1. ***The information about the paper (published in where, when, authors…)***

Title:

Federated Learning: Strategies for Improving Communication Efficiency

Authors:

Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, Dave Bacon

Publish time:

30 Oct 2017

2. ***What problem they want to solve?***

A round of Federated Learning is composed of several steps:

a. A subset of existing clients is selected, each of which downloads the current model.
b. Each client in the subset computes an updated model based on their local data.
c. The model updates are sent from the selected clients to the sever.
d. The server aggregates these models (typically by averaging) to construct an improved global model.

Original implementation of model update requires each client to send a full model back to the server, uploading models become a bottleneck of Federated Learning. Thus, authors of this paper introduce some methods to reduce the upload time of clients while keeping original model quality.

3. ***Proposed methods***

$$\mathbf{H}_t^i := \mathbf{W}_t^i - \mathbf{W}_t,$$

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \eta_t \mathbf{H}_t, \qquad \mathbf{H}_t := \frac{1}{n_t} \sum_{i \in S_t} \mathbf{H}_t^i.$$

Authors use this equation to represent model update.

W is matrix of model parameters, and dimension is **d1 x d2**, **d1** and **d2** represents the output and input dimensions respectively.

H represents the difference between local model and server model of each clients, then average each client's difference to upload to the server.

Then authors propose two kinds of method to reduce time of uploading model.

a. Structured update

First type of efficient communication update restricts the updates **H** to have a pre-specified structure, two types of structures are introduced in this paper: **low rank** and **random mask.**

Low rank.

$$\mathbf{H}_t^i = \mathbf{A}_t^i \mathbf{B}_t^i, \text{ where } \mathbf{A}_t^i \in \mathbb{R}^{d_1 \times k}, \mathbf{B}_t^i \in \mathbb{R}^{k \times d_2}.$$

Authors express **H** as product of **A** and **B**, and dimension of **A** and **B** are constrained by k, which is a fixed number. In every round, each client will refresh matrix **A** by random seed independently and optimize matrix **B** then send matrix **B** to the server.

Random mask.

In this structure, authors restrict the update **H** to be a sparse matrix. In every round, each client will have a new mask by random seed independently and use this mask to wipe out some entries of original matrix **H**. Finally, each client only sends non-zero entries and random seed of mask to the server.

b. Sketched update

In sketched update, each client will do their local training without any constraint then update matrix **H** with some lossy form.

Subsampling.

Instead of sending **H**, each client only communicates matrix **H** which is formed from a random subset of the values of **H**. The server then averages the subsampled updates, producing the global update **H'**.

Probabilistic quantization.

In Probabilistic quantization, first vectorize original matrix **H** into vector **h**, then take the maximum and minimum of vector **h**, then quantize each value to 1-bit with below equation. If we want to quantize value to b-bits, we should divide interval of maximum and minimum to b intervals, if value falls in one interval, then use upper value of this interval to represent **h'max** and use lower values of this interval to represent **h'min** to use equation to quantize value.

$$\tilde{h}_j = \begin{cases} h_{\max}, & \text{with probability } \frac{h_j - h_{\min}}{h_{\max} - h_{\min}} \\ h_{\min}, & \text{with probability } \frac{h_{\max} - h_j}{h_{\max} - h_{\min}} \end{cases}.$$

Improving the quantization by structured random rotations

When max = 1 and min = −1 and most of values are 0, the 1-bit quantization will lead to a large error. We note that applying a random rotation on **h** before the quantization (multiplying **h** by a random orthogonal matrix) solves this issue. This claim has been theoretically supported in Suresh et al. (2017).

4. *Experimental results*

They use their methods on two different tasks to test their performance, they first experiment with **CIFAR-10 image classification** task using **CNN**. Next, they experiment with **next word prediction** tasks based on **Reddit** post data by **RNN**,

which is more realistic scenario.

CIFAR-10 image classification (CNN)

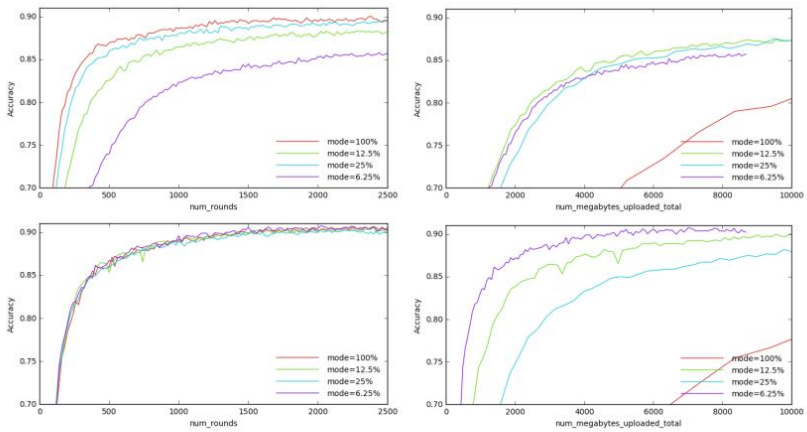
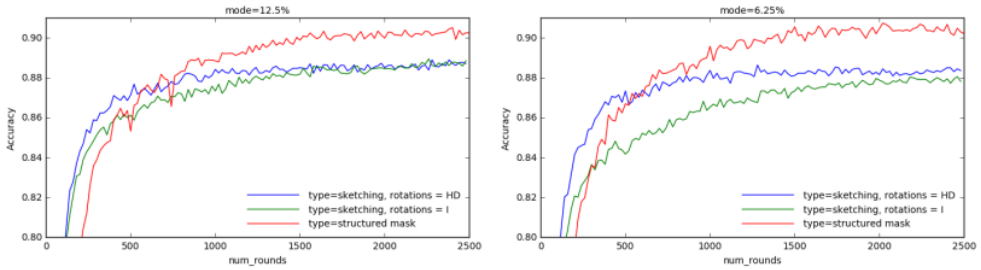


Figure 1: Structured updates with the CIFAR data for size reduction various modes. *Low rank* updates in top row, *random mask* updates in bottom row.

In this picture, they compare different result of **structure updates**, mode = 25% represents that the fixed number k is equal to 1 / 4 of original rank in **low rank** and represents that only 25% of data will remain in **random mask**.

From this result, we can see that **random mask** outperform **low rank**, **random mask** uses less round and less data sent to the server to train a good model than **low rank**.



In this picture, they compare different result of **structured updates** and **sketched updates**, rotation = I means no rotation and rotation = HD means model is sketched with rotation.

In this experiment, we can see that **structured updates** perform better than **sketched updates**. This is because **sketched updates** will loss some information when uploading models, thus reducing the performance of model.

Next word prediction using Reddit post data (RNN)

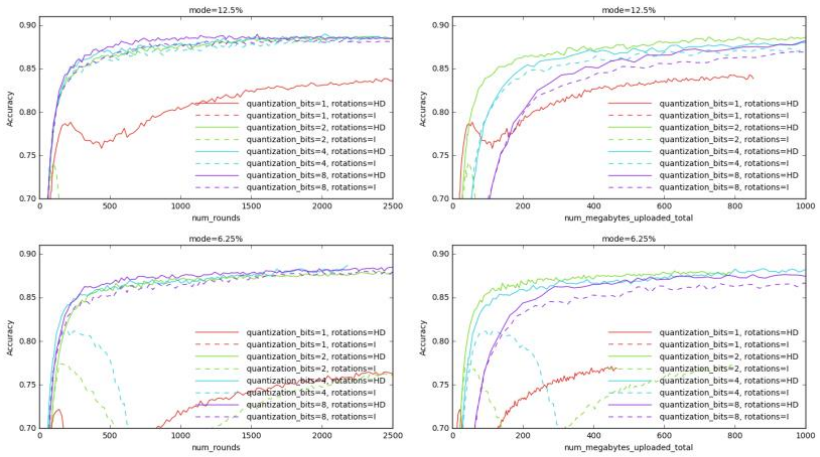


Figure 3: Comparison of sketched updates, combining preprocessing the updates with rotations, quantization and subsampling on the CIFAR data.

In this experiment, they compare different result of using different quantization bits and with or without rotation of **sketched updates**.

So, we can find that **quantized sketched update** with rotation will perform better than without rotation and if we use only 1-bit to quantize values, we might get terrible performance.

5. *Weakness and strength of this paper*

Strength:

This paper is easy to understand, partitioned very clearly and use several experiment results to validate their methods.

Weakness:

They introduce 2 methods for **structured updates** and 2 methods for **sketched updates**, but experiment result doesn't have **subsampling of sketch updates**, so reader can't know performance of **subsampling**.

6. *Possible extensions*

Suggestion engine:

Because search history is private data, if we don't use **Federated Learning** to train this model, we will encounter shared data problem, if we use **Federated Learning** with this paper' methods, we can train a good model without influencing client's experiences.