EE6367: Topics in Data Storage and Communications **2023**

Lecture 17: 18 October 2023

Instructor: Shashank Vatedka *Scribe: Gautam Singh*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

17.1 The FL Process

Let $M_t^{(g)}$ denote the *global model* at round t, in the server. The steps in each round of federated learning (FL) are as follows.

1. **Client Selection:** Clients are sampled uniformly at random.
2. **Broadcast:** Send $M_t^{(g)}$ to the selected users. We usually assume that this step is noiseless.
3. **Client Computation:** Each user i runs optimization algorithm and gets the *local model* $M_t^{(i)}$, which is transmitted to the server. Here, we assume that each client runs SGD on n_i non-iid samples.
4. **Aggregation:** Server gets $M_t^{(i)}$ for all users i that have not dropped out. We assume that this aggregation is noiseless. Note that user i can send its local model or the difference between its local model and the global model. Usually, the difference is sparse and has a small norm.
5. **Model Update:** Server updates the global model
$$M_{t+1}^{(g)} \leftarrow f\left(\left\{M_t^{(i)} : i\right\}\right). \tag{17.1}$$
Usually, a weighted average of the local models is taken.

17.2 Federated Averaging

At user i:

1. Receive $M_t^{(g)}$.
2. For $j \in \{1, 2, \ldots, N_e\}$: (here, N_e is the *number of epochs*)
   (a) Run minibatch SGD for N_m batch size
$$M_t^{(i)}(j) = M_t^{(i)}(j-1) - \eta \frac{1}{N_m} \sum_{l=1}^{N_m} g\left(x_l, M_t^{(i)}(j-1)\right) \tag{17.2}$$
3. Send $M_t^{(i)}(N_m)$ to the server.

At server:

1. Let $\left\{M_t^{(i)}\right\}_{i=1}^{N_u}$ be the number of local models obtained by the server.
2. Update the global model
$$M_{t+1}^{(g)} \leftarrow \frac{1}{n}\sum_{i=1}^{N_u} n_i M_t^{(i)} \tag{17.3}$$
where n_i is the number of samples with user i and $n = \sum_{i=1}^{N_u} n_i$ is the total number of samples.
3. For the next round, the server samples a random subset of N_u users, and transmits $M_{t+1}^{(g)}$.

17.3 Communication-Efficient Federated Learning

To make FL communication-efficient, the following methods are adopted.

1. **Structured updates:** Here, we assume that the input space is structured before the optimization algorithm is run.
   (a) *Low rank:* $M_t^{(i)} = UV$ where $U \in \mathbb{R}^{m\times k}$ and $V \in \mathbb{R}^{k\times n}$, $k << \min\{m, n\}$.
   (b) *Random mask:* $M_t^{(i)}$ is sparse. The server sends a random mask to drive the coefficients in these positions to zero.
2. **Sketched updates:** Here, no structure is added by the server, but a lossy version of the model is send to the server. $\tilde{M}_t^{(i)}$ is defined by running SGD. However, an unbiased estimator $M_t^{(i)}$ is sent.
   (a) *Subsampling:* Some entries are sampled by the client and transmitted to the server.
   (b) *Quantization:* Model coefficients are quantized before transmission to server.

The following conclusions can be made.

1. In general, sketched updates perform better than structured updates for smaller number of rounds, but worse that for larger number of rounds.
2. We can even apply a hybrid of the above methods at the client.
3. There is a marked increase in accuracy when using two bits per dimension as opposed to one, but not after that.