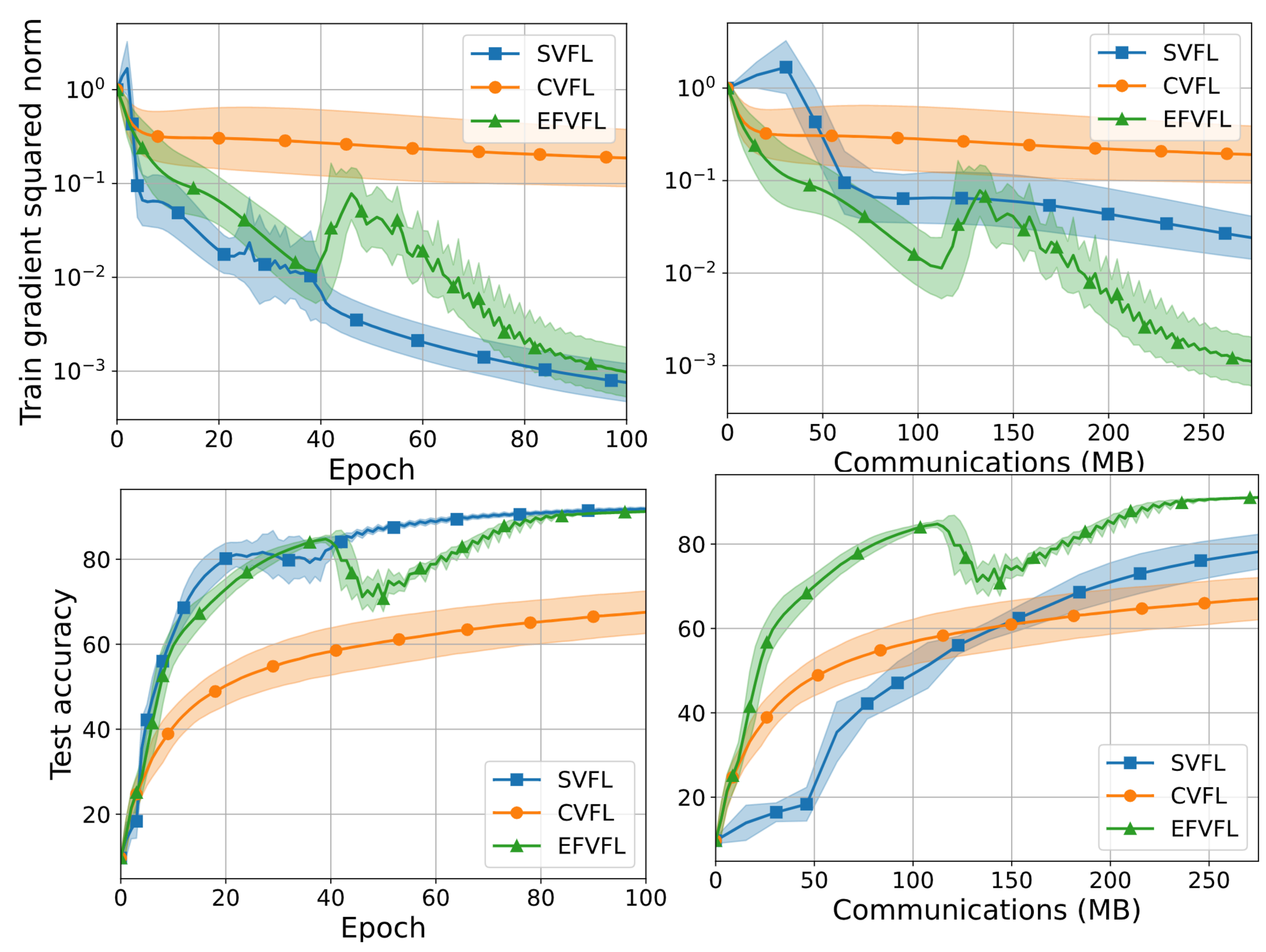
- If f and ϕ are L-smooth and $\mathbf{H}_k$ have a bounded gradient norm, EFVFL converges at a $\mathscr{O}(1/T)$ rate;
- If we further assume the PL condition, we have linear convergence.

Experiments (shallow network to classify MNIST)

Top-k sparsification (sending 10% of entries)



Quantization (5 bits per entry)



Acknowledgements: this work is supported by FCT through the Carnegie Mellon Portugal Program and project UIDB/50009/2020.

References: [1] Castiglia, Timothy J., et al. "Compressed-VFL: Communication-efficient learning with vertically partitioned data." ICML 2022.

TÉCNICO LISBOA

Carnegie Mellon University

Advisors:
João Xavier (IST), Yuejie Chi (CMU), Cláudia Soares (NOVA)

phdopendays.tecnico.ulisboa.pt