Partial Weight Exchange in Federated Learning

Yuchong Zhang & David Emerson

July 24, 2023

1 Introduction

Federated learning (FL) is the machine learning setting where training data is distributed across multiple clients, and we wish to learn a model that performs well on each client's data without collecting all the data in one place. The most common paradigm of FL is FedAvg, which performs local training on each client for several epochs and takes the average of the local models on the server's side to obtain a global model. This process is repeated several rounds to produce the eventual global model. FedAvg faces challenges such as the heterogeneity of data distributions between the clients and high communication costs between the server and clients. Recently, authors of [1] conducted experiments of training large language models in the FL setting. In the step where each client sends its model to the server for aggregation, the authors used a simple heuristic to select only certain parts of the model to be exchanged with the server rather than exchanging the entire model, as is commonly done in FedAvg. Their experiments suggest that this heuristic has the potential of significantly reducing communication costs without affecting model performance too much. Here, we implement this partial weight exchange paradigm and conduct our own experiments to evaluate its effectiveness.

2 The Algorithm and Motivation

Now we describe the heuristic used in [1] for selecting the weights to be exchanged. Let $\theta > 0$. Let c denote client c and w^t denote any tensor in the model, where t is simply some index. Let w^t_r be the value of w^t after completing local training round r. Then the tensors that are selected to be sent to the server is the set:

$$DP^r_\theta = \{w^t_r : \left\| w^t_r - w^t_{r-1} \right\| \geq \theta\}$$

That is, at the end of each local training round, we select tensors which deviate in terms of $l2$-norm from their initial values at the beginning of that training round by at least θ. On the server's side, once it receive all tensors from all the clients, weighted averaging is performed in a per-tensor fashion, where the weights used are proportional to the amount of training data each client has.

The motivation behind this selection criterion is that the tensors that change a lot in$l2$ norm in local training should be in some sense more "responsible" for learning the knowledge of a client's domain.

In practice, it can be very hard to find the appropriate value of the threshold θ, and there is no reason to believe that a fixed value of θ would be appropriate throughout training. So instead of directly using the threshold value for selecting the tensors, we use a percentile instead. More precisely, let $0 < p < 1$, and suppose our model has M tensors in total. First, for each tensor w^t, we calculate its norm change $\left\| w^t_r - w^t_{r-1} \right\|$ in round r. We then select the top $k = \lceil p \cdot M \rceil$ tensors that have the largest norm change.

3 Experiment Setup

We use RoBERTa [2] on the AG News dataset [3].

AG News is a text classification dataset with 4 classes, which comes with a training set and a test set. For each client, we first randomly split out ten percent of the training set as a validation set. To model the heterogeneity of client data distributions, we subsample a portion of the training data where the numbers of data points in different classes are disproportionate. To achieve this, we draw a vector from the Dirichlet distribution that determines the percentage of each class. The Dirichlet distribution of order K is a distribution over vectors $x \in \mathbb{R}^K$ such that $\sum_{i=1}^{K} x_K = 1$ and $x_i \in [0, 1]$ for all i. It is parameterized by $\beta_1, \ldots, \beta_K > 0$ and has density function

$$f(x_1, \ldots, x_K; \beta_1 \ldots, \beta_K) = \frac{1}{B(\beta)} \prod_{i=1}^{K} x_i^{\beta_i - 1}$$

where B is the multivariate beta function. For our purpose, we assume that $\beta_1 = \cdots = \beta_K = \beta$. The value of β determines the heterogeneity of the data distributions. The larger β is, the more homogeneous the distributions are, and conversely, the smaller β is, the more heterogeneous the distributions are.

Our experiments consists of 4 federated training rounds on 5 clients. In each round, each clients performs 1 epoch of local training with batch size 4. Each client first samples its own training set in the manner described above, and after completing every local training round, the client is evaluated on the validation set. Finally, after all 4 rounds are completed, every client is evaluated on the test set and sends the evaluation results to the server. We perform this experiment with three levels of heterogeneity: $\beta = 1.0, 0.5, 0.25$, and the exchange percentage $p = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 1.0$, where $p = 1.0$ is equivalent to full weight exchange as is done in FedAvg.

4 Experiment Results

4.1 Test Accuracy

We first show the test accuracy of the final model obtained using different exchange percentage under 3 different levels of data heterogeneity.

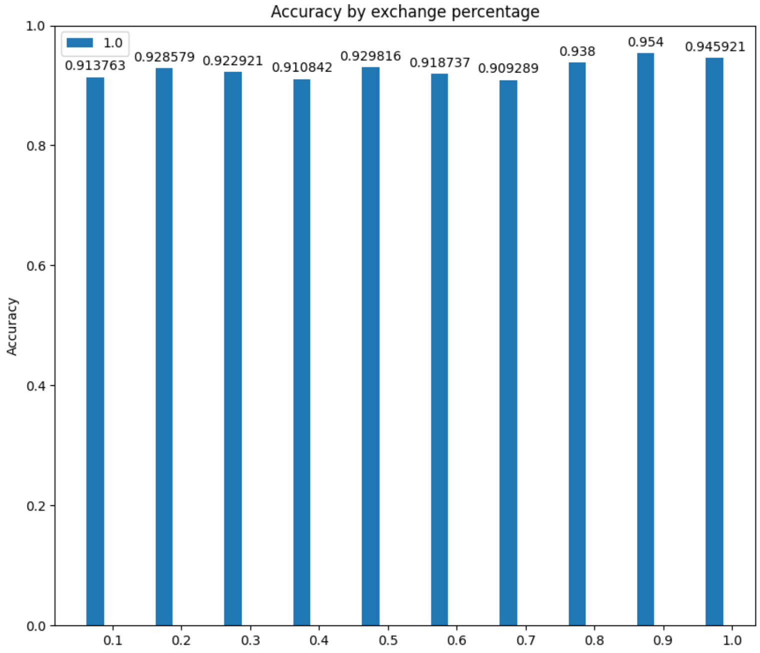

Figure 1: $\beta = 1$

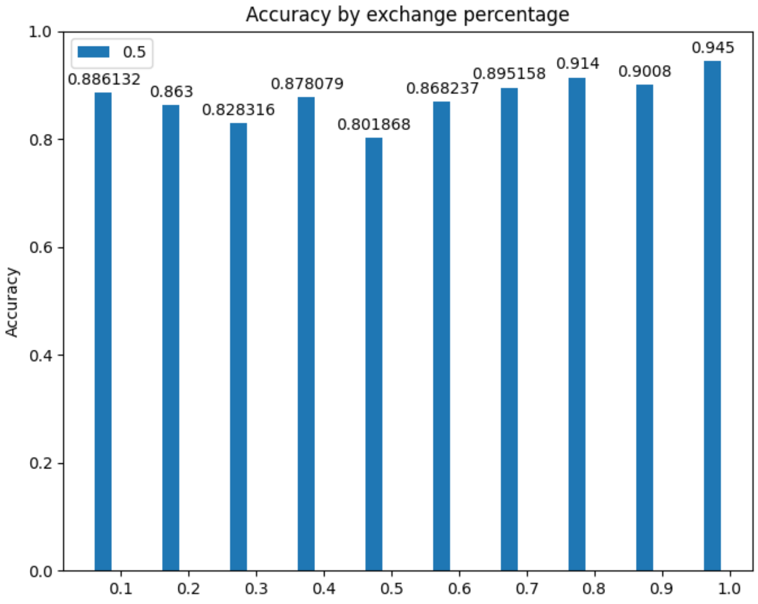

Figure 2: $\beta = 0.5$

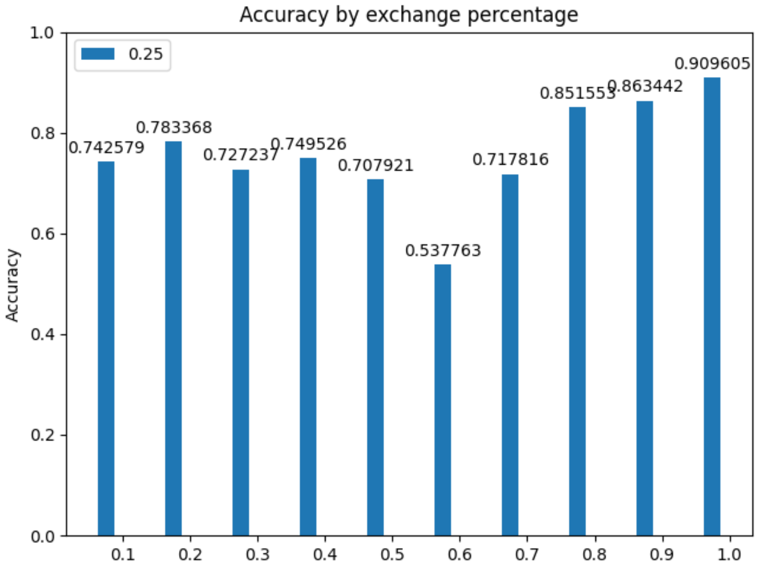

Figure 3: $\beta = 0.25$

4.2 Some Examples of f1 scores

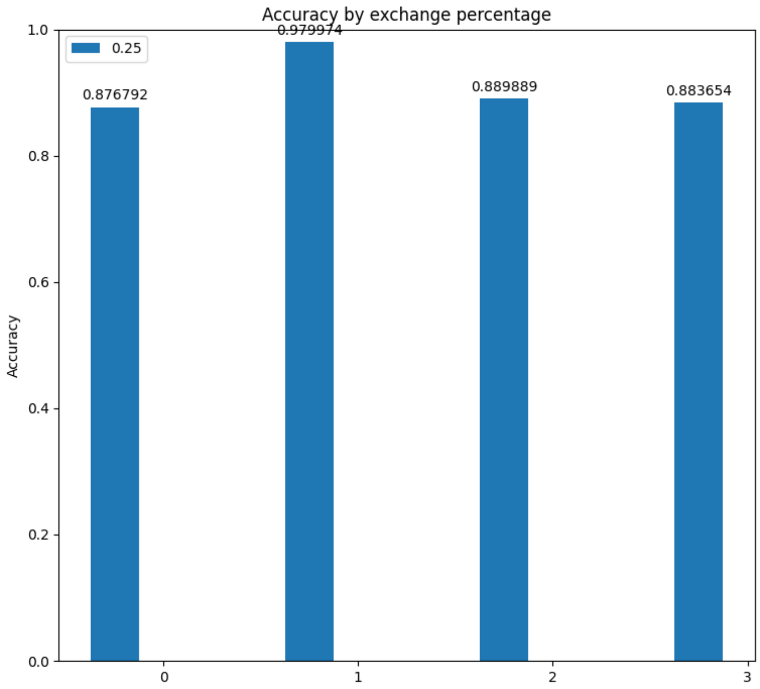

Figure 4: $\beta = 0.5, p = 0.8$

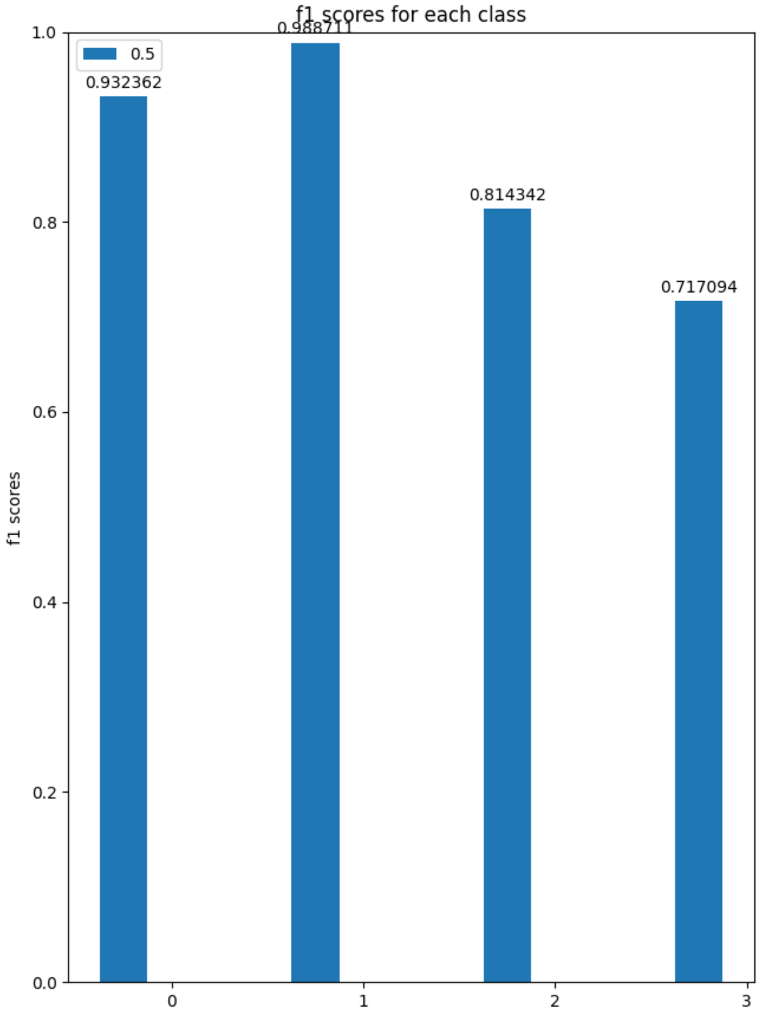

Figure 5: $\beta = 0.5, p = 0.8$

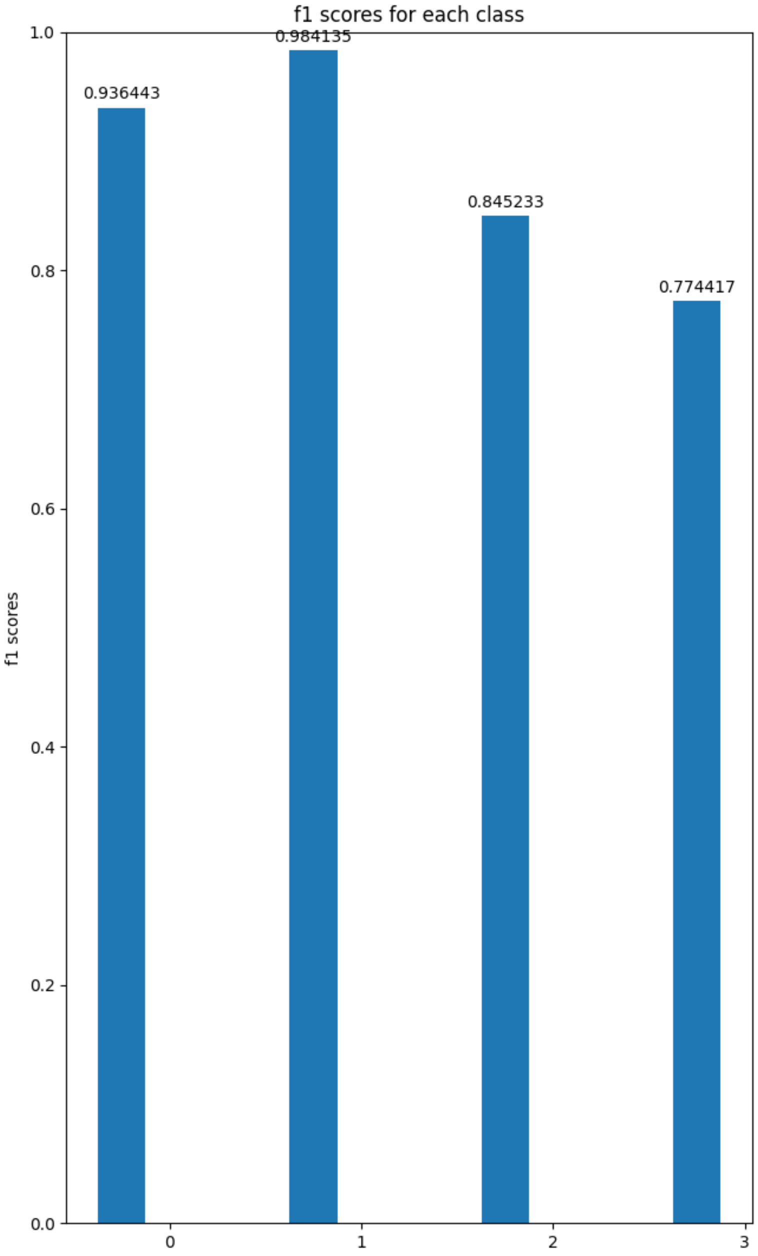

Figure 6: $\beta = 0.25, p = 0.8$

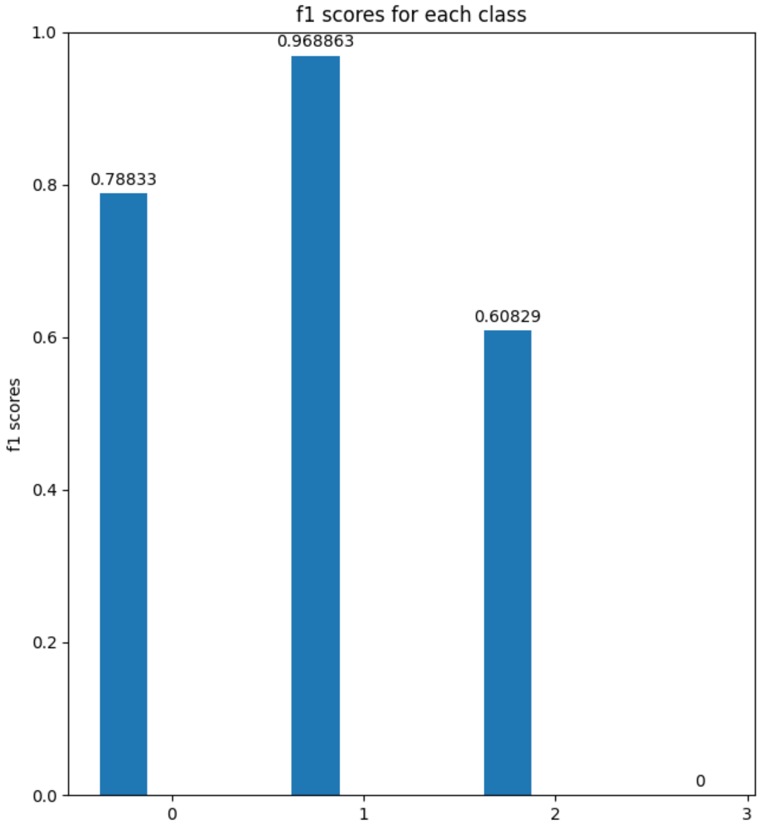

Figure 7: $\beta = 0.25, p = 0.8$

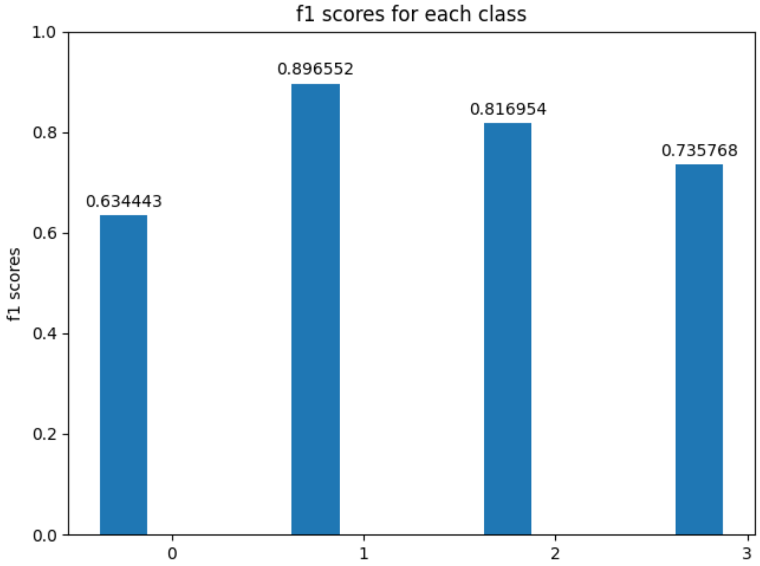

Figure 8: $\beta = 0.25, p = 0.1$

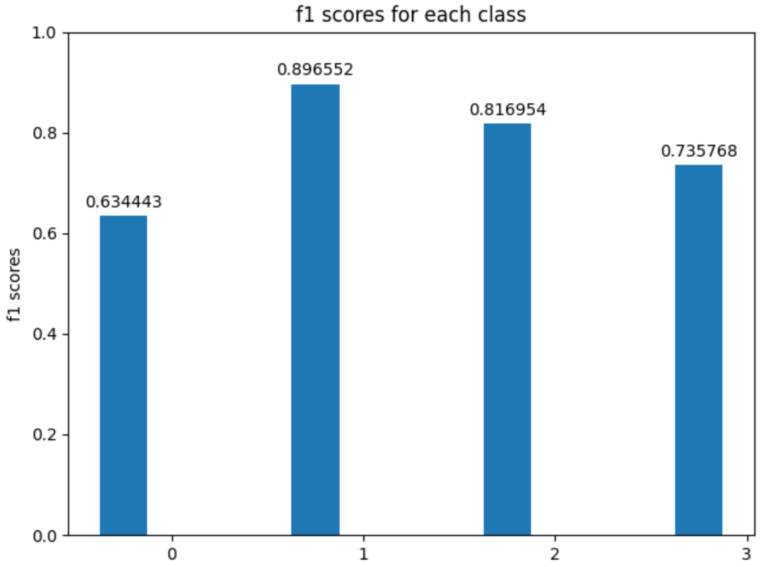

Figure 9: $\beta = 0.25, p = 0.1$

5 Analysis

Our experiments seem to show that when the heterogeneity between client data distributions is low, partial weight exchange has little impact on the final model performance, as can be seen in the results when $\beta = 1$.

When $\beta = 0.5$, the detriment of partial exchange on model performance becomes more noticeable. Even when the final accuracy is high, the model can still have somewhat skewed f1 scores.

When $\beta = 0.25$, partial exchange significantly drops model performance. Both the testing accuracy and f1 scores are worse when partial exchange is used, and the f1 scores is even more skewed.

When data heterogeneity is high, there seems to be a trend that as the exchange percentage decreases, model performance first decreases and then increases. As can be seen both when $\beta = 0.5$ and $\beta = 0.25$, the exchange rate that yields the worst model performance seems to be in the middle.

References

[1] Peyman Passban, Tanya Roosta, Rahul Gupta, Ankit Chadha, and Clement Chung. Training mixed-domain translation models via federated learning, 2022.

[2] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[3] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification, 2016.