# FEDAVG 1

Federated Optimization introduces the following key properties that differentiate it from the distributed learning setting.

Non-IID data, Unbalanced data, Massively distributed data, and Limited Communication between devices.

In particular FedAvg intends to reduce the number of communication rounds. A client performs a series of training iterations on its local data PK before broadcasting their model to the global model. This is then averaged, with a weight penalty applied based on the proportion of global data a client maintains.

# FEDAVG 2

The key observation for the Fed Average algorithm is that performing a series of epochs on a client prior to updating the global model significantly decreases the number of communication rounds and consequently increases the rate of convergence.

This is evidenced by the results on the CIFAR-10 dataset.

# SCAFFOLD 1

The first alternative learning scheme intuitively abounds as a result of the heterogeneity of local data. We can view the heterogeneity as introducing ‘client-variance’. Where the client’s optimal representation tends to drift from the global optimum.

We can see this in the following image that demonstrates the true global optimum and local optima of each client.

# SCAFFOLD 2

SCAFFOLD introduces a control variate term for both the client and server models. In effect this term corrects the local update so that the local models converge to the global optimum.

This is illustrated in the following image. The update schemes to the local gradient and to the control variates are also shown in the accompanying equations.

# FEDPROX 1

Fed Prox is another algorithm that intends to limit the drift of the local client as a result of the heterogeneity of local data. This algorithm instead uses a proximal term to reduce the variance of local models from the global model.

# FEDPROX 2

The key observation of Fed Prox is that bounding the dissimilarity between local and global models in each update allows the authors to theoretically and empirically prove convergence.

# MOON 1

The idea behind MOON is inspired by contrastive learning in Computer Vision.

With MOON, the model intends to decrease the distance between the representation learned by the local model and the representation learned by the global model. While increasing the distance between the representation learned by the local model and the representation learned by the previous local model.

In order to do this MOON introduces a new loss term to the overall loss. The weight of this loss term is a hyperparameter.

# MOON 2

In the following two images we can see a high level overview of the MOON algorithm and how it is inspired by contrastive algorithms like SimCLR.

In essence we intend to maximize the agreement between the global and local models. We do this by maintaining the previous local model and the global model which we then use to calculate the model contrastive loss.

# FEDMA 1

The last alternative approach I will present is Federated matched averaging. Federated matched averaging takes a different perspective to limiting the drift between global and local models.

Instead of viewing the issue as a drift between models, matched averaging reformulates the problem as an erroneous update from the local to global models. As a result of the potential permutations on the weight matrices of the global and local models.

To undo the permutation the models learn a maximum bipartite matching between the clients and server. These are matched based on the similar feature extraction signatures of hidden elements.

# FEDMA 2

I have attached here an overview of the Federated Matched Averaging algorithm. However, we won’t go over it.

# FEDMA 3

The key observation and outcome of this paper is that the authors observe that training longer actually benefits Fed Matched Averaging. While FedAvg and FedProx tend to require a cutoff point while training.