FedMix: A Simple and Communication-Efficient Alternative to Local Methods in Federated Learning

Elnur Gasanov[1] Ahmed Khaled[2] Samuel Horváth[1] Peter Richtárik[1]

[1]King Abdullah University of Science and Technology [2] Cairo University

King Abdullah University of Science and Technology

A New Formulation of Federated Learning

1. Cross-silo Federated Learning (FL) is typically formulated as the optimization problem

$$\min_{x\in\mathbb{R}^d}\left[f(x)=\frac{1}{n}\sum_{i=1}^n f_i(x)\right], \qquad \text{(ERM)}$$

where $i=1,2,\ldots,n$ are the clients/silos, and f_i is the loss defined by the data owned by client i.

2. We propose a new formulation of FL, which we call FedMix:

$$x^*=\arg\min_{x\in\mathbb{R}^d}\left[\tilde{f}(x)=\frac{1}{n}\sum_{i=1}^n f_i(\alpha_i x+(1-\alpha_i)x_i)\right]. \qquad \text{(FedMix)}$$

- $x_i=\text{argmin}\, f_i(x)$ is the pure model trained entirely on data owned by device i.
- $\alpha_1,\ldots,\alpha_n\in[0,1]$ are explicit personalization parameters.
- At node i we deploy the personalized model

$$T_i(x)=\alpha_i x^*+(1-\alpha_i)x_i.$$

3. Key properties of FedMix:
- Efficiently solvable as a finite-sum problem
- Adaptive to communication constraints
- Adaptive to personalization

Motivation 1: From Local GD to FedMix

1. Local GD (LGD) is a simple form of Federated Averaging for solving (ERM), used in the hope that its local steps address the communication bottleneck.
- Local GD:

$$x_{t+1}^i=\begin{cases}x_t^i-\gamma\nabla f_i(x_t^i) & \text{if } t \mod H\neq 0\\ \frac{1}{n}\sum_{i=1}^n\left[x_t^i-\gamma\nabla f_i(x_t^i)\right] & \text{if } t \mod H=0\end{cases},$$

where $H\geq 1$ is the number of local steps.
- However, LGD does not improve on GD in terms of communication complexity when solving ERM [2, 1] with heterogeneous data!

2. In light of this, Hanzely and Richtárik [2] recently proposed the implicitly personalized FL formulation

$$\min_{x\in\mathbb{R}^d}\left[f_\lambda(y_1,y_2,\ldots,y_n)=\frac{1}{n}\sum_{i=1}^n f_i(y_i)+\frac{\lambda}{2n}\|y_i-\overline{y}\|^2\right], \quad \text{(IPFL)}$$

where $\overline{y}=\frac{1}{n}\sum_{i=1}^n y_i$ and $\lambda>0$ is a regularization parameter.
- As the λ varies between 0 and ∞, the solutions of (IPFL) interpolate between the pure local optimal models (i.e., $x_i=\text{argmin}\, f_i(x)$) and the minimizer of (ERM). The solutions $y_1,\ldots,y_n$ found by Local GD are an *implicit mixture* of the local minimizers $x_1,\ldots,x_n$ and the solution of (ERM) $\text{argmin}\, f(x)$.
- Key observation: When applying SGD to (IPFL), seen as a 2-sum problem, one recovers a (variant) of LGD.
- Key result of [2]: However, when LGD is seen as a method for solving (IPFL) as opposed to (ERM), its communication complexity improves even in the heterogeneous case, and diminishes to 0 as $\lambda\to 0$. So, with increased personalization (= smaller λ), they get better communication complexity, resolving an important issue in FL.

3. Motivation 1: Can we design a different formulation of FL, one in which we would have the same benefits, but where local steps would not be needed? FedMix is the answer!

Motivation 2: from MAML to FedMix

1. Model Agnostic Meta Learning (MAML) objective is

$$\min_{x\in\mathbb{R}^d}\frac{1}{n}\sum_{i=1}^n f_i(x-\gamma\nabla f_i(x)). \qquad (1)$$

2. Key observation: Suppose that we run GD for $H\geq 1$ steps on the quadratic objective $f_i(x)=\frac{1}{2}x^T\boldsymbol{A}_i x-b_i^T x+c$, starting from some $x^0\in\mathbb{R}^d$. If the stepsize satisfies $\gamma\leq\frac{1}{L_i}$, where $L_i=\lambda_{\max}(\boldsymbol{A}_i)$, then the final iterate x_i^H can be written as

$$x_i^H=\left(\boldsymbol{I}-\boldsymbol{J}_i^H\right)x_i+\boldsymbol{J}_i^H x^0,$$

where $x_i=\arg\min f_i$ and $\boldsymbol{J}_i\in\mathbb{R}^{d\times d}$ is a matrix with maximum eigenvalue smaller than 1, i.e. $\lambda_{\max}(\boldsymbol{J})<1$.
- Plugging this result into Equation (1), observe that in MAML we find the initial model x^0 by solving the problem

$$\min_{x\in\mathbb{R}^d}\frac{1}{n}\sum_{i=1}^n f_i((\boldsymbol{I}-\boldsymbol{J}_i)x_i+\boldsymbol{J}_i x).$$

- Hence, MAML is optimizing or a specific weighted average of the initial model x^0 and the local solutions $x_1,x_2,\ldots,x_n$.

3. Motivation 2: FedMix can be can be seen as dispensing with the specific matrix $\boldsymbol{J}_i$, and instead optimizing the average weighted with an arbitrary constant $\alpha_i\in[0,1]$.

Theoretical Properties of FedMix

- FedMix preserves smoothness and convexity. In particular, suppose that each objective f_i is L_i-smooth, i.e.,

$$\|\nabla f_i(x)-\nabla f_i(y)\| \;\leq\; L_i\|x-y\|$$

for all $x,y\in\mathbb{R}^d$. Then for the FedMix objective $\tilde{f}$, we have
(i) $\tilde{f}$ is L_α-smooth with $L_\alpha\overset{\text{def}}{=}\frac{1}{n}\sum_{i=1}^n\alpha_i^2L_i$
(ii) If each f_i is convex, then $\tilde{f}$ is also convex
(iii) If each f_i is μ_i-strongly convex, then $\tilde{f}$ is μ_α-strongly convex with $\mu_\alpha\overset{\text{def}}{=}\frac{1}{n}\sum_{i=1}^n\alpha_i^2\mu_i$

- FedMix regularizes model variance. Let $T_1(x),T_2(x),\ldots,T_n(x)$ be the deployed personalized models. For $y_1,\ldots,y_n\in\mathbb{R}^d$ define

$$V(y_1,\ldots,y_n)\overset{\text{def}}{=}\frac{1}{n}\sum_{i=1}^n\|y_i-\bar{y}\|^2,$$

where $\bar{y}=\frac{1}{n}\sum_i y_i$. Then,

$$V(T_1(x),T_2(x),\ldots,T_n(x))=(1-\beta)^2V(x_1,x_2,\ldots,x_n),$$

if we assume that $\beta:=\alpha_1=\alpha_2=\ldots=\alpha_n$. So, the variance of the deployed models is smaller if β is smaller.

- FedMix enjoys a one-shot learning property. Suppose that each objective f_i is L_i-smooth, and let $\hat{L}\overset{\text{def}}{=}\frac{1}{n}\sum_{i=1}^n L_i$. Given the pure local models $x_1,x_2,\ldots,x_n$, define the weighted average

$$x^{\text{avg}}\overset{\text{def}}{=}\sum_{i=1}^n w_i x_i, \qquad w_i\overset{\text{def}}{=}\frac{\alpha_i^2L_i}{nL_\alpha}, \qquad L_\alpha\overset{\text{def}}{=}\frac{1}{n}\sum_{i=1}^n\alpha_i^2L_i. \qquad (2)$$

We further define the constants

$$D\overset{\text{def}}{=}\max_{i,j=1,\ldots,n,i\neq j}\|x_i-x_j\|, \quad \text{and} \quad V\overset{\text{def}}{=}\sum_{i=1}^n w_i\|x_i-x^{\text{avg}}\|^2. \qquad (3)$$

Fix any $\epsilon>0$. Assume that either $\max_{i=1,\ldots,n}\alpha_i\leq\sqrt{2\epsilon}/\sqrt{\hat{L}D}$, or $\alpha_i=\beta$ for all i and $\beta\leq\sqrt{2\epsilon}/\sqrt{\hat{L}D}$. Then x^{avg} is an ϵ-approximate minimizer of (FedMix).

Distributed GD Applied to FedMix

Distributed GD applied to FedMix:

$$x^{k+1}=x^k-\frac{\gamma}{n}\sum_{i=1}^n\alpha_i\nabla f_i\left(\alpha_i x^k+(1-\alpha_i)\,x_i\right). \qquad \text{(DGD)}$$

Suppose that each f_i in (FedMix) is L_i-smooth and μ_i-strongly convex. Define x^{avg}, L_α, and $\hat{L}$ by (2) and V,D by (3). Suppose that we run DGD for K iterations starting from $x^0=x^{\text{avg}}$. Then the following hold:

i) If the α_i are allowed to be arbitrary, then for $\alpha_{\max}\overset{\text{def}}{=}\max_{i=1,\ldots,n}\alpha_i$ we have

$$\tilde{f}(x^k)-\min_{x\in\mathbb{R}^d}\tilde{f}(x)\leq\left(1-\frac{\mu_\alpha}{L_\alpha}\right)^K\frac{\alpha_{\max}^2\hat{L}D}{2}.$$

ii) If $\alpha_i=\beta$ for all i, then

$$\tilde{f}(x^k)-\min_{x\in\mathbb{R}^d}\tilde{f}(x)\leq\left(1-\frac{\hat{\mu}}{\hat{L}}\right)^K\frac{\beta^2\hat{L}V}{2}, \qquad (4)$$

where $\hat{\mu}\overset{\text{def}}{=}\frac{1}{n}\sum_{i=1}^n\mu_i$.

Performance of Compressed Gradient Methods on FedMix

Theoretical performance of Distributed Compressed Gradient Descent (DCGD) and the DIANA method of Mishchenko et al [4] in different settings on FedMix. Note that in all cases, complexity improves as α decreases.

Algorithm	Assumption	Convergence guarantee
DCGD	smoothness, scvx	$\mathbb{E}\|x^k-x^*\|^2\leq(1-\gamma\alpha\overline{\mu})^kC_1+C_2$ $\mathbb{E}[\tilde{f}(x^k)-\tilde{f}^*]\leq((1-\gamma\alpha\overline{\mu})^kC_1+C_2)\alpha^2$
DIANA	smoothness, scvx	$\mathbb{E}\|x^k-x^*\|^2\leq(1-\rho)^kC$ $\mathbb{E}[\tilde{f}(x^k)-\tilde{f}^*]\leq(1-\rho)^kC\alpha^2$
DCGD	smoothness, cvx	$\mathbb{E}[f(\overline{x}^k)-f(x^*)]\leq\frac{1}{k}C_1\alpha^2+C_2\alpha$
DIANA	smoothness, cvx	$\mathbb{E}[f(\overline{x}^k)-f(x^*)]\leq\frac{1}{k}(C_1\alpha^2+C_2\alpha)$
DCGD	smoothness	$\min_{0\leq t\leq k-1}\mathbb{E}\|\nabla\tilde{f}(x^t)\|^2\leq\frac{\left(1+C_1\gamma_0^2\right)^kC_2}{\gamma_0k}\alpha^2$
DIANA	smoothness	$\mathbb{E}\|\nabla\tilde{f}(\hat{x})\|^2\leq\frac{C}{k}\alpha^2$

Table: Abbreviations: cvx = convex, scvx = strongly convex. All the constants are independent of α.

Generalization of Sine Functions

Following [4], we consider MSE loss on functions

$$f_i(x)=a_i\sin(x+b_i).$$

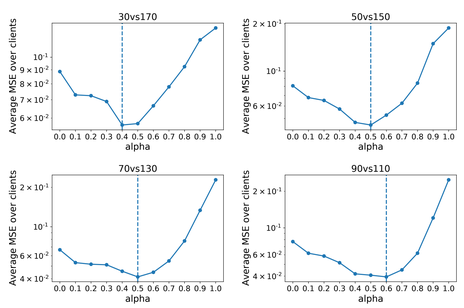
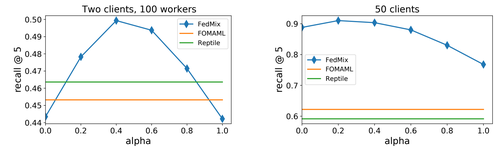


Figure: Average MSE over clients as a function of the personalization parameter α. In all cases, $0<\alpha<1$ is optimal; that is, it is optimal to use a mixture, and not to rely on either the maximally personalized models $x_i=\arg\min f_i$ (this corresponds to $\alpha=0$), or on the single global model $x^*=\arg\min f$ (this corresponds to $\alpha=1$).

Generalization on Real Clients' Data

To test the generalization of FedMix on real data we adopt Stackoverflow dataset [3], and compare its performance with two baselines, FOMAML and Reptile.



(a) 100 workers created out of two clients' data

(b) 50 workers with distinct data distributions

Figure: Test accuracy of our FedMix model for different personalization parameter values, versus FOMAML and Reptile. We choose $\alpha_i=\alpha$ for all i, (see horizontal axis). FOMAML and Reptile are independent from the personalization parameter α. Plots correspond to different data splittings.

Optimization Experiments

We consider FedMix with

$$f_i(x)\coloneqq\frac{1}{k_i}\sum_{j=1}^{k_i}\left[\log\left(1+\exp\left(-a_{i,j}^\top x\right)\right)\right]+\frac{\lambda}{2}\|x\|^2.$$

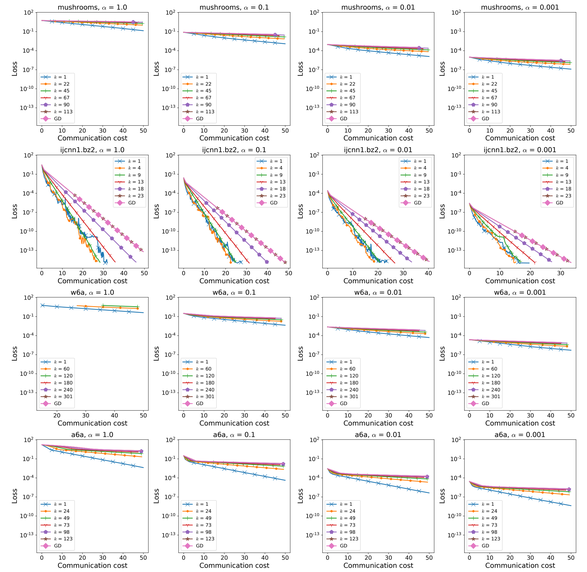


Figure: Loss $f(x)-f^*$ vs. communication cost in unit of thousands of float numbers of DIANA and GD for logistic regression problem, l_2 regularizer for four datasets, k is the sparsification parameter of Random-k compressor.

References

1. Reese Pathak and Martin J. Wainwright. FedSplit: An algorithmic framework for fast federated optimization. arXiv:2005.05238, 2020.
2. Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. arXiv:2002.05516, 2020.
3. TensorFlow Developers. TensorFlow Federated Stack Overflow dataset, 2021.
4. Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. ICML 2017.
5. Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč and Peter Richtárik. Distributed learning with compressed gradient differences. arXiv:1901.09269, 2019.