MLDL24 Federated Learning Project Report: Data and System Heterogeneity Effects on Federated Learning and a Multi-Objective Approach

Riccardo Moroni
s320002

Ivan Ludvig Tereshko
s327874

Abstract

This report investigates the effects of data and system heterogeneity on model training within the context of federated learning. Data heterogeneity is introduced through non-IID data splitting, while system heterogeneity is simulated via skewed client participation. The performance of the FedAvg algorithm is evaluated across different setups. To address these challenges the federated learning problem is tackled as a multi-objective optimization problem in two ways. First, the study explores the efficacy of a loss-biased FedAvg algorithm, then, the Multi-Gradient Descent Algorithm (MGDA), proven to converge to Pareto optimality, is implemented and its performance is assessed.

1. Introduction

This is the project report of the Machine Learning and Deep Learning course we attended during the 2023/2024 academic year. Our experiments focused on evaluating, in a federated scenario and over many different setups, the learning capabilities of a CNN for image classification using the CIFAR-100 dataset, and of a RNN for next character prediction using "*The Complete Works of William Shakespeare*" dataset. With this project we had the chance to explore the research field of Federated Learnign and, in particular, to observe how various data distributions among clients and skewed client participation can negatively impact the performance of global models. The code for this project is stored in a GitHub repository [1].

Upon completing this assignment, we focused on addressing the Federated Learning problem as a Multi-Objective one, in two ways. The first more naive approach termed Loss-Biased FedAvg involved using client-side validation losses to weigh clients' updates in order to prioritize those clients for which the global model still performs worse. The second approach involved instead implementing the Multi-Gradient Descent Algorithm (MGDA) to essentially treat each client as a different task while ensuring convergence to Pareto stationary solutions.

[1] https://github.com/Riccardo-Moroni/federated_learning_mldl24

2. Related Work

FedAvg is an established communication-efficient federated learning method [6]. In [7] the performance of federated versions of adaptive optimizers is analyzed in the presence of data heterogenity.

Loss-biased FedAvg could be defined as an heuristic of the most straightforward approach to Multi Task Learning, Scalarization, which consists of minimizing a weighted sum of all the tasks' losses (with the major difference being the fact that in the federated scenario, before summing the losses, each client performs J steps) and to the best of our knowledge it has never been explored. However Scalarization has been proven to successfully converge to the Pareto set only for very simple models and under some strict conditions [4]. This led us to implement Multi-Gradient Descent Algorithm (MGDA), which is proven to converge to the Pareto set leveraging the fact that the opposite direction of the one identified by the shortest convex linear combination of the normalized gradients of the single losses, is pointing towards a solution in which all the single losses are either decreased or stay the same [2].

A multi-objective optimization approach to federated learning based on MGDA and termed FedMGDA+ was proposed in [5], it shown to be robust against malicious actors, but the authors didn't do experiments with CIFAR-100 and Shakespeare as we did.

3. Methods

3.1. Preliminaries

3.1.1 Datasets

- CIFAR-100: A dataset for the image classification task spanning 100 classes, each containing 600 images 32x32 (500 for training + 100 for testing).
- Shakespeare: A dataset for next the character predic-

tion task, it consists of pairs sentence- character where the sentence is an 80 characters long sentence and the character is the 81^{st} character of that sentence. This dataset is provided by [1].

3.1.2 Data Augmentation

- The 32x32 images in CIFAR-100 were randomly cropped to 28x28, randomly horizontally flipped and normalized.
- No data augmentation techniques were applied to the Shakespeare dataset.

3.1.3 Models

The models differ according to the used dataset.

- For CIFAR-100, a modified LeNet-5 architecture is used, described in [3].
- For the Shakespeare dataset, an RNN with two stacked LSTM layers is used, described in [7].

3.1.4 Client Participation

In the case of uniform participation, all K clients have an equal probability of being selected at each round, i.e. the probability distribution is uniform: $\mathbf{p} = (1/K, 1/K, ..., 1/K)$.

To model skewed client participation, the probability distribution $\mathbf{p}$ is sampled according to the flat K-dimensional Dirichlet distribution $D(\alpha)$. The parameter $\gamma = 1/\alpha$ is introduced, which indicates the scale of the skew. Histograms for client selection probabilities are presented in Fig.1 for different levels of skewness. At each round the $C = 0.1$ portion of clients is sampled. In the uniform case all clients are selected with equal probability C. The higher the parameter γ is, the sparser are the histograms and the more heterogeneous is the distribution.

3.1.5 Data Splitting

For CIFAR-100, the data was split among clients in two different ways:

- IID: the data label distribution is uniform among the clients
- Non-IID: each client is given data samples belonging to N_c classes. Each client has approximately the same number of samples.

For the Shakespeare dataset the data splitting followed this same logic with the only difference that the non-IID split (which is its native one) is achieved considering each client as a playing-character in one of the Shakespeare's works and is assigned with the sentences belonging to him. We randomly pick 100 clients being assigned with at least 2000 sentences from the total of 1128 playing characters.

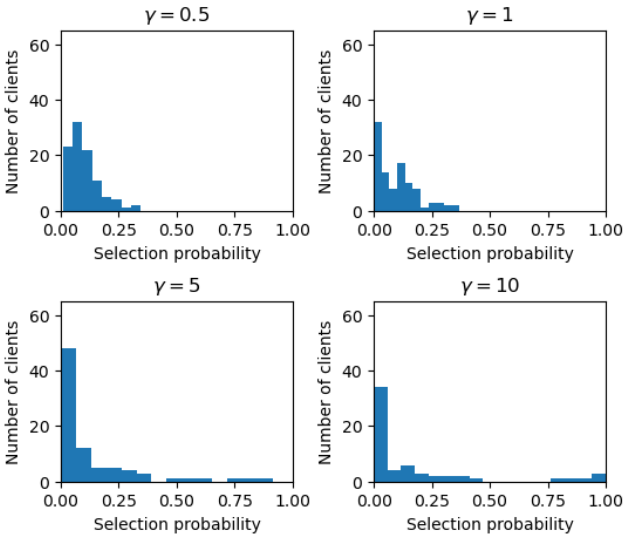

Figure 1. Client participation histograms, $C = 0.1$

3.2. Approaches

3.2.1 FedAvg

In the FedAvg algorithm [6] training happens in rounds of communication. At each round i the global model w^i is shared with randomly selected clients S_i, containing the C portion of clients. Each selected client performs J steps of SGD with learning rate η on its dataset of size n_k and sends the resulting model weights w_k to the server. The updated models are aggregated by computing the weighted average with weights being proportional to local dataset sizes and normalized:

$$w^{i+1} = \sum_{k \in S_i} \frac{n_k}{n_i} w_k^i \tag{1}$$

where $n_i = \sum_{k \in S_i} n_k$. When all clients have datasets of the same size, the aggregation expression (1) is simplified:

$$w^{i+1} = \frac{1}{|S_i|} \sum_{k \in S_i} w_k^i$$

3.2.2 Loss-biased FedAvg

The proposed algorithm is derived from FedAvg by modifying the aggregation method (1) by using weights proportional to the normalized validation loss averaged over the J steps, so the update is redefined as:

$$w^{i+1} = \sum_{k \in S_i} \frac{l_k}{l_i} w_k^i \tag{2}$$

where l_k is the averaged validation loss of client k at the i-th communication round and $l_i = \sum_{k \in S_i} l_k$. This method aims at learning more from those clients' datasets it needs more, avoiding overfitting on the others and, unlike FedAvg, it requires each client to have train and validation datasets.

3.2.3 MGDA

Multi-Gradient descent algorithm allows addressing the Federated Learning problem as a multi objective one, infact with MGDA every client is percieved as a different task and the problem translates into a Multi-Objective Minimization problem, where we try minimizing all clients' losses simoultaneously. This can be achieved by moving, in the parameters' space, in the opposite direction of the one identified by the shortest convex linear combination of the normalized gradients of all the clients.

So, defined the set of all the convex (λ-s positive and summing to 1) linear combinations of the normalized gradients u_i as:

$$\overline{U} = \left\{ u \in \mathbb{R}^N \middle| u = \sum_{i=1}^{n} \alpha_i u_i \, ; \alpha_i \geq 0 (\forall i = 1, .., n) \, ; \sum_{i=1}^{n} \alpha_i = 1 \right\} \tag{3}$$

at each communication round we compute all the clients' u_i via backpropagation and find, by a scipy routine, the shortest element of $\overline{U}$, its opposite is the direction which guarantees either descending or staying the same over all the different clients' losses. The step size is defined by a server learning rate η_s.

This has been proven [2] to allow convergence to the Pareto set, which is the set of all Pareto optimal solutions, that are not improvable for any of the objectives without sacrificing some, enforcing fairness among users.

4. Results

4.1. Centralized Baseline

4.1.1 CIFAR-100

The modified LeNet-5 model was trained with the following hyperparameters for SGD: learning rate $\eta = \{10^{-2}, 10^{-3}, 10^{-4}\}$, momentum $m = \{0, 0.9, 0.99\}$. For learning rate schedulers the linear, cosine annealing and exponential schedulers were considered.

The best performance was achieved learning rate $\eta = 10^{-3}$ and momentum $m = 0.9$ using the cosine annealing scheduler. The model achieved a test accuracy of 46.88% in 100 epochs. The plots of test accuracy and loss are presented in Fig. 2, 3.

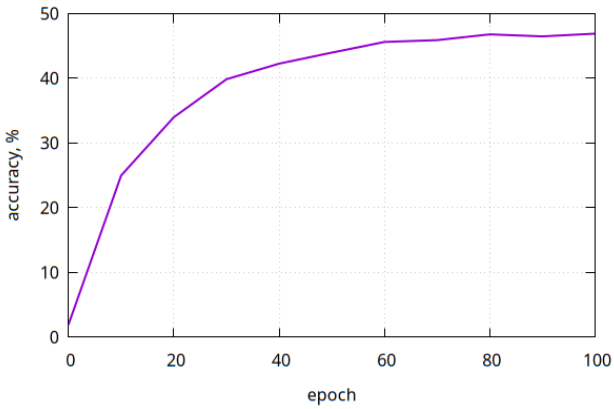

Figure 2. Test Accuracy for centralized baseline on CIFAR-100

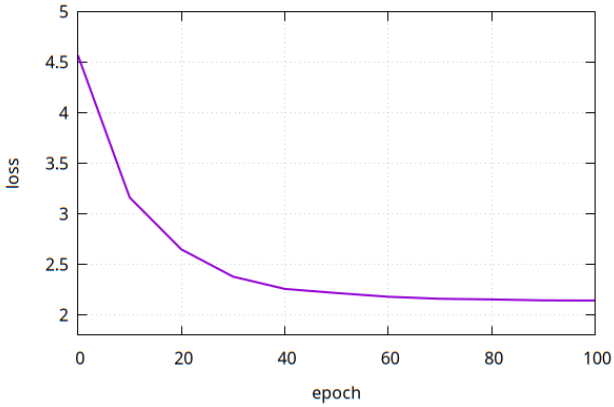

Figure 3. Test loss for centralized baseline on CIFAR-100

4.1.2 Shakespeare

The RNN model was trained with all the combinations of the following parameters: learning rate $\eta = \{0.05, 0.1, 0.5, 1\}$, batch size $B = \{50, 100, 200\}$, momentum $m = \{0, 0.9\}$, weight decay $= \{0, 4 \cdot 10^{-4}\}$. During these experiments we noticed how the use of weight decay has always led to the worst performances and, the best performing learning rate scheduler was the cosine annealing one.

The best performing experiment can be seen in Fig.4 and Fig.5 and was scored with momentum = 0.9, B = 100, learning rate $\eta = 1.0$, acheaving a test accuracy of 53.81% after 20 epochs. It is important to notice, however, that the combined use of momentum and high learning rate led to less healthy accuracy and loss curves, incentivizing the overfit. It would infact be better, in this particular configuration, stopping the training at the 14^{th} epoch when the validation loss reaches it's lowest value and the validation accuracy is also slightly higher than the final one (behaviour that we can also notice over the test accuracy and loss).

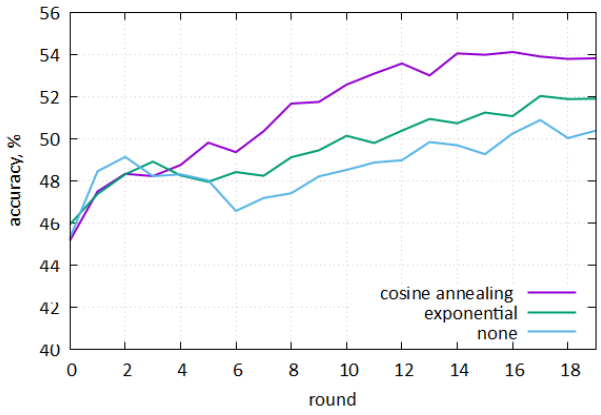

Figure 4. Comparison of test accuracies achieved by different learning rate schedulers for the centralized baseline on Shakespeare dataset. $\eta = 1.0$, $B = 100$, momentum $= 0.9$.

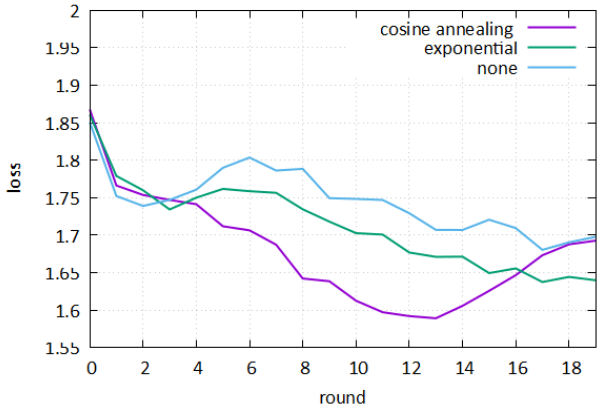

Figure 5. Comparison of test losses achieved by different learning rate schedulers for the centralized baseline on Shakespeare dataset. $\eta = 1.0$, $B = 100$, momentum $= 0.9$.

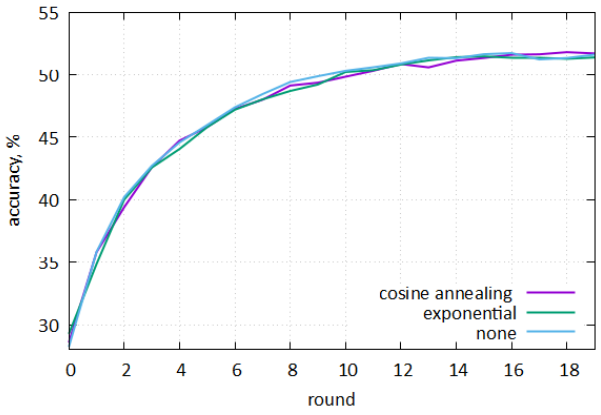

Figure 6. Comparison of test accuracies achieved by different learning rate schedulers for the centralized baseline on Shakespeare dataset. $\eta = 0.05$, $B = 10$, momentum $= 0.9$.

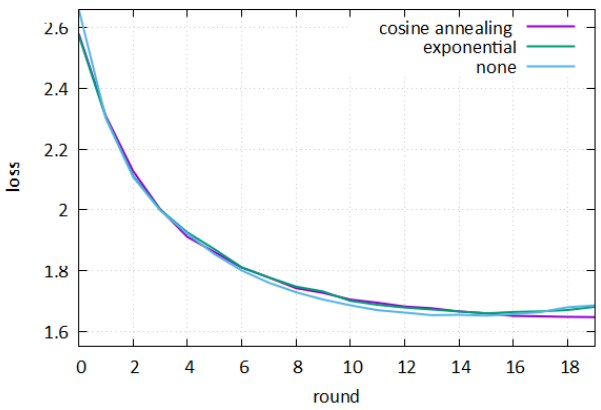

Figure 7. Comparison of test losses achieved by different learning rate schedulers for the centralized baseline on Shakespeare dataset. $\eta = 0.05$, $B = 10$, momentum $= 0.9$

Much healthier curves could be obtained by lowering η to 0.05, in this configuration the tendency to overfit was lower and the effect of using different learning rate schedulers was way less noticeable because of the different patterns in the learning rate variatons being smaller in amplitude, and thus having an overall more marginal impact in the training dynamics. The best run of these configurations achieved a test accuracy of 51.80%. Fig.6 and Fig.7

4.2. Federated

4.2.1 First Baseline

The federated learning baseline was achieved by running FedAvg on CIFAR-100 dateset distributed randomly among $K = 100$ clients. In each of the $T = 2000$ rounds of communication $C = 0.1$ portion of the clients are selected, each performing $J = 4$ local steps with learning rate $\eta = 0.1$. The model reaches a test accuracy of 43.2%. The plots presented in Fig. 8, 9 indicate that training could be stopped at 1800, when test accuracy reaches 43.8%. After this accuracy slowly deteriorates as overfitting starts.

Regarding the next character prediction task, the best performing hyperparameters configuration was found to be $\{\eta = 1.5$, weight decay=0, C=0.3, J=5$\}$, which was searched among combinations of the following parameters: $\eta = \{0.5, 1, 1{,}47, 1.5\}$, weight decay = $\{0$, 4e-4$\}$, $C = \{0.1, 0.2, 0.3\}$, $J = \{1, 5, 10\}$ and a $B = 10$.

It is worth noticing that increasing C, the fraction of clients taking part to each communication round, does not improve the performance drastically, as can be seen in Fig.10, while having great impact on the training time and being impractical in an actual federated scenario, since clients, in order to transmit their updates, need a solid connection and be available, so expecting that almost a third of all the clients respects these constraints at each communication round would be too optimistic.

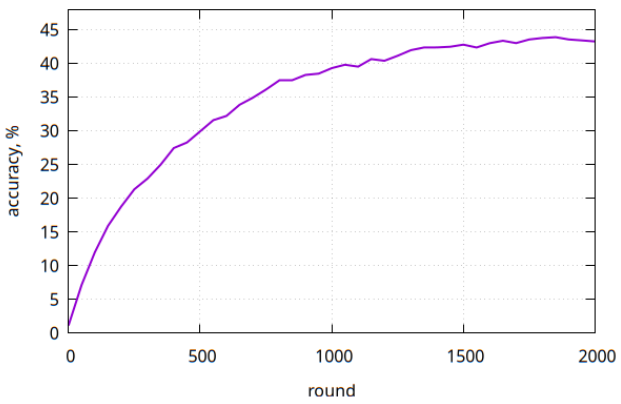

Figure 8. Test Accuracy for Federated Baseline on CIFAR-100

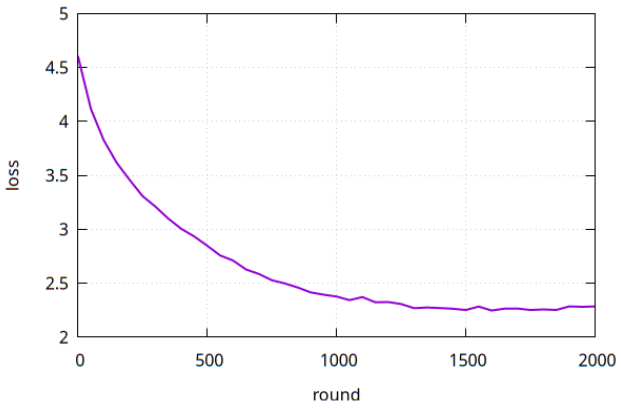

Figure 9. Test loss for federated baseline on CIFAR-100

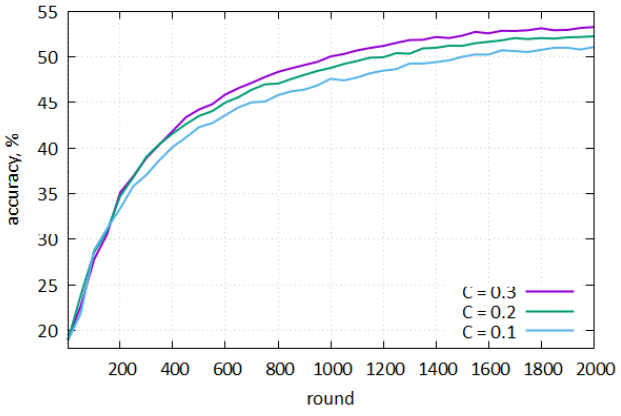

Figure 10. Federated baseline over Shakespere, comparison of differnt Cs. $\eta = 1.5$, weight decay=0, momentum=0, J=5.

4.2.2 Impact of Client Participation

The impact of client participation was tested by simulating skewed client participation using the described method in 3.1.4 for values $\gamma = \{0.5, 1, 5, 10\}$. The plot for test accuracy on the CIFAR-100 dataset is presented in Fig. 11. The behaviour is similar for all settings for the first 250 iterations, but accuracy eventually deteriorates for skewed settings. The higher skewness γ, the worse is the test performance.

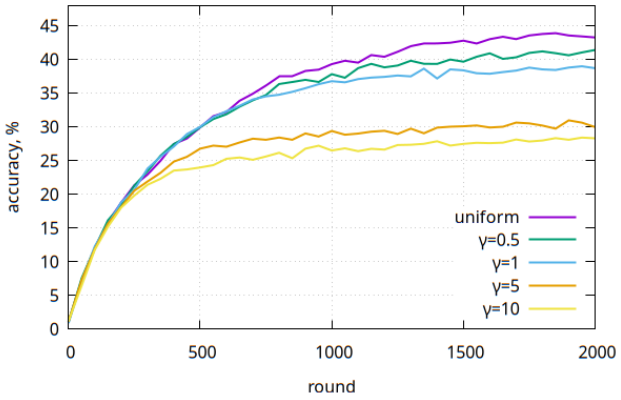

Figure 11. Test accuracy for client participation schemes on CIFAR-100

The test accuracies plots for the Shakespeare dataset are presented in Fig.12. The other parameters used for these experiments are those of the $C = 0.1$ run in Fig. 10. The best test accuracies are presented in Table 1, where the case of uniform participation corresponds to $\gamma = 0$. The performance drop with introduction of skewness is less significant for the Shakespeare dataset.

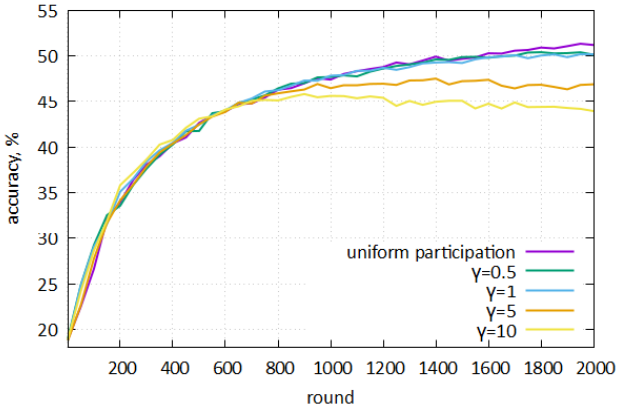

Figure 12. Test accuracy for client participation schemes on Shakespeare dataset.

Algorithm	Test Accuracy (best round) $\gamma = 0$	$\gamma = 0.5$	$\gamma = 1$	$\gamma = 5$	$\gamma = 10$
CIFAR-100	43.89	41.39	38.98	30.96	28.40
Shakespeare	51.32	50.68	50.17	47.50	46.49

Table 1. Test accuracy for different values of parameter γ

4.2.3 Heterogeneous Data Distribution

Heterogeneous data distributions were simulated by varying the number of labels clients have $N_c = \{1, 5, 10, 50\}$. The performance in such settings was compared with the IID setting for different numbers of local steps $J = \{4, 8, 16\}$. The number of communication rounds was scaled in order to keep the total number of steps constant. The plots of test accuracy are presented in Fig. 13, 14, 15.

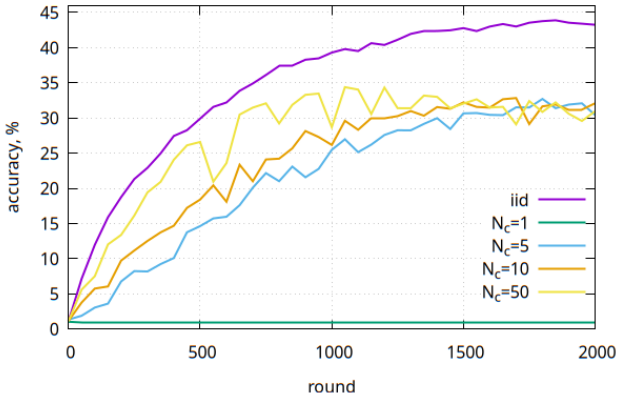

Figure 13. Test accuracy for $J = 4$ on CIFAR-100

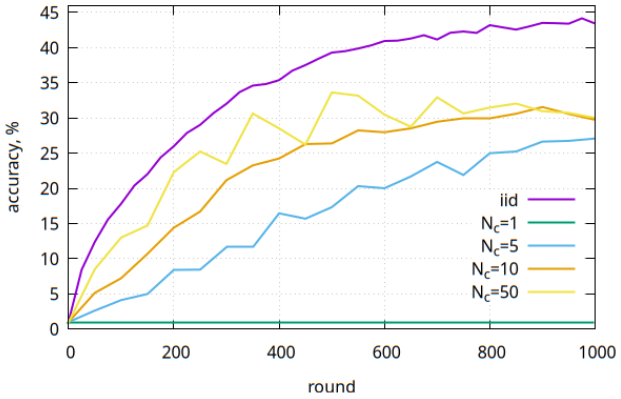

Figure 14. Test accuracy for $J = 8$ on CIFAR-100

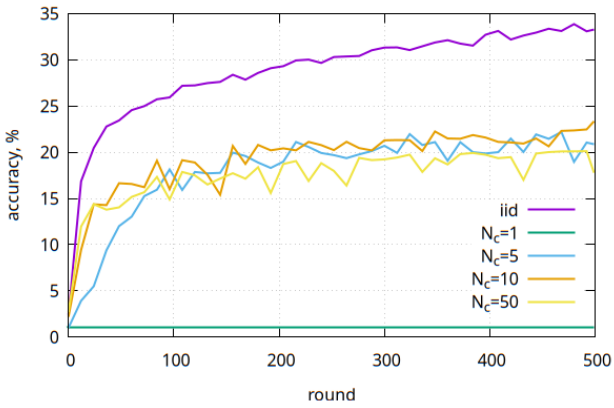

Figure 15. Test accuracy for $J = 16$ on CIFAR-100

The results regarding performance are summarised in Table 2. In general, the model performance deteriorates the more non-IID the data distribution is (lower N_c). For $N_c = 1$ the model does not train at all. For non-IID settings the performance of the global model gets worse as the the number of local steps J rises. While in the IID case the model performs best with $J = 8$.

Data Distr.	Test Accuracy (best round) $J = 4$	$J = 8$	$J = 16$
$N_c = 1$	1.00	1.00	1.00
$N_c = 5$	32.71	27.08	22.19
$N_c = 10$	32.84	31.56	23.34
$N_c = 50$	34.39	33.63	20.11
IID	43.89	44.17	33.85

Table 2. Test accuracy for different settings for CIFAR-100

For the task of next character prediction, instead, both the IID and non-IID splits were already provided (and explained in 3.1.5) and the results of using different values of J in both settings can be appreciated in Fig. 16. The curves corresponding to the same values of J are very similar, because the non-IID split of Shakespeare is far less pathological than those of CIFAR-100. This is also corroborated by the fact that repeating the previous experiment, 4.2.2, on the IID split of Shakespeare, yields almost the same results, while, if the IID split was very different from the non-IID one, we would have expected it to perform better.

4.2.4 Loss-biased FedAvg

Weighting clients' updates on their validation loss has not proven to be beneficial for the image classification task on CIFAR-100 with clients having non-IID data distributions, as can be seen in Fig. 17, but gave promising results on the non-IID split of Shakespare dataset, reaching higher performances than FedAvg over all the skewed participation pat-

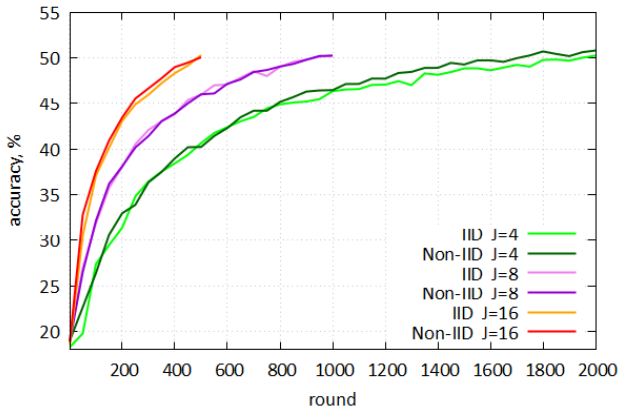

Figure 16. IID vs non-IID, different J values comparison over Shakespeare.

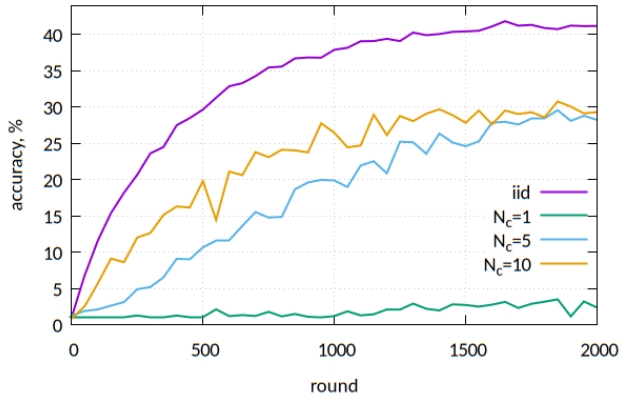

Figure 17. Test accuracy for loss-biased FedAvg on CIFAR-100

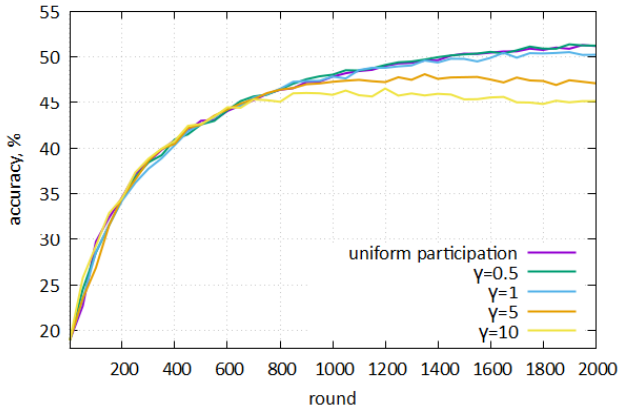

Figure 18. Loss-biased FedAvg test accuracy on the non-IID split of Shakespeare dataset.

terns. Results about the test accuracy are shown in Fig. 18 and the best rounds are compared with those of Fig. 12 in Table 3.

Algorithm	Test Accuracy (best round) $\gamma=0$	$\gamma=0.5$	$\gamma=1$	$\gamma=5$	$\gamma=10$
FedAvg	**51.32**	50.68	50.17	47.50	46.49
Loss-biased	51.28	**51.36**	**50.44**	**48.09**	**46.52**
MGDA	46.75	46.54	46.49	45.88	45.64

Table 3. Test accuracy of original FedAvg Loss-biased FedAvg and MGDA for different levels of clients' participation skewness on non-IID split of Shakespeare dataset

4.2.5 MGDA

Results obtained with MGDA over the non-IID split of the Shakespeare dataset are visible in Fig.19 and reported in Table 3 and show a slower learning, however its capability in fostering fairness among clients also in the most skewed scenarios, is supported by Fig.21 that shows how, when incrementing the batch size B to 50, the loss of FedAvg deteriorates more than MGDA's one, indicating how MGDA tends to overfit less on those clients selected more often.

Given that training MGDA is more time demanding, due to technical constraints we were not able to conduct a comprehensive analysis of configurations in order to find the best set of hyperparameters as was done for FedAvg. The hyperparameters that were explored are $\eta_s = \{1\}$, $\eta = \{1.5\}$ $B = \{10, 50, 100\}$ and C = 0.1, with no server-side weight decay and momentum.

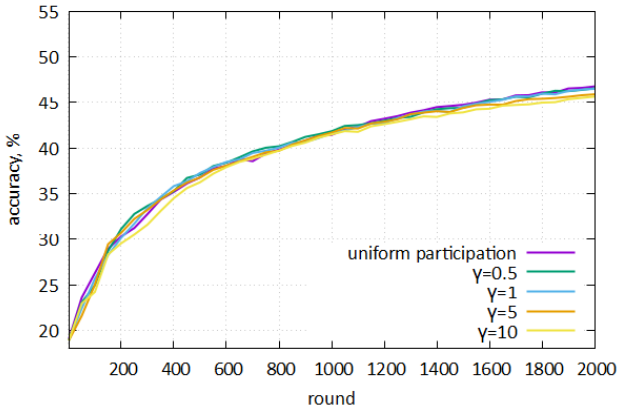

Figure 19. Test accuracy for MGDA on Shakespeare. $B = 10$ for fair comparison with Fig.12 and Fig18

MGDA was run for non-IID settings on CIFAR-100. The test accuracies plots are given in Fig. 20. The accuracies are similar to that of FedAvg and a little better for the cases with $N_c = 1$, $N_c = 5$ and IID. The comparison of performance of the three considered methods for different data distributions is presented in Table 4.

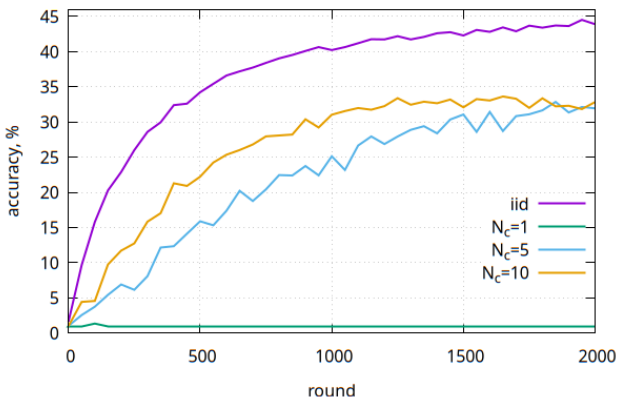

Figure 20. Test accuracy for MGDA on CIFAR-100

Method	Test Accuracy			
	IID	$N_c = 1$	$N_c = 5$	$N_c = 10$
FedAvg	43.89	1.00	32.71	**34.39**
Loss-biased	41.84	**3.21**	29.57	30.77
MGDA	**44.52**	1.35	**32.86**	33.64

Table 4. Comparison of test accuracy for different methods on CIFAR-100

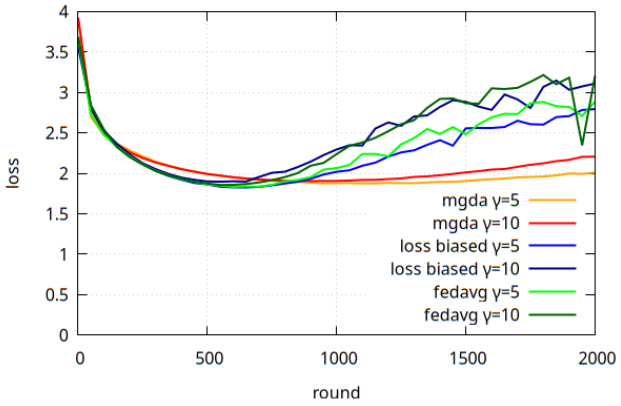

Figure 21. Losses of original FedAvg, loss-biased FedAvg and MGDA when training with $\gamma = \{5, 10\}$ using $B = 50$ over non-IID Shakespeare

5. Conclusions

The experiments show that skewed client participation leads to a performance drop, increasing as the skew rises. Increasing the heterogeneity of the data distribution leads to bigger divergence of client models, which is amplified by the number of local steps, producing a worse generalized global model.

Our loss biased approach has shown promising results in postponing overfitting in a scenario with skewed participation combined with a non-IID data distribution, outperforming FedAvg for the Shakespeare dataset.

MGDA has proven to be the least affected by different levels of clients' participation skeweness on the non-IID split of Shakespeare dataset, but a much longer training time is required due to the need of minimizing a convex linear combination at every communication round. A faster approximated version of the Multi Objective algorithm has been proposed in [8] and a federated implementation of it could be an intersting direction of further research.

References

[1] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings, 2019. 2

[2] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5):313–318, 2012. 1, 3

[3] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution, 2020. 2

[4] Yuzheng Hu, Ruicheng Xian, Qilong Wu, Qiuling Fan, Lang Yin, and Han Zhao. Revisiting scalarization in multi-task learning: A theoretical perspective, 2023. 1

[5] Zeou Hu, Kiarash Shaloudegi, Guojun Zhang, and Yaoliang Yu. Federated learning meets multi-objective optimization, 2023. 1

[6] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data, 2023. 1, 2

[7] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. Adaptive federated optimization, 2021. 1, 2

[8] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 8