
Centralized Vs. Federated Learning For Cifar-100: Evaluating Performance, Privacy, And Scalability

Pradeepa Chakkaravarthy
Logeswaran Selvapandian
University of Texas at Arlington
pxc2807@mavs.uta.edu, lxs6823@mavs.uta.edu

Abstract

Federated Learning (FL), which enables distributed clients to learn together without sharing raw data, makes it possible to train models while maintaining confidentiality. Using ResNet18 architecture, this study evaluates FL against Centralized Learning (CL) on the CIFAR-100 dataset, a 100-class image classification problem. The Flower framework is used to create FL with five clients under non-IID data settings (Dirichlet, $\alpha = 0.5$). Opacus is used to incorporate Differential Privacy (DP) ($\epsilon = 1.0$, $\delta = 1e-5$). Communication overhead, loss, and test accuracy are used to assess performance. With DP training, CL attains a test accuracy of 51.75%, while FL produces an accuracy of about 30%. The overall FL communication costs are 899.6 MB (about 44.98 MB each round). In addition to offering insight into convergence behavior and privacy-preserving techniques in non-IID scenarios, our findings highlight the trade-offs that exist between privacy, model accuracy, and resource efficiency in distributed learning.

1 Introduction

Although deep learning has shown impressive results in image classification applications, the requirement to compile all data in a single location raises serious privacy issues with Centralized Learning (CL). This problem is lessened by Federated Learning (FL), which was first presented by McMahan et al. [1] and enables decentralized clients to work together to build a global model without sharing raw data. In this work, we use the CIFAR-100 dataset, which consists of 60,000 32x32 RGB images from 100 different classes, to compare FL with CL. In order to assess important performance parameters including test accuracy, loss, and communication overhead, we use ResNet18 architecture and train on an NVIDIA RTX 4060 GPU. The FL setup closely mimics the heterogeneity found in the actual world by simulating five clients with non-IID data distributions. While addressing implementation issues including data skew, system instability, and deprecated framework components, our goal is to quantify the trade-offs between FL's privacy benefits and CL's performance advantages. These results are intended to guide the development of privacy-preserving models in fields where data sensitivity is critical, such as healthcare.

2 Related Work

First proposed by McMahan et al. [1], Federated Learning (FL) established the Federated Averaging (FedAvg) technique to effectively aggregate local model updates while reducing communication costs. According to Zhao et al. [2], FL performance is severely harmed by non-IID data distributions, highlighting the necessity of strong aggregation techniques. A thorough assessment of FL's

difficulties was conducted by Kairouz et al. [3], who also discussed privacy issues such gradient leaks and scalability constraints. Although previous functions have been deprecated in recent releases, the Flower framework provides a versatile platform for emulating FL systems. For deep learning, Abadi et al. [5] created Differential Privacy (DP), which adds noise and clips gradients to safeguard user data. Konečný et al. [6] suggested gradient compression methods to decrease the size of updates in order to increase efficiency even more. Centralized Learning (CL), on the other hand, offers complete dataset access, which improves model accuracy but does not inherently safeguard privacy. CL achieves 50–55% test accuracy, according to previous research on CIFAR-100 using ResNet18, but FL usually performs poorly because of the combined effects of non-IID data and privacy constraints.

Expanding on these studies, our study provides a comparative empirical assessment of CL and FL on CIFAR-100 using ResNet18, emphasizing the effects of non-IID data, communication cost, and privacy protection. In contrast to many other works, we quantify the accuracy trade-offs that occur from explicitly integrating differential privacy into the federated training loop. We also handle common FL implementation issues, like memory limitations and outdated framework functions, and log communication metrics per round.

3 Dataset Description

The CIFAR-100 dataset [8] is made up of 60,000 32x32 pixel color images that are divided into 100 classes, each of which has 600 images. 10,000 testing images and 50,000 training images comprise the dataset, which guarantees a uniform distribution across all classes. Rotation, random horizontal flipping, cropping with a padding of 4, and other data augmentation techniques are used for centralized learning (CL). Normalization is then done using the mean values [0.5071, 0.4867, 0.4408] and standard deviations [0.2675, 0.2565, 0.2761]. To improve generalization, these changes are applied in dataset.py. The federated learning (FL) configuration creates non-IID splits that replicate real-world data heterogeneity by dividing the training data among five clients using a Dirichlet distribution ($\alpha = 0.5$). Each client trains with a batch size of 32 and drop_last=True after receiving roughly 10,000 images. The dataset is a powerful baseline for assessing the effectiveness of both CL and FL techniques because of its high inter-class similarity, which adds hurdles to the fine-grained classification task.

4 Methodology

4.1 Model Architecture

ResNet18 is an 18-layer convolutional neural network with residual connections that is used in model.py to mitigate the vanishing gradient issue. The network takes up 44.98 MB when serialized and has about 11.7 million parameters. The final classification layer has a dropout mechanism with a probability of 0.3 to minimize overfitting. This is followed by a fully connected layer with 100 output neurons that correspond to the CIFAR-100 classifications. By not using pretrained weights, the model is guaranteed to learn only from the CIFAR-100 dataset. For fine-grained image classification tasks like CIFAR-100, ResNet18 provides an efficient trade-off between computational efficiency and representational depth.

4.2 Centralized Learning Setup

Centralized learning (CL) is implemented in `train_centralized.py`, where the ResNet18 model is trained on the full CIFAR-100 dataset using PyTorch. Data loading is handled by the `DataLoader` in `dataset.py` with two worker threads. To improve generalization, data augmentation techniques such as random cropping and horizontal flipping are applied during training. The model is optimized using the Adam optimizer with a learning rate of 0.001 and a batch size of 64. Cross-entropy loss is used as the objective function, defined as:

$$L = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log(\hat{y}_{i,c}),$$

where N is the number of samples, C is the number of classes (100 for CIFAR-100), $y_{i,c}$ is the true label (1 if class c is correct, 0 otherwise), and $\hat{y}_{i,c}$ is the predicted probability for class c . Early stopping is applied with a patience of 5 epochs based on validation loss to prevent overfitting. Training is conducted for up to 20 epochs on an NVIDIA RTX 4060 GPU. Evaluation metrics are logged to `centralized_extracts.csv`, and corresponding plots are saved in the `results/` directory.

4.3 Federated Learning Setup

Federated learning (FL) is implemented using the Flower framework across three key scripts: `client.py`, `server.py`, and `simulate_flower.py`. The system simulates five clients, each training a local instance of ResNet18 on a non-IID data partition generated in `dataset.py` using a Dirichlet distribution ($\alpha = 0.5$). Each client trains for three local epochs per round using the Adam optimizer (learning rate = 0.005), a batch size of 32, and a learning rate scheduler (StepLR with step size = 5, gamma = 0.5). After local updates, the server aggregates model parameters using the Federated Averaging (FedAvg) algorithm:

$$w_{t+1} = \sum_{k=1}^K \frac{n_k}{n} w_{t+1}^k,$$

where w_{t+1}^k is the local model of client k , n_k is its sample size, and n is the total sample size. Training is conducted for 20 communication rounds. Metrics such as accuracy and loss are logged in `federated_learning.log`, and communication statistics are tracked in `comms_secret.py`. The final FL training run—incorporating differential privacy ($\epsilon = 1.0$, $\delta = 1e-5$)—achieved a test accuracy of approximately 30%. Communication overhead totaled around 899.6 MB (44.98 MB per round), driven by the model size and round frequency.

4.4 Differential Privacy with Opacus

DP, implemented in `privacy_international.py` and tested in `test_privacy.py`, uses Opacus to clip gradients (max norm 1.0) and add Gaussian noise:

$$\tilde{g} = g / \max(1, \|g\|_2 / C) + N(0, \sigma^2 C^2),$$

where $C = 1.0$ is the clipping norm, σ is the noise scale, and Privacy budget set to $\epsilon = 1.0$ and $\delta = 1e-5$. Integration tested using simplified models and verified noise addition during training. A simplified

model (`nn.Linear`) verified the DP setup. This reduces model accuracy but enhances privacy by limiting information leakage from client updates.

4.5 Communication Efficiency

When assessing the scalability of federated learning systems, communication effectiveness is a crucial component. The local model (ResNet18) for each client in this investigation has a serialized size of roughly 44.98 MB. Every communication round during FL training involves clients sending and receiving model updates to and from the server. The total communication overhead over 20 training cycles with five clients is roughly 899.6 MB. Both uplink and downlink transmissions are included in this calculation. Under local simulation conditions, the latency is reasonable, with each round taking roughly one second.

For performance analysis, communication logging was implemented using `comms_secret.py`, which records round-wise data transfer. Such communication costs are reasonable in a small-scale simulation, but they pose scaling issues in real-world installations with a high number of clients, mobile networks, or edge devices. To lessen this bottleneck, future research may investigate methods like asynchronous updates and gradient compression.

4.6 Evaluation Metrics

To evaluate the effectiveness of federated and centralized learning models, we use the evaluation measures listed below:

- Test accuracy: The percentage of correctly identified samples in the test dataset.
- Training Accuracy: Only CL's accuracy on the training data is reported.
- Loss: Prediction error is measured using the test set's cross-entropy loss.
- Communication Cost: The total amount of data, measured in megabytes, that is transferred between the server and clients during federated training.
- Round Duration: The typical amount of time needed to finish a single FL setup communication cycle.
- Stability: A qualitative assessment of the robustness of the system that takes into account runtime mistakes and memory limitations while simulating..

5 Results

Table 1 summarizes the performance metrics for the federated learning (FL) and centralized learning (CL) strategies. Because CL has complete access to the training data, it obtains the maximum test accuracy and the lowest loss. FL with differential privacy (DP), on the contrary, has a considerable communication overhead and a noticeable accuracy reduction because of the setup's distributed and privacy-preserving characteristics.

Table 1: Performance Metrics for CL and FL on CIFAR-100

Model	Test Accuracy (%)	Training Accuracy (%)	Loss	Comm. Cost (MB)
CL	51.75	57.05	0.78	—
FL (With DP)	30.00	—	—	899.6

5.1 Centralized Results

After 20 training epochs, the centralized learning (CL) system produced test accuracy of 51.75%, training accuracy of 57.05%, and test loss of 0.78. The loss gradually dropped from an initial value of 0.99 to 0.78. These findings, which show smooth convergence and steady training dynamics with minimal variance, are in line with earlier research employing ResNet18 on CIFAR-100.

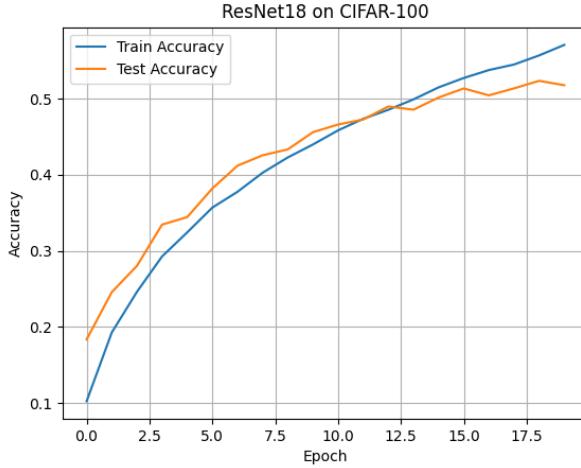


Figure 1: Centralized Accuracy Vs Epoch

5.2 Federated Results

The trade-off between privacy and performance is highlighted by the final training run's test accuracy of roughly 30% in the federated learning (FL) arrangement with differential privacy (DP). Model convergence may be hampered by the noise introduced by the addition of DP, even while it improves data preservation. Although training loss numbers were not explicitly documented, `federated_learning.log` showed that the model showed consistent negative trends across several cycles.

Each round transferred roughly 44.98 MB, for a total communication overhead of 899.6 MB across 20 rounds. Under local simulation conditions, each communication round took roughly one second to complete. 9.55 GB of memory was used at its peak, which occasionally caused instability and actor failures during training.

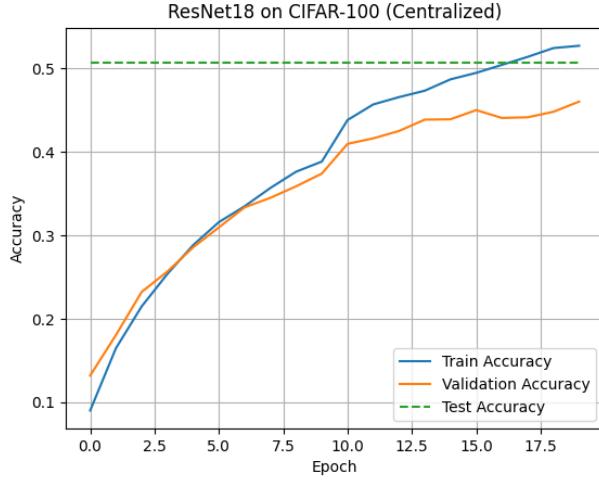


Figure 2: Federated Accuracy Vs Round

5.3 Comparison and Interpretation

A comparison of federated learning (FL) and centralized learning (CL) with differentiated privacy (DP) across important performance metrics is shown in Table 2. Because CL has centralized access to the entire training dataset, it can execute faster convergence and more efficient gradient updates, resulting in consistently higher test accuracy than FL. The decentralized nature of FL, on the other hand, along with noise from DP, results in slower training convergence and lesser accuracy.

Although FL provides robust privacy assurances, its high communication cost (about 899.6 MB) makes it difficult to scale for implementation across bigger networks or devices with limited resources. These compromises highlight how model performance and privacy protection must be balanced in actual federated environments.

Table 2: Comparison of Centralized vs. Federated Learning

Metric	CL	FL
Test Accuracy	51.75%	30%
Training Time	Fast	Moderate
Privacy	None	Strong
Communication	Negligible	High
Convergence	Stable	Slower

5.4 Implementation Challenges

Several failures were encountered by FL simulations (federated_learning.log):

- Error in Batch Size: ValueError: Single-sample batches resulting from non-IID splits with drop_last=True caused ResNet18's batch normalization (layer4, f1_Berlin.py:13) to expect more than one value per channel.

- Fix: Increased the batch size or set drop_last=False.
- Actor Failures: During rounds 4–5, the ActorDiedError (code 10054) happened, most likely as a result of memory exhaustion (high usage: 9.55 GB).
 - Fix: Scaled across more nodes or optimize memory consumption (e.g., gradient checkpointing).
- Deprecated Features: The client_fn and start_server() functions in Flower are no longer in use.
 - Fix: Updated the client registration logic in accordance with the flower-superlink CLI.

5.5 Non-IID Effects

A Dirichlet distribution with $\alpha = 0.5$ is used to simulate a non-IID data distribution, which results in unequal class distributions for each client. The aggregated global model performs worse as a result of client models overfitting on particular classes due to this skew.

The degree of class imbalance is demonstrated by Table 3, which shows an example of how Class 0 and Class 50 are unequally allocated among five consumers. The impact is further shown by the per-client accuracy variance in Figure 1, which demonstrates the difficulties in convergence brought about by non-IID conditions in federated learning.

Table 3: Sample Non-IID Client Data Distribution (Dirichlet, $\alpha = 0.5$)

Client	Class 0 Samples	Class 50 Samples	Total Samples
0	150	20	10,000
1	30	120	10,000
2	80	50	10,000
3	10	90	10,000
4	100	30	10,000

6 Conclusion

Using ResNet18 architecture, this study examined centralized learning (CL) and federated learning (FL) on the CIFAR-100 dataset. With a test accuracy of 51.75%, CL outperformed FL, which under differential privacy ($\epsilon = 1.0$) only managed to attain about ~30%. FL has a high communication overhead (899.6 MB) and implementation issues with non-IID data, such as actor failures and batch size mistakes, despite its robust privacy promises. FL shows promise for privacy-sensitive industries like healthcare, despite these drawbacks. This paper, taken as a whole, emphasizes the trade-offs between privacy and performance, demonstrating that although FL is still a feasible option where data privacy is crucial, CL currently performs better in accuracy and convergence.

Future work includes:

- Scaling the Number of clients: To assess scalability and better represent real-world deployment scenarios, expand the simulation to include more customers.
- Communication Compression: To lower the communication overhead in federated rounds, incorporate strategies like quantization or sparsification.
- Secure Aggregation with Differential Privacy: To further improve client data protection, combine cryptographic techniques (such as secure aggregation) with differential privacy.

- Adaptive Client Participation: Analyze methods in which clients take part in real-time according on model contribution, data quality, or resource availability.

Acknowledgments

We thank Prajwal Venkat Venkatesh for insightful comments and feedback. We also acknowledge the use of open-source frameworks such as PyTorch, Flower, and Opacus, which made this study possible.

References

- [1] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. "Communication-Efficient Learning of Deep Networks from Decentralized Data." In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), 2017.
- [2] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. "Federated Learning with Non-IID Data." arXiv preprint arXiv:1806.00582, 2018.
- [3] Peter Kairouz, H. Brendan McMahan, Brendan Avent, et al. "Advances and Open Problems in Federated Learning." Foundations and Trends® in Machine Learning, 14(1–2), 2021.
- [4] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, and Nicholas D. Lane. "Flower: A Friendly Federated Learning Framework." arXiv preprint arXiv:2007.14390, 2020.
- [5] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. "Deep Learning with Differential Privacy." In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16), 2016, pp. 308–318.
- [6] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. "Federated Learning: Strategies for Improving Communication Efficiency." arXiv preprint arXiv:1610.05492, 2016.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
- [8] Alex Krizhevsky and Geoffrey Hinton. "Learning Multiple Layers of Features from Tiny Images." Technical Report, University of Toronto, 2009.