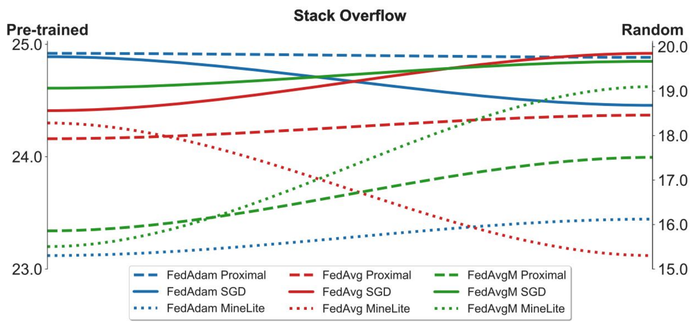
Where to Begin? On the Impact of Pre-Training and Initialization in Federated Learning

John Nguyen, Jianyu Wang, Kshitiz Malik,
Maziar Sanjabi, Michael Rabbat

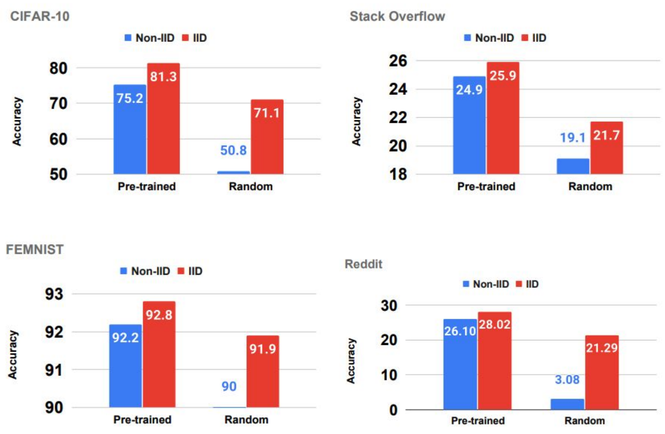
Pre-training changes the ranking of federated optimization algorithms.

If one sorts federated optimization methods based on their performance when starting from a random initialization, the order is substantially different from when using a pre-trained initialization.



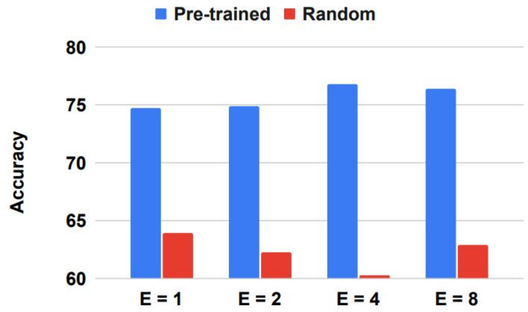
Pre-training closes the accuracy gap between non-IID and IID.

The gap between models trained on IID data and models trained on non-IID data is significantly smaller when starting with pre-trained weights.



The average accuracy on 3 different seeds for FEDADAM trained on IID and non-IID data. For CIFAR-10 Non-IID, we generate 100 non-IID clients using a Dirichlet(0.1). For other three datasets, we use the natural non-IID client partitions.

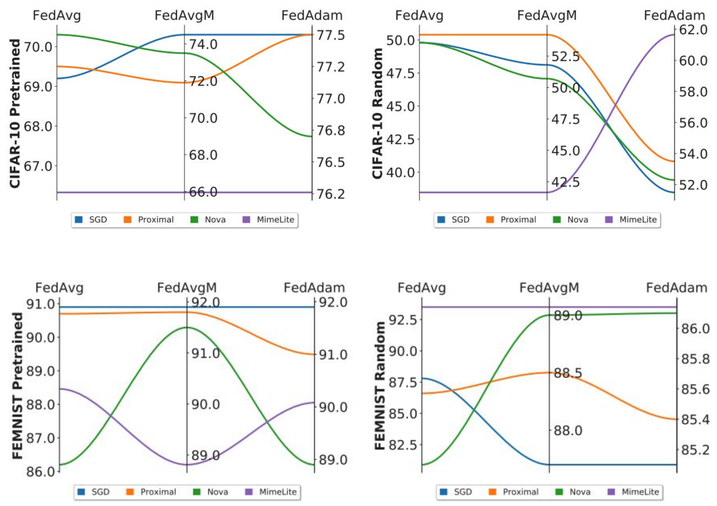
Pre-training reduces the negative effects of client drift.



We observe that when training from a pre-trained model, increasing the number of local updates does not degrade the final accuracy, in contrast to training from a random model.

Pre-training reduces the impact of system heterogeneity.

The accuracy gap between algorithms is more pronounced in the random initialization setting, whereas in the pre-trained setting, all algorithms converge to more similar accuracies.



Understanding why pre-training helps federated optimization.

	CIFAR-10	FEMNIST	Stack Overflow	Reddit
Pre-trained	661.99	26.29	151.05	647.19
Random	4843.13	355.51	185.02	1309.68

The top eigenvalue of the Hessian matrix for each dataset between the pre-trained and random initialized models

We examine the largest eigenvalue of the Hessian matrix (i.e., local Lipshitz constant) at the beginning of training, a larger value of which suggests a harder-to-optimizer loss surface.

Pre-training helps align client updates.

Training and gradient statistics of a Resnet18 on CIFAR-10 with Dirichlet distribution with parameter 0.1. Top row: Train loss of global model; train accuracy of global model; evaluation accuracy of global model; evaluation loss of global model. Bottom row: Gradient diversity of client updates; cosine similarity between client updates; L2 distance of server weights from their final values at the end of training.

Recommendations

1. When evaluating FL algorithms, researchers should experiment with both pre-trained and random weights.
2. Using adaptive server optimizers such as FEDADAM together with SGD at the client is a simple and competitive approach to start from a pre-trained model.
3. When focusing on heterogeneity, it may be worth considering whether or not proxy data is available for pre-training to motivate the application considered

Take a photo to learn more: