On the connection between FedDyn and FedPD

Xinwei Zhang and Mingyi Hong

February 2021

Abstract

In this note, we consider two recent algorithms developed for Federated Learning – FedDyn and FedPD. Each algorithm is designed for settings in which users are heterogeneous, and each tries to reduce the communication burdens of the system (in different manners). Specifically, FedPD reduces communication by skipping some interaction between the server and users whenever possible, while FedDyn allows the users to perform partial participation. In this note, we provide a short discussion about the connections of these two algorithms – without communication reduction, these two algorithms are identical. In particular, the so-called "dynamic regularization" step in FedDyn is precisely the dual update step in FedPD.

1 Setting

Let us consider the following standard federated learning problem:

$$\arg\min_{\mathbf{x}} \left[f(\mathbf{x}) \triangleq \frac{1}{N} \sum_{i=1}^{N} f_i(\mathbf{x}) \right]$$

where N is the total number of users, and $f_i(x) : \mathbb{R}^m \to \mathbb{R}$ is a smooth and possibly non-convex local objective function. Other notations used in this note is listed in the table below.

Table 1: Summary of notation used in the paper

N, i	The total number, and the index, of clients
$\mathcal{N}$	The set of all clients
M, B, b	The total number, batch size and index of samples
T, t	The total number and index of communication rounds
Q, q	The total number and index of local updates
$\mathbf{x}_0, \mathbf{x}_i$	The global and local model parameters
λ_0, λ_i	The global and local auxiliary variables
$.^{t,q}$	The variable index at q^{th} local iteration of t^{th} global iteration

2 The FedPD and FedDyn Algorithms

The Federated Primal-Dual (FedPD) algorithm is proposed in [2]. The idea is to use a set of linear constraints to indicate that the global variable is shared among the local clients. Each iteration t,

the local client updates its local variable $\mathbf{x}_i^{t+1}$ based on (inexactly) optimizing a local Lagrangian function, parameterized by the previous model $\mathbf{x}_{0,i}^t$, followed by a dual variable update. Then the local nodes with either (with probability $p-1$) send $\mathbf{x}_i^{t+1}+\eta\lambda_i^{t+1}$ to the server, or will continue update. In the former update, the server will perform an averaging and broadcast the results to the users as a set of new $\mathbf{x}_{0,i}^{t+1}$'s; in the latter case, the server will do nothing, and the clients will continue their local updates. Please see the table below.

Algorithm 1 Federated Primal-Dual Algorithm

Input: $\mathbf{x}^0, \lambda^0, \eta, p, T$
Initialize: $\mathbf{x}_0^0 = \mathbf{x}^0$,
for $t = 0, \ldots, T-1$ **do**
  for $i = 1, \ldots, N$ in parallel **do**

$$\mathbf{x}_i^{t+1} = (\text{inexact}) \arg\min_{\mathbf{x}} f_i(\mathbf{x}) + \left\langle \lambda_i^t, \mathbf{x} - \mathbf{x}_{0,i}^t \right\rangle + \frac{1}{2\eta} \left\| \mathbf{x} - \mathbf{x}_{0,i}^t \right\|^2 \tag{1}$$

$$\lambda_i^{t+1} = \lambda_i^t + \frac{1}{\eta}(\mathbf{x}_i^{t+1} - \mathbf{x}_{0,i}^t) \tag{2}$$

  end for
  With probability $1-p$, do global communication:

$$\mathbf{x}_0^{t+1} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i^{t+1} + \eta\lambda_i^{t+1}) \tag{3}$$

$$\mathbf{x}_{0,i}^{t+1} = \mathbf{x}_0^{t+1},\ i = 1, \ldots, N \tag{4}$$

  With probability p, skip global communication:
    Local Update: $\mathbf{x}_{0,i}^{t+1} \triangleq \mathbf{x}_i^{t+1} + \eta\lambda_i^{t+1}$
end for

The Federated Dynamic Regularization Algorithm (FedDyn) proposes "a dynamic regularizer for each device at each round, so that in the limit the global and device solutions are aligned" [1]. Further, in each iteration t, only a subset of users $\mathcal{P}_t \subseteq \mathcal{U}$ is selected, out of a total of N users. The FedDyn algorithm is given below.

Algorithm 2 Federated Dynamic Regularizer

Input: $\mathbf{x}^0, \eta, T$,
Initialize: $\mathbf{x}_0^0 = \mathbf{x}^0$, $\mathbf{h}^0$
for $t = 0, \ldots, T-1$ **do**
  for $i \in \mathcal{P}_t$ in parallel do local updates **do**

$$\mathbf{x}_i^{t+1} = \arg\min_{\mathbf{x}} f_i(\mathbf{x}) + \left\langle \nabla f_i(\mathbf{x}_i^t), \mathbf{x} - \mathbf{x}_0^t \right\rangle + \frac{1}{2\eta} \left\| \mathbf{x} - \mathbf{x}_0^t \right\|^2 \tag{5}$$

$$\nabla f_i(\mathbf{x}_i^{t+1}) = \nabla f_i(\mathbf{x}_i^t) + \frac{1}{\eta}(\mathbf{x}_i^{t+1} - \mathbf{x}_0^t) \tag{6}$$

  end for
  for $i \notin \mathcal{P}_t$ in parallel do local updates **do**

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t, \quad \nabla f_i(\mathbf{x}_i^{t+1}) = \nabla f_i(\mathbf{x}_i^t) \tag{7}$$

  end for
    Global Communicate:

$$\mathbf{h}^{t+1} = \mathbf{h}^t + \frac{1}{\eta N} \sum_{i \in \mathcal{P}_t} (\mathbf{x}_i^{t+1} - \mathbf{x}_0^t) \tag{8}$$

$$\mathbf{x}_0^{t+1} = \frac{1}{|\mathcal{P}_t|} \sum_{i \in \mathcal{P}_t} \mathbf{x}_i^{t+1} + \eta \mathbf{h}^{t+1} \tag{9}$$

end for

3 Comparisons

To see the relation between these two algorithms, let us assume the following:

- Let $\mathcal{P}_t = \mathcal{N}$ for FedDyn, that is, all clients will participate in communication in all the iterations;
- Consider $p = 0$ for FedPD, that is, communication will take place in all the iterations;
- Consider a simplified version of FedPD where the local problem (1) is solved *exactly*.
- FedPD and FedDyn are initialized such that their initial $\mathbf{x}^0$ are the same, and that the following holds:

$$\lambda_i^0 = \lambda_j^0 = \mathbf{h}^0 = \nabla f_i(\mathbf{x}^0), \ \forall \ i, j. \tag{10}$$

Further, we will show that the $\{\mathbf{x}_i^t\}$ and $\{\mathbf{x}_0^t\}$ iterates generated by the two algorithms are the same.

Below, we will show that the following two relations hold:

$$\nabla f_i(\mathbf{x}_i^t) = \lambda_i^t, \quad \mathbf{h}^t = \frac{1}{N} \sum_{i=1}^{N} \lambda_i^r \quad t = 0, 1, \cdots, T. \tag{11}$$

First, at $t = 0$, the two relations holds trivially because of the initialization.

Let us consider the update in $t = 0$. Clearly, the $\mathbf{x}_i^1$ updates in (5) and (1) are exactly the same, since they are both minimizing the local augmented Lagrangian function, and that $\nabla f_i(\mathbf{x}_i^0) = \lambda_i^0$, $\forall\ i$. Therefore, the two algorithms generate the same $\mathbf{x}_i^1$. It follows that for the FedDyn, the following hold:

$$\nabla f_i(\mathbf{x}_i^1) = \nabla f_i(\mathbf{x}_i^0) + \frac{1}{\eta}(\mathbf{x}_i^1 - \mathbf{x}_0^0) \overset{(i)}{=} \lambda_i^0 + \frac{1}{\eta}(\mathbf{x}_i^1 - \mathbf{x}_0^0) \overset{(ii)}{=} \lambda_i^1,\ \forall\ i, \tag{12}$$

where in (i) we used the initialization (10), and in (ii) we used (2), and the fact that the $\mathbf{x}_i^1$ generated by the two algorithms are exactly the same.

Next, we note that the following relations hold for FedDyn:

$$\begin{aligned}\mathbf{h}^{t+1} &= \mathbf{h}^t + \frac{1}{\eta N}\sum_{i=1}^N (\mathbf{x}_i^{t+1} - \mathbf{x}_0^t) \\ &\overset{(6)}{=} \mathbf{h}^t + \frac{1}{N}\sum_{i=1}^N \Big(\nabla f_i(\mathbf{x}_i^{t+1}) - \nabla f_i(\mathbf{x}_i^t)\Big).\end{aligned} \tag{13}$$

And in particular

$$\mathbf{h}^1 = \mathbf{h}^0 + \frac{1}{N}\sum_{i=1}^N \Big(\nabla f_i(\mathbf{x}_i^1) - \nabla f_i(\mathbf{x}_i^0)\Big) = \frac{1}{N}\sum_{i=1}^N \nabla f_i(\mathbf{x}_i^1) = \frac{1}{N}\sum_{i=1}^N \lambda_i^1,\ \forall\ i, \tag{14}$$

where the last equality comes from (12).

Utilizing the fact that $\mathbf{h}^1 = \frac{1}{N}\sum_{i=1}^N \lambda_i^1$, and the two algorithms have the same $\mathbf{x}_i^1$, by a direct comparison of (9) and (3), we obtain that $\mathbf{x}_0^1$ generated by the two algorithms are the same.

The case for all $t \geq 1$ can be similarly derived.

In conclusion, under the initialization (10), and assuming that $p = 1$ for FedPD and $\mathcal{P}_t = \mathcal{N}$ for all t, and the FedPD solves local problem exactly, then the FedPD and FedDyn are identical. The key observation from the above analysis is that, the so-called "dynamic regularization" updates in FedDyn are the dual variable updates in FedPD.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021.

[2] Xinwei Zhang, Mingyi Hong, Sairaj Dhople, Wotao Yin, and Yang Liu. Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data. *arXiv preprint arXiv:2005.11418*, 2020.