Efficient Federated Tumor Segmentation via Normalized Tensor Aggregation and Client Pruning

Youtan Yin[2], Hongzheng Yang[3], Quande Liu[1(✉)], Meirui Jiang[1], Cheng Chen[1], Qi Dou[1], and Pheng-Ann Heng[1,4]

[1] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China
{qdliu,qdou}@cse.cuhk.edu.hk
[2] Department of Computer Science and Technology, Zhejiang University, Hangzhou, China
[3] Department of Computer Science and Engineering, Beihang University, Beijing, China
[4] Guangdong-Hong Kong-Macao Joint Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

Abstract. Federated learning, which trains a generic model for different institutions without sharing their data, is a new trend to avoid training with centralized data, which is often impossible due to privacy issues. The Federated Tumor Segmentation (FeTS) Challenge 2021 has two tasks for participants. Task 1 aims at effective weight aggregation methods given a pre-defined segmentation algorithm for clients training. While task 2 looks for robust segmentation algorithms evaluated on unseen data from remote independent institutions. In federated learning, heterogeneity in the local clients' datasets and training speeds results in non-negligible variations between clients in each aggregation round. The naive weighted average aggregation of such models causes objective inconsistency. As for task 1, we devise a tensor normalization approach to solve the objective inconsistency. Furthermore, we propose a client pruning strategy to alleviate the negative impact on the convergence time caused by the uneven training time among local clients. Our method achieves a projected convergence score of 74.32% during the training phase. For task 2, we dynamically adapt model weights at test time by minimizing the entropy loss to address the domain shifting problem for unseen data evaluation. Our method finally achieves dice scores of 90.67%, 86.23%, and 78.90% for the whole tumor, tumor core, and enhancing tumor, respectively, on the task's validation data. Overall, the proposed solution ranked first for task 2 and third for task 1 in the FeTS Challenge 2021.

Keywords: Federated learning · Brain tumor segmentation · nnU-Net · Test-time adaptation

The first two authors contributed equally. The work was done when they did summer internship at CUHK.

A. Crimi and S. Bakas (Eds.): BrainLes 2021, LNCS 12963, pp. 433–443, 2022.
https://doi.org/10.1007/978-3-031-09002-8_38

1 Introduction

It is widely accepted that segmentation for magnetic resonance imaging of the brain is beneficial for clinical treatment [1]. Although the Brain Tumor Segmentation (BraTS) challenges' [2–4] state-of-the-art average Dice has reached an uplifting score as high as around 85 [5,6] on the test set, applying these algorithms in the real world remains challenging due to regulatory and privacy hurdles [7]. Clinical institutions usually do not allow access to each other's data, in which scenario models cannot be trained with centralized data. Federated learning (FL) [8–19] trains local models using each institution's data and aggregates local models' weights to the global model. This server-clients framework solves the above issue well. In this trend, the Federated Tumor Segmentation (FeTS) Challenge [20] is the first Challenge proposed for FL. The Challenge is structured in two explicit tasks, and we participate in both of the tasks.

Task1 requires customizing core functions of a baseline federated learning system implementation, aiming to improve the baseline consensus models. There are four functions we can adjust: **(i) Aggregation Function:** In which way the clients will aggregate their wights to the global model. **(ii) Collaborator Training Selection:** Which set of collaborators will be selected to train in each aggregation round. **(iii) Hyperparameters for training:** Hyperparameters for training: Hyperparameters for collaborators' training stage. We can only custom two aspects according to the official API. One is the learning rate for each local model, and the other one is batches or epochs per round. Notice that we can only set either batch or epoch, and all collaborators share the same hyperparameters for each communication round. **(iv) Validation Function:** Validation metrics used for local clients and the global model. Except for the above four functions, all other details in the federated learning pipeline have been prepared by the organizers and are the same for all participants. Participants will be ranked considering both the performance and convergence speed in task 1.

For this task, noting that our task provides a heavy, uneven partition of data, we should design methods to address this issue. Meanwhile, this task asks for a general algorithm that can be applied without involving local training settings [21,22] like adjusting gradient descent. Putting all these factors together, we find Fed-Nova [23], which eliminates objective inconsistency while preserving fast error convergence, is the most appropriate aggregation function for this task. In addition, as the convergence speed is also an important metric, we propose a client pruning strategy to "prune" the clients if their local training time is larger than a threshold in the first ten aggregation rounds. This method is proven to be effective in speeding up the convergence process a lot with a cost of slight performance drop in early rounds, thus guaranteeing an increase of the projected convergence score.

Task2 expects robust segmentation algorithms evaluated on unseen data during the testing phase without any limitations to participants. Specifically, participants will submit a trained model with their inference code, and models will be evaluated on data from various remote independent institutions, which is not released to participants.

For this task, we need to design algorithms to enhance the generalization ability of our model to perform better on unseen data. Models work well if the training datasets can represent the distribution of the test datasets. However, there is often a mismatch between training and test cases regarding their acquisition details in medical image segmentation, especially in this Challenge, the test data coming from different institutions. According to [5,24,25], they observe a depressing performance drop when shifting from the training dataset to the test dataset, which confirms the fact that there is a deviation between the training dataset and the test dataset in the FeTS Challenge. We propose to dynamically adapt model's parameters by optimizing the entropy loss calculated on its test dataset predictions. Entropy is an unsupervised objective as it only requires model predictions and not annotations. According to previous research [26], a noticeable performance improvement can be achieved for one iteration on each test set's case without altering training and violating the inferencing time limitation of 180s per case.

2 Solution for Task 1

Task 1 aims to develop effective weight aggregation and clients sampling strategy for real-world medical federated optimization. The local network architecture and training algorithm for tumor segmentation are pre-defined by organizers. Given the data partition across multiple individual institutions, our job is to design task-specific aggregation function for global model updates, clients chosen for each federated round and the training hyper-parameters like learning rate and local epoches.

2.1 Efficient Federated Aggregation with Tensor Normalization

In each communication round, the clients participating in federated optimization are highly heterogeneous in the number of local stochastic gradient descent iterations. To alleviate this problem, we propose to re-normaliaze aggregated tensors at each round. Assuming there are totally m clients, the update rule of federated learning can be written as:

$$x^{t+1} - x^t = \sum_{i=1}^{m} p_i \delta_i^t \tag{1}$$

where p_i is the relative sample size at client i, x^t denotes the model parameters at round t and δ_i^t is the local parameters change at client i.

To alleviate the objective inconsistency problem [23] caused by heterogeneous number of local iterations, we re-weight the aggregated local tensors for fair federated aggregation and rewrite the update rule as:

$$x^{t+1} - x^t = -(\sum_{i=1}^{m} p_i \tau_i) \sum_{i=1}^{m} p_i \frac{\delta_i^t}{\|a_i^t\|_1} \tag{2}$$

where τ_i is the number of local updates at client i which varies widely across all clients and $\|a_i^t\|_1$ is a normalizing scalar defined by how stochastic gradients are locally accumulated, i.e. local optimizers. This update rule is taken from FedNova [23].

For SGD with momentum [27] used in this challenge, a_i can be set as $a_i = [1-\rho^{\tau_i}, 1-\rho^{\tau_i-1}, ..., 1-\rho]/(1-\rho)$ where ρ is the momentum factor.

Comparing to previous algorithms such as FedAvg, local parameters change in our method is re-scaled based on different number of local iterations for each client. Thus the inconsistency optimization can be eliminated and faster convergence is achieved in real clinical scenarios.

2.2 Pruning of Slower Local Clients for Fast Federated Training

In realistic scenarios, federated collaborators are also heterogeneous in different computation, download and upload speeds. Within the same communication round, slower clients will consistently need more wall-clock cpu time to finish the download, upload and training tasks. This speed gap across participating clients will harm the parallel performance of federated learning thus incurring a slower convergence rate.

To address this issue, we design a client pruning mechanism to adaptively filter out some slower clients. Specifically, we collect the simulated time of last round for each client as the pruning basis:

$$Times = [s_1^{t-1}, s_2^{t-1}, ..., s_m^{t-1}] \tag{3}$$

where s_m^t denotes the simulated time for client m at round $t-1$

Then we calculate a threshold μ according to mean value of $Times$ as following:

$$\mu = (\frac{1}{m}\sum_{i=1}^{m} s_i^{t-1}) * \alpha \tag{4}$$

where α is a hyper-parameter to control the filter range. We empirically set it as 0.8. All clients with simulated time above threshold μ will be filtered and do not participate in the federated optimization. This pruning mechanism will be performed only for the first c rounds to save the overall computation time as well as preserve the global model performance. In practice, to collect the simulated time, all clients will participate the federated training at odd rounds and we perform the pruning strategy at even rounds.

2.3 Learning Rate and Local Iterations/epochs

We carried out experiments on static and dynamic learning rate schedules to find out the most appropriate hyper-parameters. Supposing η stands for learning rate, η_0 represents the initial learning rate, t represents aggregation round. Then for static learning rate schedule, learning rate for each round is simply:

$$\eta = \eta_0 \tag{5}$$

For dynamic learning rate schedule, we proposed the polynomial decay in the form:

$$\eta = \eta_0 * (1 - \frac{t}{t_{max}})^{\epsilon} \tag{6}$$

where t_{max} refers to the max training round number and ϵ is the polynomial factor.

As for the iterations/epochs, we conduct experiments under $\eta = 1e - 3$ with static schedule. In general, local epochs between every aggregation round are set to 1 in federated learning because this guarantees the balance between convergence performance and speed. Though we still test other settings and observe that increasing epoch indeed improves performance lightly but with a cost of an increase in training time to convergence while decreasing batches speeds up convergence significantly. However, its final segmentation performance drops sharply, respectively.

3 Solution for Task 2

We base our algorithms on the rank one team of the BraTS 2020 Challenge. We refer to [5] for a detailed description and a thorough baseline.

3.1 Backbone Architecture

We integrate the best BraTS-specific modifications discussed in [5] to complete our backbone. According to the source code (GitHub), these optimizations are further improved after the BraTS 2020 Challenge. We follow the latest ones.

Specifically, this pipeline adopts a pure nnU-Net as network architecture. The input patch is cropped from randomly selected training cases to the size of $128 \times 128 \times 128$, then the encoder generates a $4 \times 4 \times 4$ feature map with five downsampling steps. The number of the encoder's first layer's convolutional kernels is set to 32 and doubled with each layer up to a maximum of 320. The decoder architecture mirrors the encoder, which upsamples the feature map back to the original input size. Leaky ReLUs and BN are used as nonlinearities and normalizations in the network. Furthermore, brain tumor-specific operations involve improved data augmentation, region-based dice loss optimization, and a threshold to filter predictions.

3.2 Dynamic Adaptation of Parameters at Test Time

As mentioned in [26], lower entropy indicates reasonable performance. At the same time, higher entropy reflects notable domain shift. Based on this observation, we implement a test time dynamic parameter adaptation by updating the model's weights to minimize the entropy loss calculated from the prediction on the test set for one iteration. This little trick has been proven to work well and not increase much time in domain shift tasks [28,29].

Specifically, our test-time objective is to minimize the entropy loss L_e (in [26]) of model predictions $\hat{y} = M_\theta(x^t)$, supposing M represents the model itself and θ represents the model's weights. x^t means model input at test time. The entropy loss is given as follows:

$$L_e = -\Sigma_c(\hat{y}_c * \log(\hat{y}_c + \epsilon)) \tag{7}$$

where c represents each class, in this task 3 classes respectively, and ϵ represents an add-on hyperparameter, which avoids the logarithm of zero. As region-based training uses sigmoid activations at the end of the model layers, resulting in three binary classifications, we calculate the mean entropy loss of each region's output.

4 Experiments

4.1 Dataset

Based on the previous BraTS Challenge, the FeTS Challenge provides the MRI dataset with their source information indicating the relationship between each case and their original institutions. Task 1 and task 2 share the same training data with 341 cases [4,20,30–33]. The Challenge also provides information on how to partition the training data into non-IID institutional subsets. There are two partitions. Partition one is the originating institution split, and partition two is derived from partition one by further splitting the "larger" institutions according to whether a record's tumor size fell above or below the mean size for that institution. As a supplement, participants are given un-labeled data, which can be evaluated on the online platform, to assess their algorithms before the testing stage.

We employ pre-processed training and validation data from FeTS 2021 Challenge and mainly use the provided partition-2 for data split across different institutions. In the FeTS 2021 Challenge dataset, three kinds of regions have been manually segmented as annotations, including the whole tumor (consisting of all three original classes), the tumor core (the original non-enhancing and necrosis + enhancing tumor), and the enhancing tumor (the original enhancing tumor). All images have been pre-processed and further partitioned into 22 clients according to their original institutions and tumor sizes. MRI scans can be highly heterogeneous among participating clients in this Challenge as various scanners and image protocols were employed.

4.2 Implementation Details

As for task 1, we implement FedNorm (federated aggregation with tensor normalization) and the client pruning mechanism based on the open-fl framework. The momentum factor in FedNorm was set as 0.6 based on the experimental results in this challenge. For client pruning, we choose to filter out slower clients for the first 10 rounds. We adopt the polynomial decaying learning rate with an

initial value of 5e-4, polynomial factor of 0.9, and max training epoch of 1000. We perform one epoch for local client training at each communication round. All experiments were conducted under a fixed train validation split and random seed to make our results convincing and deterministic.

As for task 2, the backbone's details are described as follow: **(i) Training settings:** The loss function is the sum of dice and cross-entropy loss with an SGD optimizer with the initialize learning rate of 1e-2 and a Nesterov momentum of 0.99, training 200 epochs with 250 iterations. An polynomial learning rate decay schedule which is the same of task 1 is used. The batch size is 2. **(ii) Data augmentation:** We use the following combination in [5]:

- increase the probability of applying rotation and scaling from 0.2 to 0.3.
- increase the scale range from (0.85, 1.25) to (0.65, 1.6)
- select a scaling factor for each axis individually.
- use elastic deformation with a probability of 0.2 and scale range (0, 0.25).
- use additive brightness augmentation with a probability of 0.3.
- Increase the aggressiveness of the gamma augmentation to range (0.5, 1.6).

(iii) Postprocessing threshold: We follow the example of the official implementation directly to filter the prediction values for enhancing tumor regions if there are less than 200 positive predictions. **(iv) Test-time Dynamic adaptation:** The add-on hyperparameter value ϵ is 1e-6 and we update all of the model's weights with a case of test data at each step for one iteration.

4.3 Results

Abbreviations. We will represent our methods using the following abbreviations. **(i) FedAvg:** refers to the weighted average aggregation. **(ii) FedNorm:** refers to the Fed-Nova aggregation described in Sect. 2.1. **(iii) WT:** refers to the whole tumor region. **(iv) TC:** refers to the tumor core region. **(v) ET:** refers to the enhancing tumor region.

Evaluation Metrics. As for task 1, both the performance of our algorithm and convergence speed will be taken into consideration. Specifically, besides the Dice coefficient (Dice) and Hausdorff distance-95% (HD95), a projected convergence score, calculated by multiplying the Dice with the programs' total running time, is used to assess convergence speed. Moreover, during the training process, Sensitivity (TP/P) and Specificity (TN/N) are also provided by the organizers.

For task 2, there is no exact ranking algorithm before we submit our algorithm. So we adopt the commonly used Dice as criteria.

Ablation Study for Task 1. As shown in Table 1, it is observed that FedNorm consistently outperforms FedAvg under dice performance and convergence score metric, demonstrating the effectiveness of our method for fair federated aggregation. Furthermore, we fine-tune the FedNorm scheme by adjusting the (i)learning

Table 1. Ablation study for task 1

Method			Rounds	Dice [%]				Convergence score
Aggre. Func.	Filt. Clien.	Learn. Rate		ET	TC	WT	Mean	
FedAvg	✗	5e-4	40	0.7563	0.6201	0.7655	0.7140	0.7114
FedNorm	✗		40	0.7560	0.6376	0.7698	0.7211	0.7190
FedNorm	first 5	5e-4	50	0.7421	0.6304	0.7633	0.7119	0.7103
	first 10		50	0.7573	**0.6431**	0.7724	**0.7243**	**0.7432**
	first 20		50	0.7543	0.6358	0.7602	0.7168	0.7122
FedNorm	✗	1e-3	40	**0.7897**	0.5930	0.6707	0.6845	0.7167
	✗	5e-4	40	0.7560	0.6376	0.7698	0.7211	0.7190
	✗	dynamic 1e-3	40	0.7597	0.6074	0.7426	0.7033	0.7227
	✗	dynamic 5e-4	40	0.7535	0.6090	**0.7981**	0.7202	0.7261

Table 2. Test results for task 1

Task	LeaderBoard 1				LeaderBoard 2				LeaderBoard 1				LeaderBoard 2			
Label	ET	WT	TC	Avg.	ET	WT	TC	Avg	ET	WT	TC	Avg	ET	WT	TC	Avg
	Dice Coefficient (Dice) ↑								**Hausdorff Distance (HD) ↓**							
Mean	0.5200	0.7027	0.5878	0.6035	0.6689	0.7771	0.6528	0.6996	57.17	43.46	51.71	50.78	42.47	19.85	40.76	34.36
Std	0.2590	0.2348	0.2926	0.2621	0.2791	0.1861	0.2776	0.2476	94.14	28.86	79.59	67.53	95.82	23.80	79.30	66.31

Table 3. Ablation study for task 2

Model	ET	WT	TC	Average
80% clients	0.7696	0.8933	0.8445	0.8358
All training data	0.7782	0.9073	0.8473	0.8443
test time + all training data	**0.7890**	**0.9067**	**0.8622**	**0.8526**

rate from set {1e-3,5e-4} with (ii)constant/polynomial decay/cosine cycle (T = 10/20) and (iii)more epochs (2)/partially batches (50% or 25%) strategies and achieve the highest performance with epoch one and polynomial learning rate decaying setting beginning at 5e-4. We add the client pruning strategy that filters out time-consuming clients in the first c rounds for faster convergence with the above basis. The value of c is chosen from set 5,10,20. Notably, the convergence score of 74.32 is achieved when we perform client pruning for the first ten rounds, indicating that rejecting time-consuming clients to participate in federated optimization can further improve the convergence speed on top of the FedNorm scheme. And the results are sensitive to the hyper-parameter c. After extensive experiments, we set it as 10.

In the end, we obtained the performance of our method on the test dataset from the organizers as Table 2 shows. The results show that the mean dice score has dropped from 72.43 to 60.35.

Ablation Study for Task 2. We train two models for task 2. One uses the fixed train and validation split in task 1, and another one drops the last 20% of

Table 4. Test results for task 2

Unseen data	Dice ↑		HD95 ↓		Sensitivity ↑		Specificity ↑	
	mean	std	mean	std	mean	std	mean	std
ET	0.7602	0.2736	56.666	110.22	0.7758	0.2806	0.9996	0.0006
WT	0.8715	0.1311	11.198	21.208	0.8814	0.1229	0.9993	0.0007
TC	0.7731	0.2393	15.693	28.515	0.8669	0.1637	0.9992	0.0008
Average	0.8016	0.2147	27.852	53.316	0.8414	0.1891	0.9994	0.0007

clients as unseen test data to simulate the domain shift problem, with the first 80% of clients' data then randomly divided into train and validation set in a ratio of 8:2. The two models are both trained using the backbone discussed in Sect. 3.1. As we can see from Table 3, the model trained with complete training data still performs better than the 80% one on the validation platform with about one percent dice scores improvement. So we finally choose the model in the second row as our submission. Finally, we add test time adaptation to the submission model and observe an approximate one-point increase for enhancing tumor and tumor core while slightly dropping for the whole tumor (Fig. 1).

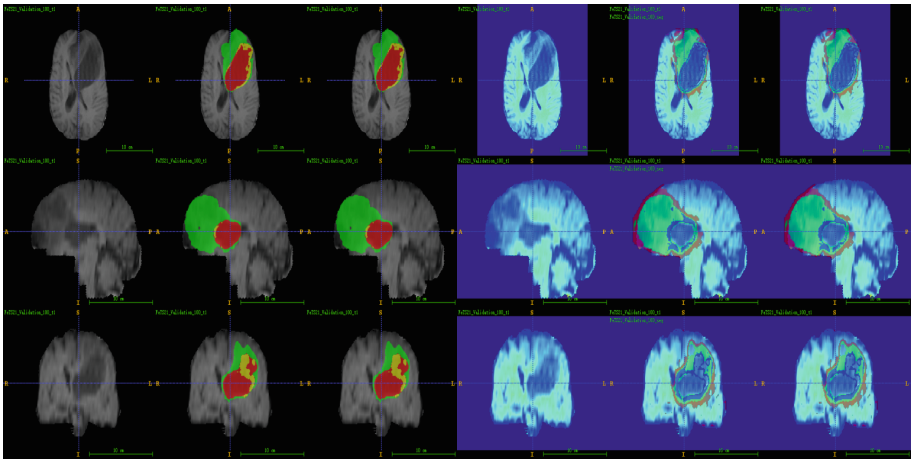

Fig. 1. Visualisation for task 2: We visualise case 100 in validation set. From left to right, the pictures are the original image, the segmentation result before adaptation and after adaptation, following by their corresponding entropy map. From top to bottom, in turn are the slices observed along z, y, and x axes.

Also, we obtained the performance on the test dataset as Table 4 shows. Compared to the sharp performance drop from training set to test set in task 1, the results show that our test time adaptation may mitigate the impact of the domain shifting problem to some degree.

Since we only had the training set and validation set without ground truth, we visualized the case in which the overall segmentation results were good-corresponding Dice is high. As we can see from the figure, the entropy map after adapting is smoother than the one without adapting. However, the effect is not significant because there is no severe domain shifting between the training set and the validation set recognized by the Challenge's organizer.

Acknowledgement. This work was supported by Key-Area Research and Development Program of Guangdong Province, China (2020B010165004), Hong Kong RGC TRS Project No. T42-409/18-R, National Natural Science Foundation of China with Project No. U1813204 and Shenzhen-HK Collaborative Development Zone.

References

1. Pati, S., et al.: Reproducibility analysis of multi-institutional paired expert annotations and radiomic features of the Ivy glioblastoma atlas project (Ivy GAP) dataset. Med. Phys. **12**, 6039–6052 (2020)
2. Menze, B.H., et al.: The multimodal brain tumor image segmentation benchmark (BRATS). IEEE Trans. Med. Imaging **34**(10), 1993–2024 (2015)
3. Bakas, S., et al.: Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge (2019)
4. Bakas, S., et al.: Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. Sci. Data **4**, 170117 (2017)
5. Isensee, F., Jäger, P.F., Full, P.M., Vollmuth, P., Maier-Hein, K.H.: nnU-Net for brain tumor segmentation. In: Crimi, A., Bakas, S. (eds.) BrainLes 2020. LNCS, vol. 12659, pp. 118–132. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72087-2_11
6. Isensee, F., Jaeger, P.F., Kohl, S.A.A., Petersen, J., Maier-Hein, K.H.: nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nat. Methods **18**(2), 203–211 (2021)
7. Annas, G.J.: HIPAA regulations - a new era of medical-record privacy? N. Engl. J. Med. **348**(15), 1486–1490 (2003)
8. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data (2017)
9. Yang, T., et al.: Applied federated learning: improving google keyboard query suggestions (2018)
10. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. (TIST) **2**, 1–19 (2019)
11. Rieke, N., et al.: The future of digital health with federated learning. NPJ Digital Med. **1**, 1–7 (2020)
12. Sheller, M.J., et al.: Federated learning in medicine: facilitating multi-institutional pruning without sharing patient data. Sci. Rep. **1**, 1–12 (2020)
13. Dou, Q., et al.: Federated deep learning for detecting COVID-19 lung abnormalities in CT: a privacy-preserving multinational validation study. NPJ Digital Med. **4**(1), 1–11 (2021)
14. Roth, H.R., et al.: Federated learning for breast density classification: a real-world implementation. In: Albarqouni, S., et al. (eds.) DART/DCL -2020. LNCS, vol. 12444, pp. 181–191. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60548-3_18

15. Li, W., et al.: Privacy-preserving federated brain tumour segmentation. In: Suk, H.-I., Liu, M., Yan, P., Lian, C. (eds.) MLMI 2019. LNCS, vol. 11861, pp. 133–141. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32692-0_16
16. Kaissis, G.A., Makowski, M.R., Rückert, D., Braren, R.F.: Secure, privacy-preserving and federated machine learning in medical imaging. Nat. Mach. Intell. **2**(6), 305–311 (2020)
17. Li, D., Kar, A., Ravikumar, N., Frangi, A.F., Fidler, S.: Federated simulation for medical imaging. In: Martel, A.L., et al. (eds.) MICCAI 2020. LNCS, vol. 12261, pp. 159–168. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59710-8_16
18. Sheller, M.J., Reina, G.A., Edwards, B., Martin, J., Bakas, S.: Multi-institutional deep learning modeling without sharing patient data: a feasibility study on brain tumor segmentation. In: Crimi, A., Bakas, S., Kuijf, H., Keyvan, F., Reyes, M., van Walsum, T. (eds.) BrainLes 2018. LNCS, vol. 11383, pp. 92–104. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11723-8_9
19. Silva, S., Gutman, B.A., Romero, E., Thompson, P.M., Altmann, A., Lorenzi, M.: Federated learning in distributed medical databases: meta-analysis of large-scale subcortical brain data. In: ISBI, pp. 270–274. IEEE (2019)
20. Pati, S., et al.: The federated tumor segmentation (FeTS) challenge (2021)
21. Guo, P., Wang, P., Zhou, J., Jiang, S., Patel, V.M.: Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning (2021)
22. Mohri, M., Sivek, G., Suresh, A.T.: Agnostic federated learning (2019)
23. Wang, J., Liu, Q., Liang, H., Joshi, G., Poor, H.V.: Tackling the objective inconsistency problem in heterogeneous federated optimization (2020)
24. Liu, Q., Chen, C., Qin, J., Dou, Q., Heng, P.-A.: FedDG: federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In: CVPR (2021)
25. Chen, C., Liu, Q., Jin, Y., Dou, Q., Heng, P.-A.: Source-free domain adaptive fundus image segmentation with denoised pseudo-labeling. In: de Bruijne, M., et al. (eds.) MICCAI 2021. LNCS, vol. 12905, pp. 225–235. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87240-3_22
26. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: fully test-time adaptation by entropy minimization (2021)
27. Liu, C., Belkin, M.: Accelerating SGD with momentum for over-parameterized learning. arXiv preprint arXiv:1810.13395 (2018)
28. Karani, N., Erdil, E., Chaitanya, K., Konukoglu, E.: Test-time adaptable neural networks for robust medical image segmentation. Med. Image Anal. **68**, 101907 (2021)
29. Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., Hardt, M.: Test-time training with self-supervision for generalization under distribution shifts (2020)
30. Reina, G.A., et al.: OpenFL: an open-source framework for federated learning (2021)
31. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-GBM collection (brats-TCGA-GBM). Cancer Imaging Archive (2017)
32. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-LGG collection (brats-TCGA-LGG). Cancer Imaging Archive (2017)
33. Liu, Q., Yang, H., Dou, Q., Heng, P.-A.: Federated semi-supervised medical image classification via inter-client relation matching. arXiv preprint arXiv:2106.08600 (2021)