Unified Analysis of Federated Learning with Arbitrary Client Participation

Theophilus Siameh

March, 2024

Advisor: Dr. Liu Chun-Hung

Outline

Definition

- Understand how partial client participation affects convergence.
- A unified convergence analysis for FL with arbitrary client participation

Contributions

- Introduced a generalized FedAvg algorithm which amplifies the updates aggregated over multiple rounds, where the amplification interval P can be tuned for the best convergence.
- Presented a novel analysis and unified methodology for obtaining competitive convergence upper bounds with arbitrary client participation patters.
- According to them they also discussed important analysis from both theoretical and experimental results.

Datasets

- FashionMNIST used total rounds of $2,000$
- CIFAR-10 used total rounds of $4,000$

Dataset Partitioning

Partitioning a dataset into multiple subsets:

1. **NodeSampler Class**: This class is used for sampling nodes (or subsets) from the dataset. It initializes with the number of nodes and a boolean indicating whether permutation is enabled. The sample method samples a subset of nodes from the dataset. If permutation is enabled, it shuffles the dataset indices and then selects nodes from the shuffled list.
2. **DatasetSplit Class**: This class represents a split of the dataset. It takes the original dataset and a list of indices indicating which samples are included in this split. The `len` method returns the length of the split, and the `getitem` method allows retrieving items from the split.

Dataset Partitioning

1. **mnist_partition Function**: This function partitions the MNIST dataset into subsets for each node. It initializes an empty dictionary where keys represent node IDs and values are arrays of indices. Then, it iterates over the dataset labels and assigns each sample to a node based on a random choice with probability defined by mixing_ratio or based on the label modulo the number of nodes.
2. **cifar_partition Function**: Similar to mnist_partition, but for CIFAR-10 dataset. It partitions the dataset based on labels into subsets for each node.
3. **split_data Function**: This function selects the appropriate partitioning function based on the dataset name and calls it to partition the data.

Continue ...

- The code doesn't directly consider the size of the dataset when partitioning it among nodes. Instead, it partitions the dataset based on the number of nodes (n_nodes) and the distribution of labels within the dataset.
- the size of the dataset does affect the partitioning process because the distribution of labels, which is used for partitioning, might vary depending on the dataset size.
- The dataset was partitioned into unequal sizes.
- the dataset is partitioned based on the labels of the samples. Each label is assigned to a specific node, and the samples belonging to that label are allocated to that node. Since the distribution of labels in the dataset may not be uniform, the resulting partitions will likely have different sizes.

Continue

- dataset was not partitioned into equal sizes. The partitioning logic in both the mnist_partition and cifar_partition functions does not aim for equal-sized partitions. Instead, it distributes the samples unevenly among the nodes based on the labels of the samples.
- Additionally, in the code, there's a mechanism to handle cases where the number of nodes (n_nodes) exceeds the number of unique labels in the dataset. In such cases, the labels are evenly spread across the nodes, potentially leading to further variation in partition sizes based on label frequency and the number of nodes.

Client Participation Methods

1. Total clients(N $=$ 250)
2. Subset of clients used (S $=$ 10)
3. FedAvg with large learning rate ($\gamma = 0.1, \gamma = 0.05, \eta = 1$) for amplication scenario

Client Participation Methods

1. Algorithm that waits for all clients
2. Algorithm that selects only a random clients
3. GridSearch learning rates: $\lambda = 1, 0.1, 0.01, 0.001, 0.0001$

Client Availability Methods

1. Always
2. Periodic

Experiments

1. Algorithm 1: with amplification gives the best performance given $P = 500$ clients
2. Wait Mini Batch (partial participation)
3. Wait Full
4. Algorithm 1 without amplification
5. Algorithm 2 with amplification

Experiments

1. $\eta = 1$ for case where clients are always available with no amplification
2. $\eta > 1$ for case where clients are always periodically available
3. Simulation with 10 different random seeds for FashionMNIST
4. Simulation with 5 different random seeds for CIFAR-10
5. Applied moving average over a window length of 3%

Prior Works: Assuming unequal dataset sizes

Given:

$$w_m = \frac{|s_m|}{\sum_{i=1}^{M} |s_i|}$$

Consider three datasets s_1, s_2, s_3 with sizes $|s_1| = 100$, $|s_2| = 200$, and $|s_3| = 300$.
For $m = 3$:

$$w_3 = \frac{|s_3|}{\sum_{i=1}^{3} |s_i|} = \frac{300}{100 + 200 + 300} = \frac{300}{600} = 0.5$$

This means the dataset s_3 constitutes 50% of the total data points in the combined datasets $s_1 = 0.166$, $s_2 = 0.333$, and $s_3 = 0.5$.

Optimization Problem: Our work

The given expression is:

$$w_m(\alpha) = \frac{\alpha_m \cdot |s_m|}{\sum_{i=1}^{M} \alpha_i \cdot |s_i|}$$

This expression is a weighted ratio where α_m are additional weights associated with each dataset s_m.

How do we compute these α_m's ?

Optimization Problem Cont...

1. Numerator ($\alpha_m \cdot |s_m|$):
   - $|s_m|$: The size (cardinality) of the m-th dataset.
   - α_m: A weight factor assigned to the m-th dataset.
   - The product $\alpha_m \cdot |s_m|$ gives a weighted size for the m-th dataset.
2. Denominator ($\sum_{i=1}^{M} \alpha_i \cdot |s_i|$):
   - This represents the sum of the weighted sizes of all datasets from s_1 to s_M.
   - It is the total weighted size of all datasets.
3. Ratio ($w_m(\alpha)$):
   - The ratio $w_m(\alpha)$ represents the proportion of the total weighted data points that are in the m-th dataset.
   - Essentially, it indicates the relative importance of the m-th dataset after considering both its size and the weight α_m.

Example

Let's consider an example with three datasets and specific weights:
- Dataset sizes: , $|s_1| = 100$, $|s_2| = 200$, $|s_3| = 300$
- Weight factors: , $\alpha_1 = 0.5$, $\alpha_2 = 1.5$, $\alpha_3 = 1.0$

Now, we calculate $w_m(\alpha)$ for each dataset s_m:

1. Calculate the numerator for each dataset: , $\alpha_1 \cdot |s_1| = 0.5 \cdot 100 = 50$, $\alpha_2 \cdot |s_2| = 1.5 \cdot 200 = 300$, $\alpha_3 \cdot |s_3| = 1.0 \cdot 300 = 300$
2. Calculate the denominator: - $\sum_{i=1}^{3} \alpha_i \cdot |s_i| = 50 + 300 + 300 = 650$
3. Calculate the weighted ratio $w_m(\alpha)$: , $w_1(\alpha) = \frac{50}{650} \approx 0.077$, $w_2(\alpha) = \frac{300}{650} \approx 0.462$, $w_3(\alpha) = \frac{300}{650} \approx 0.462$

Interpretation: Case of unequal dataset sizes

- Dataset s_1: - Despite having a size of 100, its weight factor $\alpha_1 = 0.5$ reduces its relative importance, resulting in a weighted ratio of approximately 0.077.
- Dataset s_2: - With a size of 200 and a higher weight factor $\alpha_2 = 1.5$, its relative importance is increased, giving it a weighted ratio of approximately 0.462.
- Dataset s_3: - Having the largest size of 300 and a weight factor of 1.0, its weighted ratio is also approximately 0.462, indicating it has significant importance in the total weighted context.

Remarks

1. This weighted ratio $w_m(\alpha)$ provides a way to combine both the size and the importance (through α factors) of datasets when determining their relative significance.
2. np.random.permutation used in the code: is a function that randomly permutes a sequence or returns a permuted range.
3. Find out which FedAVG used?: In the code base, when $P = 1$ implies FedAVG else if $P > 1$ implies Generalized FedAVG. P is the number of rounds.

Hardware

1. RTX 3070 GPU with 300,0000 training rounds, took a day to complete for FashionMNIST.
2. CIFAR-10, 600,000 rounds took about 3days to complete.

Future

1. Include analysis of advanced FL algorithms and more detailed empirical study.

Thank You!

ts2434@msstate.edu

References

A Unified Analysis of Federated Learning with Arbitrary Client Participation https://arxiv.org/pdf/2205.13648.pdf