Federeted Learning Overview - Report

Massimo Francios, Sergio Lampidecchia, Nicolò Bonincontro
Politecnico di Torino
{s328914,s331080,s328754}@studenti.polito.it

Abstract

Federated Learning (FL) is a distributed machine learning approach in which several clients keep their data secret and learn a model under the coordination of a central server. In this work, we propose a study on the standard federated optimization method (FedAvg) under various settings, including heterogeneous data distribution and client skewed participation. We also propose a novel method to address the problem of client participation.

1. Introduction

Federated Learning (FL) is a machine learning paradigm in which multiple clients cooperate to learn a model under the coordination of a central server. One key component of FL is that data is never shared between clients or with the server. In real world scenario two main difficulties arise: statistical heterogeneity and system heterogeneity. The first one is related to the non-identical and non-independent (non-IID) nature of data distributed among clients. System heterogeneity refers instead to the difference regarding hardware. These problems in FL require careful consideration during the implementation of this technique.

In this paper we explore standard FL scenarios, studying the effects of heterogeneity and the impact on training. Heterogeneity is studied at two levels: data and client participation. Finally we propose a novel method to tackle FL challenges about client selection. Code can be found at: https://github.com/maxfra01/federated-learning.

Main contributions.

1. We study standard FL methods such as FedAvg
2. We analyzed the effect of heterogeneity in data distribution among clients
3. We analyzed the effect of skewed participation
4. We propose a novel method to address the challenges of client selection

The paper is organized as follows. Section 2 provides details on our experimental setup, datasets, models, and sharding settings. Section 3 provides the results of our experiments for both centralized and federated models. Section 4 proposes a potential contribution for FL main challenges.

2. Methodology

In this section, we provide details on the methodology adopted in the experiments.

Datasets. We used two datasets to conduct experiments: CIFAR-100 and Shakespeare. The first one is an image dataset, the second one is composed of text. For CIFAR-100 we manually created a validation split, and functions to simulate heterogenous and homogenous data distribution among clients. Shakespeare naturally comes in two splits, using LEAF [1].

Models. For CIFAR-100 we trained a Convolutional Neural Network, based on Le-Net5 following this work [2]. For Shakespeare we trained a Recurrent Neural network for next character prediction, as described in [5].

Baseline Centralized Training. We trained the two models in a centralized setting, applying SGDM optimizer. As a learning rate scheduler we experimented with CosineAnnealingLR, StepLR and ExponentialLR. We conducted a grid search for learning rates and weight decay, fixing batch size and momentum.

Federated Training. We implemented the Federated Averaging method described in [3], where each client computes model updates and transmit it to the server. In this section we fixed the number of clients K to 100 and the fraction of clients to be selected C to 0.1. Clients have equal selection probability. We adopted an IID data sharding, meaning that each of the K clients is given an approximately equal number of training samples uniformly distributed over the class labels. We experimented with different numbers of local steps J performed by clients in each round of communication. We scaled accordingly the number of training rounds to maintain constant the computational effort.

Skew participation. In this part, clients were sampled based on a Dirichlet distribution, parametrized by γ, which indicates the "severity" of the skewness i.e. the lower γ the

higher the heterogeneity of client sampling.

Heterogeneous data. We simulated statistical heterogeneity by assigning client samples from a subset of classes. Each client receives an approximately equal number of training samples, belonging to N_c classes, We conducted experiments in which clients received samples from N_c classes ($N_c = [1, 5, 10, 50]$) and varying number of local steps J.

Implementation Details. All the code parts were implemented using Python and Pytorch. The experiments were conducted in Google Colab. We also develop a checkpointing system to handle interruptions.

3. Experiments and results

Centralized Training. We trained our CNN model on CIFAR-100 and Shakespeare using the SGDM optimizer. We trained for $E = 50$ epochs for both datasets, with a batch size $B = 64$ for CIFAR and $B = 4$ for Shakespeare. We report plots for Validation loss and accuracy in figure 1 and 2:

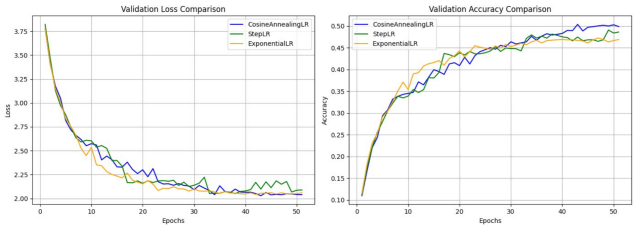

Figure 1. CIFAR-100 Validation loss and accuracy, using different learning rate schedulers.

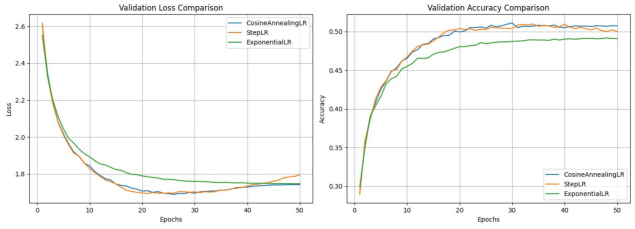

Figure 2. Shakespeare Validation loss and accuracy, using different learning rate schedulers.

In CIFAR 100 all three schedulers (CosineAnnealingLR, StepLR and ExponentialLR) show significant performance improvement during the first 30-40 epochs. The loss reduces substantially, and the accuracy improves steadily, with the learning rate stable. After 40 epochs, the model stabilizes, and the rate of improvement in loss and accuracy diminishes across all schedulers (learning has reached a convergence phase).
CosineAnnealingLR achieves the best results with the lowest validation loss and the highest accuracy. This suggests that the periodic annealing of the learning enables finer optimization in later stages, helping to avoid getting stuck in suboptimal local minima.
StepLR performs slightly worse than CosineAnnealingLR but demonstrates competitive results with stable accuracy throughout the training process.
ExponentialLR shows less competitive performance with higher validation loss and lower accuracy than the other two schedulers.
For CIFAR100, CosineAnnealingLR is the preferred choice due to its more stable learning dynamics and better final performance metrics.
Similarly to CIFAR100, the Shakespeare dataset shows notable performance improvement until the 40th epoch, when the improvements in performance metrics slow down.
CosineAnnealingLR again demonstrates superior performance, and it is the preferred choice for Shakespeare due to its more stable learning dynamics and better final performance metrics.

Federated Training. In this experiments we run FedAvg for 2000 communication rounds on CIFAR-100 and for 200 rounds on Shakespeare. Batch sizes were fixed as other experiments. Momentum was not applied. We adopted an IID data sharding for both datasets. In figure 3 and 4 we report validation losses and accuracies for both experiments, with different values of J.

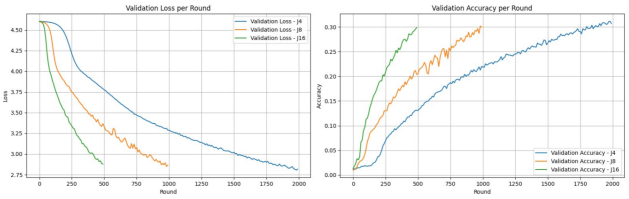

Figure 3. CIFAR-100 Validation loss and accuracy, IID scenario, varying number of local steps performed by clients

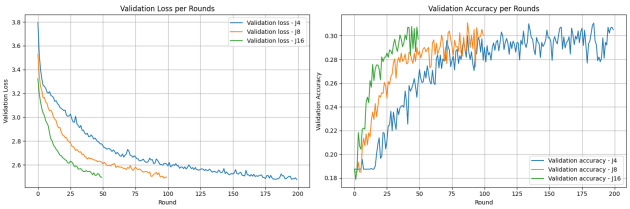

Figure 4. Shakespeare Validation loss and accuracy, IID scenario, varying number of local steps performed by clients

According to the results shown in [3], increasing client-side computation is generally expected to reduce overall accuracy due to the compounded effects of stale updates and local overfitting. While this effect is mitigated in both datasets by the homogeneous data sharding, in CIFAR-100 (Figure 3), only a slight reduction in accuracy is observed

as J increases. In contrast, for Shakespeare, this effect has an even lower impact on accuracy, primarily due to the inherently unstable learning curve.

These findings suggest that, in the context of IID data, for both CIFAR-100 and Shakespeare, local steps (J) and communication rounds can be effectively interchanged.

Skew Participation. In this set of experiments we run FedAvg on CIFAR-100 Dataset. Communication rounds were fixed to 2000, batch size to 64. Momentum was set to 0. We changes values of γ to study the effects of a skewed client participation, based on the Dirichlet distribution. In figure 5 we report plots with the clients sampling over 2000 rounds.

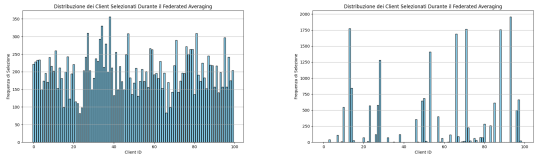

Figure 5. Client participation distribution over 2000 communication rounds, using different values of γ. In the left plot we used $\gamma = 10$, in the right one $\gamma = 0.1$

We tried different values of $\gamma = [0.001, 0.01, 0.1, 0.8, 5, 10, 50]$. We expect that both performance and convergence of accuracy decrease as increasing the severity of the skew: high severity has the effect to select repetitively the same clients, leading to a non-optimal learning process. In figure 6 we report Validation loss and accuracy, for different values of γ.

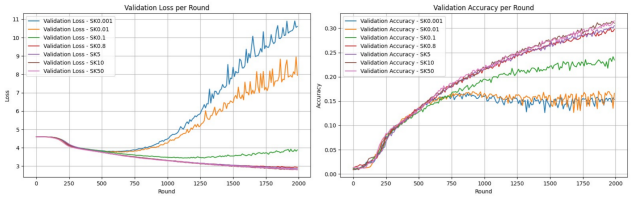

Figure 6. CIFAR-100 Validation loss and accuracy, varying the severity of skewed participation

As expected the behavior is as described: increasing the severity of the skew (decreasing γ) we get lower accuracies, as some clients dominates the model's updates. This results in a poor generalization, excessive overfitting of data and lower performance on validation set. As γ increases, participation become more uniform, leading to an improved generalization and convergence.

Heterogeonous data. In this part we run FedAvg on CIFAR-100 with our custom data sharding procedure to simulate non-IID. We also run FedAvg on Shakespeare with a pre-made non-IID data sharding from LEAF. In particular, with our CIFAR-100 setup we vary the N_c parameter that controls the number of different classes assigned to each client. J is fixed to 4. Figure 7 shows validation losses and accuracies trend for CIFAR-100, varying N_c.

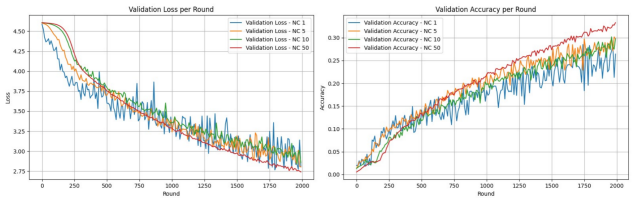

Figure 7. CIFAR-100 Validation loss and validation accuracy, varying number of classes N_c for each client with J=4

Looking at the results, it is clear that performance improves as N_c increases. When $N_c = 50$, clients have a good variety of data: each client has access to nearly half of the labels in the dataset. This allows the model to generalize better because the updates from the clients are more representative of the overall data distribution. As a result, in this scenario, we observe a faster reduction in loss and a more consistent and substantial increase in accuracy on the validation set. On the other hand, when $N_c = 1$, the situation changes dramatically. In this case, each client only has data corresponding to a single label, making the local updates of the model much less representative of the entire dataset's distribution. This leads to significantly worse performance: the loss remains unstable, and the accuracy increases more slowly, reaching lower levels compared to the other scenarios. Additionally, we can observe greater variability in the results, with noticeable oscillations in the graphs, especially in the loss. This happens because the local updates from the clients vary greatly from one another, creating instability during the aggregation process with the FedAvg algorithm. The intermediate scenarios, $N_c = 5$ and $N_c = 10$, show mixed behavior.

Finally, we conducted tests to study the effect of varying local steps in a non-IID scenario, choosing from $J = [4, 8, 16]$. Figure 8 and 9 shows results related to this last experiment.

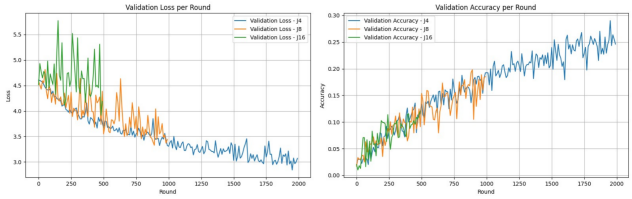

Figure 8. CIFAR-100 Validation loss and accuracy, non-IID scenario, varying number of local steps J with Nc fixed

In a non-IID environment with fixed N_c, varying the number of local steps J significantly impacts the global model's performance, as shown in Figure 8. Increasing J

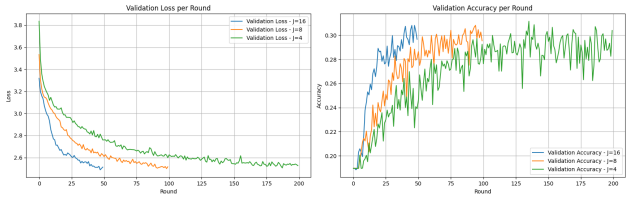

Figure 9. Shakespeare Validation loss and accuracy, non-IID scneario, varying number of local steps J

while scaling the number of communication rounds leads to performance degradation.

The performance degradation with larger J is caused by:

- *Stale Updates*: Clients perform more local updates before aggregation, which makes their updates less aligned with the current global model and the global objective.
- *Local Overfitting*: Higher values of J increase the risk of overfitting to the local data distribution, reducing the ability of the model to generalize across clients.
- *Amplified Non-IID* Effects: In a non-IID setting, larger J exacerbates the discrepancies between the local client distributions and the global distribution, leading to poorer global performance.

Thus, smaller J (e.g., $J = 4$) provides a better balance between local training and global aggregation, leading to superior global model performance.

In the context of the Shakespeare dataset, the client data is relatively similar in both IID and non-IID configurations. This is because the tasks involve completing sentences with letters, and while the writing style of different characters in the plays may vary, the underlying grammar, structures, and patterns of the English language remain consistent across all clients. These linguistic consistencies ensure that the data across clients remains largely aligned, minimizing the impact of stylistic differences. Additionally, compared to a dataset like CIFAR-100, Shakespeare has significantly fewer labels (letters instead of 100 classes), which further reduces variability and the potential for divergence in local model updates. As a result, even with an increased number of local steps (J), the model maintains good generalization performance, as the discrepancies in local data do not substantially affect the global aggregation. This highlights why performance does not degrade despite the increase in local iterations.

Effect of Varying N_c with Fixed J. Figure 10 and 11 shows that increasing N_c improves performance by reducing data heterogeneity across clients, making local updates more representative of the global distribution. However, the impact of J is also evident: higher J values amplify the effects of client heterogeneity, as clients perform more local updates before aggregation. This leads to slower convergence and less generalization for smaller N_c values, while higher N_c mitigates this issue by providing more diverse local datasets.

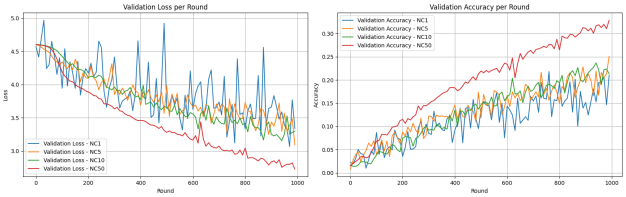

Figure 10. CIFAR-100 Validation loss and accuracy, varying number of classes N_c for each client with J=8

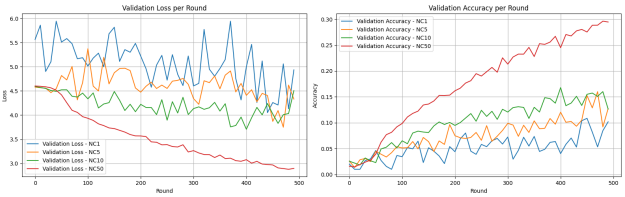

Figure 11. CIFAR-100 Validation loss and validation accuracy, varying number of classes N_c for each client with J=16

4. Personal contribution

As discussed in Section 1, one of the main problems of real-world scenarios is the heterogeneity of data among clients. Data distribution differs by number of samples and classes per clients. In our previous experiments we used a non-IID data sharding that assign approximately the same number of samples to each client, sampled from N_c labels. In this section we implemented a different strategy to perform non-IID data sharding at two levels: number of samples per client and class labels per client.

Given this background, the main challenge in FL that we want to tackle is client selection in heterogeneous scenario. As starting point we review the information in [4]. Random selection employed in previous experiment does not take in account factors like number of samples or label distribution.

Entropy. The entropy of a client in FL represents a quantitative measure of the uncertainty or diversity within the client's local data distribution. Higher entropy values indicates more balanced and diverse data within the client. Lower values suggest less diverse data distribution. For the i-th client we can calculate its entropy with the following expression:

$$H_i = -\sum_{c=1}^{C} p_c log(p_c) \tag{1}$$

- H_i represents the entropy of client i, measuring the diversity of its local data distribution.
- C is the total number of classes.
- p_c is the proportion of samples belonging to class c within the client's dataset.

Data size. A second factor taken into account is the size of the data set of each client. Clients with a larger number of samples can provide more robust updates during training, whereas those with fewer samples may contribute less representative and noisier updates.

Proposed client selection. We implemented a weighted selection mechanism that considers the two factors described above. Specifically, the weight of each client in the selection process is calculated as a linear combination of normalized entropy and dataset size, balanced by a parameter α:

$$P_i = \alpha \cdot \frac{H_i}{\sum_j H_j} + (1 - \alpha) \cdot \frac{D_i}{\sum_j D_j} \tag{2}$$

where D_i represents the size of the dataset of client i.

We varied the parameter α to study the effect of diversity and dataset size on the convergence and performance of the global model. Specifically:

- $\alpha = 0.2$: This configuration prioritizes clients with larger datasets. It achieves the best performance in terms of accuracy, as it stabilizes updates and reduces noise, although it may limit generalization in highly heterogeneous scenarios.
- $\alpha = 0.8$: The selection is strongly influenced by data diversity (entropy). This improves the generalization of the global model and outperforms the baseline, but the accuracy is slightly lower compared to $\alpha = 0.2$ due to the instability introduced by smaller datasets.
- $\alpha = 0.5$: This value balances both factors, leading to stable convergence and better generalization compared to the baseline. Although it does not reach the accuracy of $\alpha = 0.2$, it remains competitive and surpasses random selection.

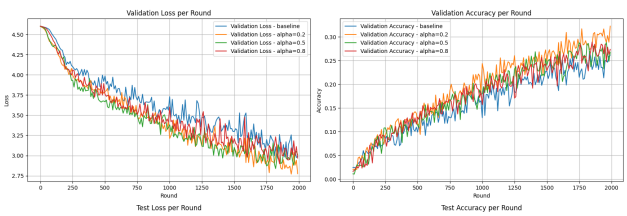

Figure 12. Our contribution compared with the baseline, CIFAR-100, Validation loss and accuracy

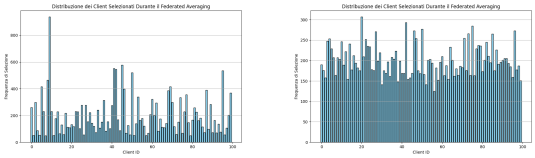

Figure 13. Client selection in our contribution (to the left $\alpha = 0.2$, to the right $\alpha = 0.8$)

Results. The results, shown in Figure 12, highlight the impact of varying the parameter α on accuracy and loss for validation datasets. The main observations are as follows:

- $\alpha = 0.2$ achieves the highest accuracy, benefiting from stable updates derived from clients with larger datasets.
- $\alpha = 0.8$ and $\alpha = 0.5$ both perform slightly better than the baseline, with $\alpha = 0.8$ emphasizing data diversity and $\alpha = 0.5$ balancing diversity and dataset size. However, their accuracy remains slightly lower than that achieved with $\alpha = 0.2$.

While all configurations outperform the baseline, the differences in accuracy are relatively small. These results demonstrate that the proposed client selection strategy can adapt to various scenarios, with $\alpha = 0.2$ being the most effective in the tested setup.

5. Conclusions

In this work, we analyzed the performance of the Federated Averaging (FedAvg) algorithm on two datasets, CIFAR-100 and Shakespeare, under various settings to study the effects of client heterogeneity, local steps, and the proposed client selection strategy.

For CIFAR-100, our experiments demonstrated that:

- Increasing the number of classes per client (N_c) reduces the effects of non-IID data, leading to improved generalization and faster convergence.
- Reducing the number of local steps (J) mitigates overfitting and stale updates, resulting in smoother convergence and better global performance.
- The proposed client selection strategy, which combines entropy and dataset size, outperforms the baseline. With $\alpha = 0.2$, the highest accuracy is achieved by prioritizing clients with larger datasets, while $\alpha = 0.8$ and $\alpha = 0.5$ balance diversity and size to achieve slightly lower, but still superior, performance.

For Shakespeare, our findings show:

- The effects of heterogeneity in data and client participation are less pronounced compared to CIFAR-100. This is due to the consistent grammatical structure and linguistic patterns across clients, which reduce the impact of non-IID configurations despite stylistic differences in the texts.
- Increasing the number of local steps (J) does not degrade performance, as the similarity of the sentence completion task across clients minimizes divergence in local updates.

References

[1] Sebastian Caldas, Jakub Konecny, H. Brendan McMahan, and Ameet Talwalkar. Leaf: A benchmark for federated settings. In *Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019. 1

[2] Tiffany Hsu, Hang Qi, Matthew Brown, Sharat Guha, and Brian Lopour. Federated visual classification with real-world data distribution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1

[3] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. pages 1273–1282, 2017. 1, 2

[4] Vincent T Nemeth, Mahmoud Abdou, Ahmed Elkordy, and Matthew Gill. A snapshot of the frontiers of client selection in federated learning. *Transactions on Machine Learning Research (TMLR)*, 2022. 4

[5] Sashank J Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konecný, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations (ICLR)*, 2021. 1