FeTS Challenge 2022 Task 1: Implementing FedMGDA+ and a new partitioning

Vasilis Siomos[1], Giacomo Tarroni[1], and Jonathan Passerrat-Palmbach[1,2]

[1] City, University of London, UK
[2] Imperial College London, UK

Abstract. Federated Learning is becoming ubiquitous in settings where privacy and data ownership make sharing raw data infeasible. Medical imaging presents a prominent such scenario. Despite fervent interest in Federated Learning from the Medical Imaging community, there is a general lack of standardised test-beds, datasets, and challenges that can fast-track progress in the domain. The Federated Tumour Segmentation Challenge attempts to fill that gap for the task of brain tumour segmentation. For this iteration of FeTS, we present two additional dataset splits for prototyping and test how the FedMGDA+ algorithm performs on the problem. Code for this report is provided at https://github.com/siomvas/FeTS_2022

Keywords: Federated Learning · Tumour Segmentation · Medical Imaging

1 Introduction

In this short study, we take a look at the challenges of the competition, develop two new splits that reduce the idle time and allow us to perform more aggregation calls, and implement the FedMGDA+[5] algorithm to provide a model that performs well across all institutions.

Federated Learning[6] is a collaborative learning paradigm where clients can jointly train a machine learning model without their local data leaving the premises; instead, only model updates are exchanged, aggregated, and redistributed in an iterative process. This inherent data protection mechanism is appealing in all the scenarios where data privacy and ownership are paramount, such as Medical Imaging. However, despite the momentum that research into Federated Learning has gathered [12, 14, 13, 2, 4], there is a distinct lack of standard experiment settings, which are necessary to facilitate fair comparisons [7]. The FeTS initiative[9] is the largest federation of medical institutions, and the FeTS Challenge is one of the first federated learning challenges in the medical imaging community.

Task 1 of the competition concerns the study of robust aggregation methods that leverage the clients' local updates most effectively to produce a global model. for this reason, an infrastructure is provided with only specific modifications allowed in four areas: collaborator sampling, aggregation, hyper-parameter

choice, and dataset partitioning. The infrastructure stack consists of OpenFL[11] to handle the federated logic and GaNDLF[10] to handle the deep learning logic.

2 Data

FeTS 2022 is the second iteration of the Federated Tumour Segmentation Challenge. The challenge dataset and format are based on the BraTS Challenge[1], except the data in FeTS cannot leave the local sites. This year's training set contains mpMRI (T1, T1-Gd, T2, T2-FLAIR) brain scans from 1251 patients across 33 institutions. The data has been centrally pre-processed and expertly annotated, as described in the challenge manuscript[9].

2.1 Partitioning

The organisers have provided three splits for the challenge. A split corresponds to the number of clients in the federation, and the patient records at each simulated federated site.

- A small split with a handful of samples for debugging purposes.
- The natural split; each collaborator corresponds to a different physical institution.
- An artificial refinement of the natural split, where the records from the biggest collaborators were split to different artificial collaborators based on each record's tumour size compared to the median tumour size for the original collaborator. More concretely, the 5 biggest collaborators were split into 3 new collaborators each.

Early on in the challenge, we emphasised training using the original split, as it introduces no confounding variables, such as an appropriate metric or threshold to use for splintering a collaborator's samples or aggregating the samples from smaller collaborators. However, the challenge's simulated time limit means the original split is quite restrictive; in the original split, the largest contributor takes 25 simulated hours to complete one epoch of training, capping the maximum possible number of rounds to 6 (using the minimum of 1 local epoch). Moreover, the number of local epochs has to be the same for all collaborators, essentially imposing idle time for every collaborator proportional to the difference between the size of their data and the size of the biggest collaborator's data.

This incentivizes us to fraction the dataset into smaller, more uniformly sized participants to minimise idle time and maximise the number of federated rounds that fit within the simulation threshold.

Additionally, since the total number of records is high, training on the full dataset takes a lot of wall time. For the purposes of prototyping and exploring the potential of different hyperparameters and aggregation methods, we propose the following splits, based on the original method of splitting the largest collaborators according to the tumour size:

– A small split, which includes a total of 117 patients (10% of the original dataset) from 7 collaborators, containing the records of all the physical collaborators that contributed between 10 and 30 records. This ensures no collaborator dominates the split, and that the heterogeneity mimics a real-world scenario.

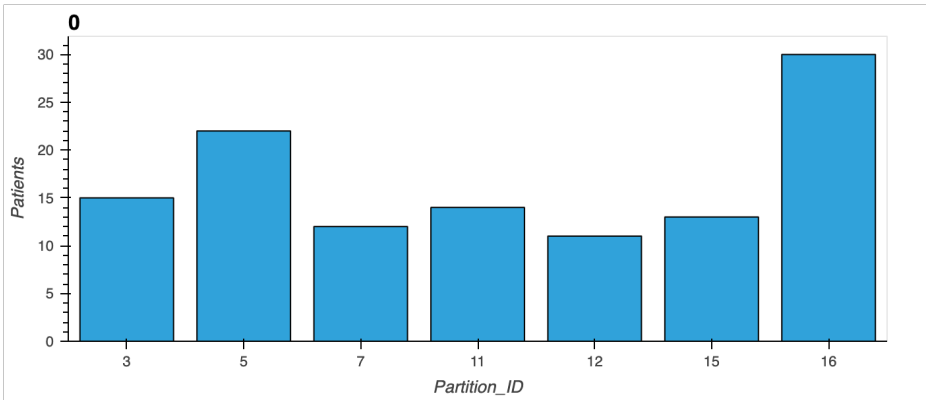

Fig. 1. The small split for prototyping.

– A medium split, which includes all the natural collaborators except the two biggest ones, which we split into 10 bins each, based on tumour volume, then include the middle two bins only. This leads to a dataset with 536 patients (42% of the original size) from 25 collaborators. We find this to present an adequate middle ground between the small and original split, both in imbalance and size.

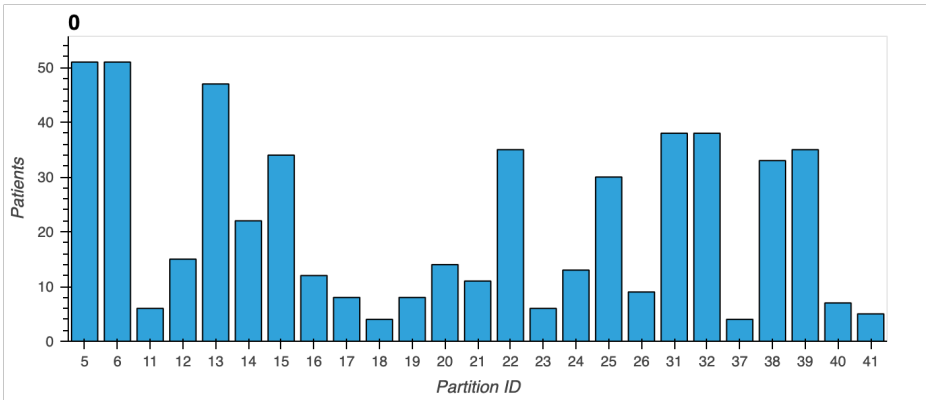

Fig. 2. The medium split for scaling up experiments.

– An alternative split of the full dataset, for which we use the same methodology as the original artificial partitioning, but we splinter the two largest participants into 10 instead of 3 quantiles, based on tumour volume, resulting in 41 collaborators. As shown in figure 3, this new split greatly smooths out the imbalance of the original split, alleviating the aforementioned problems of maximum federated rounds and idle time on smaller collaborators.

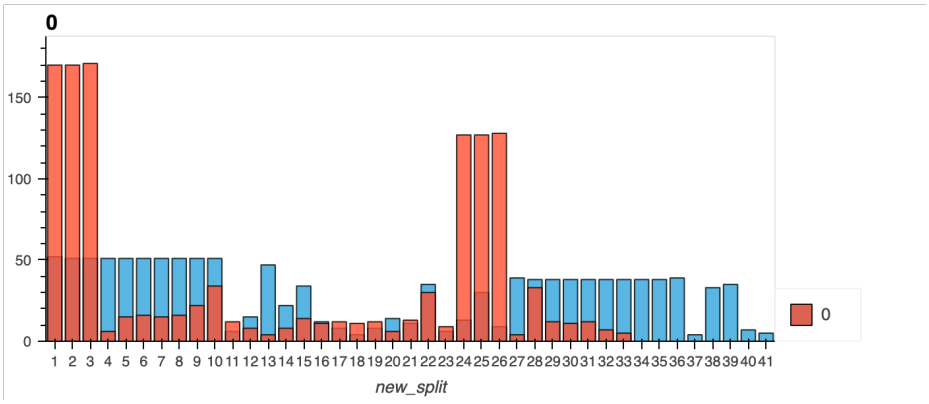

Fig. 3. The full new split compared to the given artificial split.

3 Methods

3.1 Sampling participants

One crucial dimension of control in federated learning systems is the choice of the participating clients in each round. This choice is especially important in scenarios with millions of possible clients, such as smartphone users. In cross-silo scenarios, like the one for the challenge, where all participants can participate in every round, we argue constant and full participation is needed to allow each local site's characteristics to influence the model, in line with our ultimate goal of personalised FL models, as described in section 1. Hence, we used all the collaborators in every round.

3.2 Hyperparameter choice

The default available hyperparameters are the learning rate and the number of epochs e. As e had to be an integer, and, as explained in section 2, more local epochs extend the idle time for all collaborators except the largest, we keep e fixed at 1. Regarding the learning rate, we found that doubling the default learning rate to 1e-4 improved performance.

Additionally, while the MGDA algorithm automatically tunes the collaborator weights without any hyperparameters, FedMGDA+ interpolates between that result and a uniform weighing based on a hyper-parameter ϵ. Setting ϵ equal to 0 recovers FedAvg, while setting it equal to 1 recovers MGDA. We used ϵ=0.5 for our experiments.

3.3 Aggregation function

Task 1 focuses on aggregation methods that can effectively pool the information from the participants' local updates. The collaborator computes the weights of the global model as the weighted sum of the local weights:

$$\mathbf{w}^r = \sum_{i=1}^{N} \lambda_i \mathbf{w}_i^r \tag{1}$$

The popular Federated Averaging algorithm[8] sets these coefficients to be the ratio of the clients' local data to the total dataset size, thus biasing the update towards the optima of the contributors with the largest number of records. Instead of treating the Federated Learning aggregation problem as a server trying to find a single model that performs best *on average* across different client distributions, we can attempt to find a Pareto optimal solution, such that no client is disadvantaged by the aggregation. Especially in cross-silo[6] scenarios, where the trained model will be deployed to the institutions that participated in the training, or others with local datasets similar to those of the original participants, instead of a heuristic approach to determine the aggregation weights, we can use the Multiple Gradient Descent Algorithm, borrowed from multi-objective optimisation, to determine the common descent direction for all participants. This ensures optimisation moves only towards areas of the solution space that do not worsen the model's performance on any client. One such method is the FedMGDA+[5] algorithm, which interpolates between an even weighing of $1/N$ and the common descent direction as produced by the Multiple Gradient Descent Algorithm (MGDA)[3].

For every federated round, every participating client executes an SGD step, in parallel:

$$\mathbf{w}_i^{\tau+1} = \mathbf{w}_i - \eta \nabla f_i(w_i) \tag{2}$$

Instead of equation 1, MGDA uses the following update rule:

$$\mathbf{w}_i^{\tau+1} = \mathbf{w}_i - \eta d_t, \qquad d_t = J_f(w_t)\lambda_t^*, \qquad \lambda_t^* = \arg\min_{\lambda \in \Delta} \parallel J_f(w_t)\lambda \parallel \tag{3}$$

Here η is a server learning rate (assumed to be 1 in our case), and the vector of coefficients λ is found by solving a simple quadratic programming problem once per federated round. We note that the MGDA algorithm requires the gradient of the models to compute the Jacobian, and the vector λ. In the case of multiple local epochs, we would instead have to approximate the gradients by the model delta, i.e. the difference between the weight values at the beginning and end of

the round, but since we use a single epoch to minimise the idle time as explained in section 1, the delta *is* the model gradient.

FedMGDA+ refines that, by setting the update rule to be:

$$\lambda_t^* = \arg \min_{\lambda \in \Delta, \|\lambda - \lambda_0\|_\infty \leq \epsilon} \| J_f(w_t)\lambda \| \tag{4}$$

meaning the solution for λ is forced to lie close to the FedAvg solution, with the closeness dictated by ϵ.

4 Results

Despite the theoretical reasoning, we found that FedMGDA+ actually underperformed the FedAvg baseline in the challenge's setting. Our hypothesis is that due to the large number of participants in our split, and the small number of records in some participants, optimising towards the common descent direction performs worse on average than biasing the updates towards the biggest collaborators as FedAvg does.

Table 1. ET test scores using our new split

ET	Dice	H95	Sensitivity	Specificity
FedMGDA+	0.56	44.13	0.52	0.99
FedAvg	0.67	36.57	0.65	0.99

Table 2. TC test scores using our new split

TC	Dice	H95	Sensitivity	Specificity
FedMGDA+	0.59	30.25	0.57	0.99
FedAvg	0.69	23.63	0.69	0.99

Table 3. WT test scores using our new split

WT	Dice	H95	Sensitivity	Specificity
FedMGDA+	0.67	24.73	0.59	0.99
FedAvg	0.77	35.62	0.81	0.99

5 Discussion

5.1 Memory Requirements

We found training with the full dataset to be very memory demanding, requiring 130 GB of RAM during the preprocessing phase, and increasingly more of it while training continued. After investigation, we found that the culprit is the `pin_memory` argument in the configuration files, which caused the whole dataset to be pinned to CPU memory during the initial pre-processing. By setting this to `false`, we find that initial processing only takes 10GB of RAM, with no negative performance impact on training; on the contrary, training was significantly sped up for a 128GB RAM system which previously had to use disk swapping to run the experiment. There is an additional memory leak that causes memory to fill proportionally to the number of rounds in an ongoing run, but restoring from a checkpoint can alleviate that.

5.2 Time constraints

This year, all the collaborators had to use the same number of local epochs regardless of their size. Additionally, there was a fixed time limit calculated based on the time the biggest collaborator needed to complete training.

The combination of these two factors imposed idle/wasted time in all but the biggest collaborators, and perhaps even more importantly, heavily deincentivised training for more than a single local epoch, as the idling/wasted time increases linearly with the number of local epochs.

This limits the solution space, while not mirroring real-world conditions.

Bibliography

[1] Baid, U., Ghodasara, S., Mohan, S., Bilello, M., Calabrese, E., Colak, E., Farahani, K., Kalpathy-Cramer, J., Kitamura, F.C., Pati, S., et al.: The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification. arXiv preprint arXiv:2107.02314 (2021)

[2] Bernecker, T., Peters, A., Schlett, C.L., Bamberg, F., Theis, F., Rueckert, D., Weiß, J., Albarqouni, S.: Fednorm: Modality-based normalization in federated learning for multi-modal liver segmentation (2022). https://doi.org/10.48550/ARXIV.2205.11096, https://arxiv.org/abs/2205.11096

[3] Désidéri, J.A.: Multiple-gradient descent algorithm (mgda) for multiobjective optimization. Comptes Rendus Mathematique **350**(5-6), 313–318 (2012)

[4] Guo, P., Yang, D., Hatamizadeh, A., Xu, A., Xu, Z., Li, W., Zhao, C., Xu, D., Harmon, S., Turkbey, E., et al.: Auto-fedrl: Federated hyperparameter optimization for multi-institutional medical image segmentation. arXiv preprint arXiv:2203.06338 (2022)

[5] Hu, Z., Shaloudegi, K., Zhang, G., Yu, Y.: Federated learning meets multiobjective optimization. IEEE Transactions on Network Science and Engineering (2022)

[6] Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al.: Advances and open problems in federated learning. Foundations and Trends® in Machine Learning **14**(1–2), 1–210 (2021)

[7] Karargyris, A., Umeton, R., Sheller, M.J., Aristizabal, A., George, J., Bala, S., Beutel, D.J., Bittorf, V., Chaudhari, A., Chowdhury, A., Coleman, C., Desinghu, B., Diamos, G., Dutta, D., Feddema, D., Fursin, G., Guo, J., Huang, X., Kanter, D., Kashyap, S., Lane, N., Mallick, I., Mascagni, P., Mehta, V., Natarajan, V., Nikolov, N., Padoy, N., Pekhimenko, G., Reddi, V.J., Reina, G.A., Ribalta, P., Rosenthal, J., Singh, A., Thiagarajan, J.J., Wuest, A., Xenochristou, M., Xu, D., Yadav, P., Rosenthal, M., Loda, M., Johnson, J.M., Mattson, P.: Medperf: Open benchmarking platform for medical artificial intelligence using federated evaluation (2021). https://doi.org/10.48550/ARXIV.2110.01406, https://arxiv.org/abs/2110.01406

[8] Konečnỳ, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492 (2016)

[9] Pati, S., Baid, U., Zenk, M., Edwards, B., Sheller, M., Reina, G.A., Foley, P., Gruzdev, A., Martin, J., Albarqouni, S., et al.: The federated tumor segmentation (fets) challenge. arXiv preprint arXiv:2105.05874 (2021)

[10] Pati, S., Thakur, S.P., Bhalerao, M., Thermos, S., Baid, U., Gotkowski, K., Gonzalez, C., Guley, O., Hamamci, I.E., Er, S., et al.: Gandlf: A generally

nuanced deep learning framework for scalable end-to-end clinical workflows in medical imaging. arXiv preprint arXiv:2103.01006 (2021)

[11] Reina, G.A., Gruzdev, A., Foley, P., Perepelkina, O., Sharma, M., Davidyuk, I., Trushkin, I., Radionov, M., Mokrov, A., Agapov, D., et al.: Openfl: An open-source framework for federated learning. arXiv preprint arXiv:2105.06413 (2021)

[12] Sheller, M.J., Edwards, B., Reina, G.A., Martin, J., Pati, S., Kotrotsou, A., Milchenko, M., Xu, W., Marcus, D., Colen, R.R., et al.: Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Scientific reports **10**(1), 1–12 (2020)

[13] Tedeschini, B.C., Savazzi, S., Stoklasa, R., Barbieri, L., Stathopoulos, I., Nicoli, M., Serio, L.: Decentralized federated learning for healthcare networks: A case study on tumor segmentation. IEEE Access **10**, 8693–8708 (2022)

[14] Xu, A., Li, W., Guo, P., Yang, D., Roth, H.R., Hatamizadeh, A., Zhao, C., Xu, D., Huang, H., Xu, Z.: Closing the generalization gap of cross-silo federated medical image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20866–20875 (2022)