Federated Learning - Summary

Willy Fitra Hendria
willyfitrahendria@gmail.com

Introduction

- Introduced in a paper entitled “Communication-Efficient Learning of Deep Networks from Decentralized Data”, H. Brendan McMahan, et al
- The client’s data doesn’t need to be shared in order to learn a global model. Instead, the global model is learned by aggregating the locally-computed updates from each client devices.
- Addresses the concerns of privacy and communication costs .

Ideal Problems for Federated Learning

Have the following properties:

1) Training on real-world data provides a distinct advantage
2) The data is privacy sensitive or large in size
3) Labels on the data can be inferred naturally from user interaction

Federated Optimization

Several key properties that differ from typical distributed optimization problem:

- Non-IID
  - Any particular client's dataset will not be representative of the population distribution
- Unbalanced
  - Varying amounts of local training data
- Massively distributed
  - The number of clients to be much larger than the average number of examples per client
- Limited communication
  - Clients are frequently offline or on slow or expensive connections

Algorithm (FederatedAveraging)

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

```
Server executes:
  initialize w_0
  for each round t = 1, 2, ... do
    m ← max(C · K, 1)
    S_t ← (random set of m clients)
    for each client k ∈ S_t in parallel do
      w^k_{t+1} ← ClientUpdate(k, w_t)
    w_{t+1} ← Σ_{k=1}^{K} (n_k / n) w^k_{t+1}

ClientUpdate(k, w):  // Run on client k
  𝓑 ← (split 𝒫_k into batches of size B)
  for each local epoch i from 1 to E do
    for batch b ∈ 𝓑 do
      w ← w − η∇ℓ(w; b)
  return w to server
```

Source: McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics, 1273–1282.

- The amount of computation is controlled by three key parameters: C (Client Fraction), E (Local Epoch), and B (Local Minibatch).
- Models are locally-trained on each 1client:

$$w_{t+1}^{k} \leftarrow \mathrm{ClientUpdate}(k, w_t)$$

- Central server averages the resulting models
-

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^{k}$$

-
- With E=1 and B=∞, it is corresponds to FedSGD

Experimental Results

2NN	IID		Non-IID	
C	$B = \infty$	$B = 10$	$B = \infty$	$B = 10$
0.0	1455	316	4278	3275
0.1	1474 (1.0×)	87 (3.6×)	1796 (2.4×)	664 (4.9×)
0.2	1658 (0.9×)	77 (4.1×)	1528 (2.8×)	619 (5.3×)
0.5	— (—)	75 (4.2×)	— (—)	443 (7.4×)
1.0	— (—)	70 (4.5×)	— (—)	380 (8.6×)
CNN, $E = 5$				
0.0	387	50	1181	956
0.1	339 (1.1×)	18 (2.8×)	1100 (1.1×)	206 (4.6×)
0.2	337 (1.1×)	18 (2.8×)	978 (1.2×)	200 (4.8×)
0.5	164 (2.4×)	18 (2.8×)	1067 (1.1×)	261 (3.7×)
1.0	246 (1.6×)	16 (3.1×)	— (—)	97 (9.9×)

With B=∞, there is only a small advantage in increasing C. Using smaller B=10 shows a significant improvement in using C >= 0.1, especially in the non-IID case

Source: McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics, 1273–1282.

Experimental Results

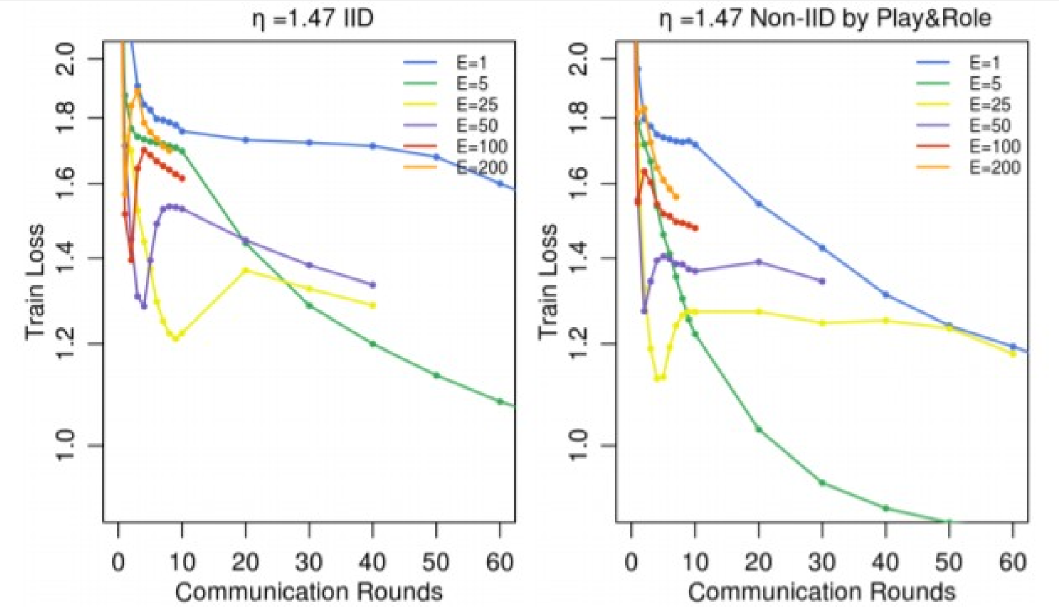
MNIST CNN, 99% ACCURACY					
CNN	E	B	u	IID	NON-IID
FEDSGD	1	∞	1	626	483
FEDAVG	5	∞	5	179 (3.5×)	1000 (0.5×)
FEDAVG	1	50	12	65 (9.6×)	600 (0.8×)
FEDAVG	20	∞	20	234 (2.7×)	672 (0.7×)
FEDAVG	1	10	60	34 (18.4×)	350 (1.4×)
FEDAVG	5	50	60	29 (21.6×)	334 (1.4×)
FEDAVG	20	50	240	32 (19.6×)	426 (1.1×)
FEDAVG	5	10	300	20 (31.3×)	229 (2.1×)
FEDAVG	20	10	1200	18 (34.8×)	173 (2.8×)
SHAKESPEARE LSTM, 54% ACCURACY					
LSTM	E	B	u	IID	NON-IID
FEDSGD	1	∞	1.0	2488	3906
FEDAVG	1	50	1.5	1635 (1.5×)	549 (7.1×)
FEDAVG	5	∞	5.0	613 (4.1×)	597 (6.5×)
FEDAVG	1	10	7.4	460 (5.4×)	164 (23.8×)
FEDAVG	5	50	7.4	401 (6.2×)	152 (25.7×)
FEDAVG	5	10	37.1	192 (13.0×)	41 (95.3×)

MNIST 2NN	E	B	u	IID	NON-IID
FEDSGD	1	∞	1	1468	1817
FEDAVG	10	∞	10	156 (9.4×)	1100 (1.7×)
FEDAVG	1	50	12	144 (10.2×)	1183 (1.5×)
FEDAVG	20	∞	20	92 (16.0×)	957 (1.9×)
FEDAVG	1	10	60	92 (16.0×)	831 (2.2×)
FEDAVG	10	50	120	45 (32.6×)	881 (2.1×)
FEDAVG	20	50	240	39 (37.6×)	835 (2.2×)
FEDAVG	10	10	600	34 (43.2×)	497 (3.7×)
FEDAVG	20	10	1200	32 (45.9×)	738 (2.5×)

With C=0.1, adding more local updates per round (increase E & decrease B) can produce a dramatic decrease in communication costs

Source: McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics, 1273–1282.

Experimental Results

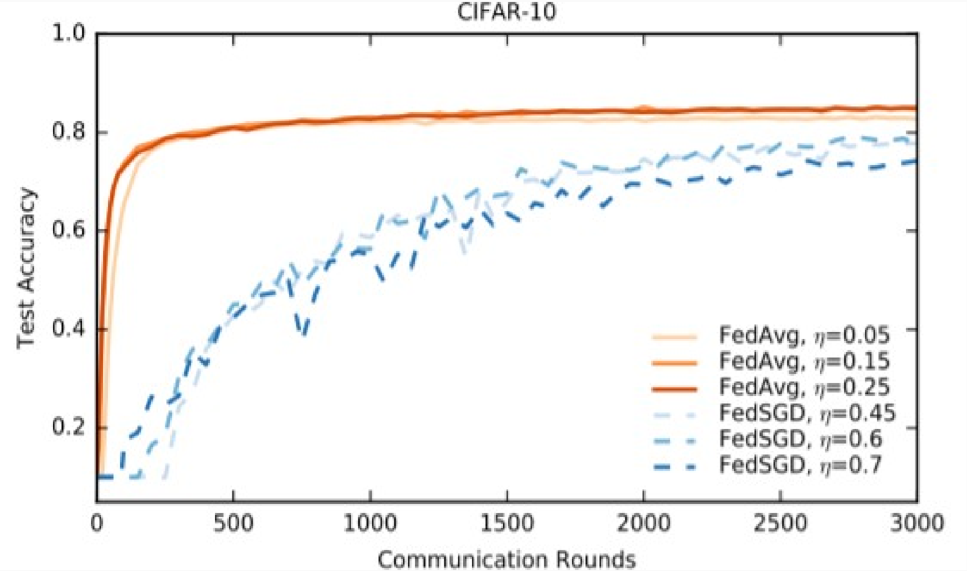


For very large E, FedAvg can plateau or diverge.

It may be useful to decay the amount of local computation per round (moving to smaller E or larger B)

Source: McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics, 1273–1282.

Experimental Results

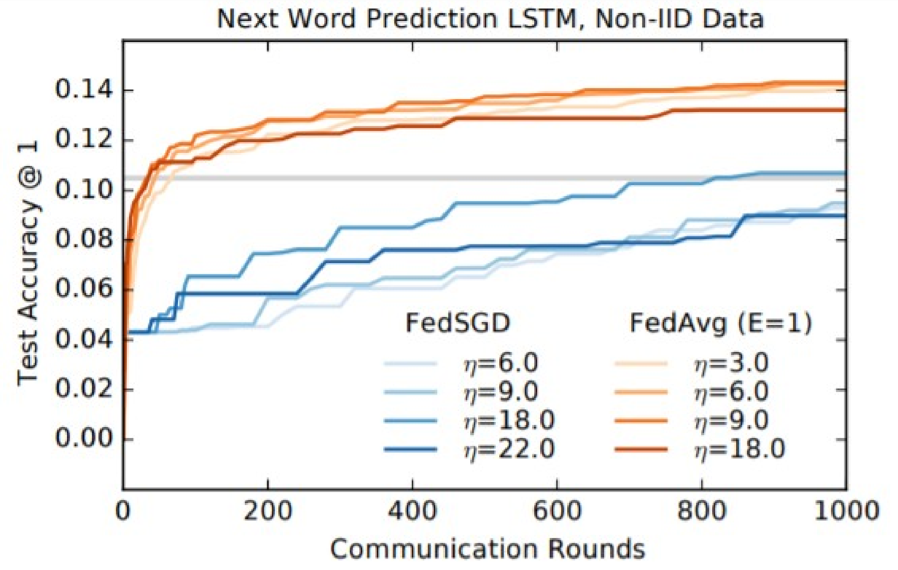


Acc.	80%		82%		85%	
SGD	18000	(—)	31000	(—)	99000	(—)
FedSGD	3750	(4.8×)	6600	(4.7×)	N/A	(—)
FedAvg	280	(64.3×)	630	(49.2×)	2000	(49.5×)

On the CIFAR-10 dataset, FedAVG has less number of communication rounds compares to FedSGD and baseline SGD.

Source: McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics, 1273–1282.

Experimental Results



On a large-scale LSTM experiment, FedSGD with η = 18.0 required 820 rounds to reach 10.5%, while FedAvg with η=9.0 reached an accuracy of 10.5% in only 35 rounds, which is 23X fewer than fedSGD.

Source: McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics, 1273–1282.