MLDL project report - Federated Learning and Distributed Training

Mohamed Mostafa Kotb Hatem
s346906

Emin Oral
s343124

Sergio Petrone
s346070

Alican Akkiprik
s349610

Abstract

In an era where Machine Learning models demand increasingly larger datasets and concerns regarding data privacy are growing, Federated Learning (FL) emerges as a crucial paradigm for distributed training. By enabling multiple clients to collaboratively learn a global model without sensitive local data leaving their devices, FL offers an innovative solution to challenges related to privacy, security, and data decentralization. This approach represents a key frontier in Distributed Training, facilitating the development of intelligent systems that operate at scale with efficiency and data protection.

This work presents a detailed framework for applying Federated Learning to image classification on complex datasets such as CIFAR-100. The methodology employs Vision Transformer (ViT) models, specifically the vit_small_patch16_224 architecture, pre-trained via DINO (Self-distillation with NO labels) to ensure robust feature extraction. Our pipeline supports fine-tuning of the model's linear head directly on decentralized data, simulating scenarios of statistical heterogeneity (IID and non-IID) among clients. To address computational efficiency and scalability challenges in distributed environments, advanced pruning strategies (based on Fisher sensitivity, magnitude, and a hybrid approach) are integrated and evaluated to introduce sparsity into local models. Local training on each client is further optimized by the use of SparseSGDM, a specialized variant designed to efficiently manage gradients of sparse models. Model aggregation among clients is handled by the Federated Averaging (FedAvg) algorithm.

Our work includes a comparative analysis between the proposed FL approach and an equivalent centralized training setup. The effectiveness of ViT fine-tuning is attested by an accuracy of 45.1% on CIFAR-100 in the centralized setting. In summary, our framework provides concrete contributions to the optimization of federated training for computer vision, demonstrating how sparsity techniques can significantly improve the efficiency and robustness of models in decentralized data scenarios.

The code is available at `https://github.com/Mohamed-Hatem346906/FL-Project`

1. Introduction

The significant increase in edge devices and the growing demand for privacy-preserving applications have notably boosted interest in Federated Learning (FL). This distributed learning paradigm allows local clients to jointly train a global model without ever disclosing their raw data. Traditionally, FL research has mainly focused on convolutional neural networks (CNNs), largely due to their computational efficiency. However, recent advancements in Transformer architectures have demonstrated superior performance across various computer vision tasks, thus compelling their integration into FL workflows.

In this study, a Vision Transformer (ViT-small, with a patch size of 16) was employed as the backbone model and applied to the CIFAR-100 dataset, which comprises 100 fine-grained image categories. The investigation explored three primary axes:

- Centralized training: Used as a baseline for performance evaluation.
- Classical Federated Learning (FL): Implemented using the FedAvg algorithm.
- Sparse training: Adopted to improve overall efficiency.

Furthermore, the impact of mask-based pruning techniques on model performance was thoroughly examined, and the similarities between masks generated by different clients under non-IID data distributions were analyzed.

The contributions of this study are summarized as follows:

1. A robust pipeline for training ViTs in both centralized and federated settings has been provided.
2. Structured sparsity has been integrated through mask-based pruning methods, including Fisher sensitivity, magnitude, and hybrid schemes.
3. The inter-client pruning mask similarity has been analyzed as a proxy for collaborative coherence in FL scenarios.

2. Related work

Federated Learning (FL) has emerged as a promising solution for data privacy and communication efficiency. The Federated Averaging (FedAvg) algorithm, introduced by McMahan et al. [1], is a foundational method that allows local clients to collaboratively train deep neural networks without sharing their raw data. Since its inception, FedAvg has inspired extensive research focused on improving aggregation strategies, managing data heterogeneity, and reducing communication overhead.

The integration of Transformer architectures into FL is a relatively new area of research. Originally developed for natural language processing, Transformers—especially the Vision Transformer (ViT) pioneered by Dosovitskiy et al. [2]—have shown exceptional performance in various image classification tasks. Unlike CNNs, ViTs use self-attention mechanisms to model long-range dependencies, offering a more comprehensive view of the input. However, their application in FL is complex due to their higher computational requirements and sensitivity to data imbalance, particularly with non-IID data.

This project aims to bridge this gap by systematically adapting ViT architectures for federated scenarios. Our approach begins by establishing a centralized training pipeline to serve as a performance benchmark. We then implement the FedAvg algorithm to simulate distributed training across multiple clients with varying degrees of data heterogeneity. Through controlled experiments, we analyze how parameters like the number of classes per client (NC) and the number of local epochs influence the global model's convergence and generalization.

Beyond challenges related to data distribution, FL faces constraints from computational and memory limitations. To address these, we incorporate sparse training techniques into our framework. Sparse training seeks to prune less critical parameters during the learning process, thereby reducing model complexity without significantly sacrificing accuracy. We explore several pruning methods, including those based on Fisher sensitivity [3], magnitude-based strategies, and hybrid techniques that combine different heuristics.

Our work also builds on previous research on the consistency and transferability of pruning strategies in distributed environments. We introduce a mask overlap analysis, which involves comparing the pruning masks independently generated by each client to gauge their similarity. This metric provides insight into how well clients are aligned in identifying important model parameters, with implications for collaborative training and model compression.

In summary, our project combines three contemporary research areas: (1) ViT-based image classification, (2) Federated Learning under realistic data conditions, and (3) structured model sparsification. By bringing these domains together, we offer an empirical groundwork for future studies exploring the viability of deploying Transformer models in environments that prioritize privacy and have limited resources.

3. Method

3.1. Dataset and Preprocessing

The experimental framework is built upon the CIFAR-100 dataset, a widely used benchmark in image classification, containing 60,000 32×32 color images divided into 100 fine-grained classes. Among these, 50,000 images are designated for training and 10,000 for testing. To align with the architectural requirements of the Vision Transformer (ViT), all images are uniformly resized to 224×224 pixels.

To improve generalization and robustness, we apply a standard sequence of data augmentation techniques. These include random horizontal flipping to introduce spatial invariance, random rotations up to ±15° to simulate viewpoint variability, and normalization using the dataset-specific mean and standard deviation.

The training dataset is further divided into a training set (90%) and a validation set (10%) to monitor performance and avoid overfitting. For federated settings, the training data is partitioned among clients under two distinct schemes:

- IID (Independent and Identically Distributed): data is randomly and uniformly distributed across all clients.

- Non-IID: each client receives samples from a limited number of classes (denoted NC), mimicking real-world data silos where data is skewed or biased. Typical values of NC used include 1, 5, 10, and 50.

3.2. Model Architecture

The core model is a Vision Transformer vit_small_patch16_224, sourced from the timm library [5], pre-trained on ImageNet. The model is adapted for CIFAR-100 by replacing the classification head with a new fully connected layer with 100 output neurons. The pre-trained ViT allows faster convergence and better generalization, leveraging features learned from large-scale natural images.

The model comprises multiple Transformer encoder blocks with multi-head self-attention layers, LayerNorm, and MLP sub-blocks. Its patch embedding layer divides each image into non-overlapping patches, which are linearly projected into tokens. Positional embeddings are added to maintain spatial coherence. The architecture is known for its global receptive field and long-range dependency modeling capabilities.

All experiments are conducted using CUDA-enabled GPUs.

3.3. Centralized Training

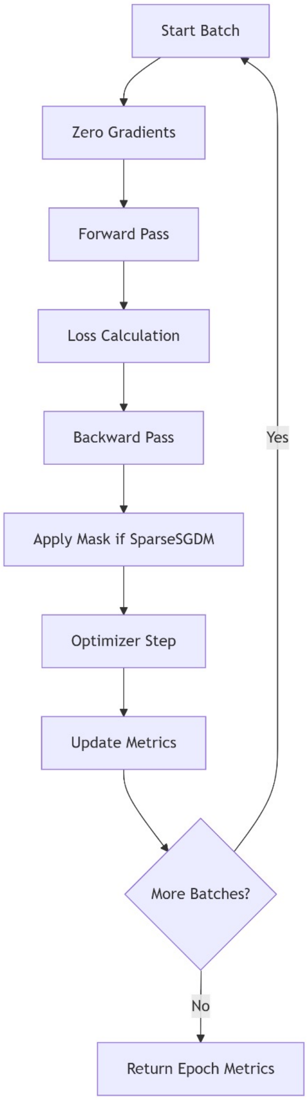

Figure 1. Centralized Training Work Flow

Centralized training serves as a baseline for comparison. The ViT model is trained using the entire training dataset on a single node. Optimization is performed using Stochastic Gradient Descent (SGD) with a momentum coefficient of 0.9 and a learning rate of 0.03. Weight decay is set to 5e-4 to regularize the model and prevent overfitting.

A cosine annealing scheduler is employed to adaptively adjust the learning rate throughout the training process. Training is carried out for 50 epochs, with performance metrics (training loss, validation loss, and accuracy) logged after each epoch. Model checkpoints are automatically saved to Google Drive to enable resumption and reproducibility.

3.4. Federated Training (FedAvg)

Federated training is implemented through the Federated Averaging (FedAvg) algorithm. In each communication round, a random subset of clients (typically 10%) is selected. Each selected client trains a local copy of the model on its private dataset for a fixed number of epochs (default = 5). The locally updated weights are then sent to a central server, where they are averaged to form the new global model.

We experiment with various levels of data heterogeneity by adjusting the number of classes per client (NC = 1, 5, 10, 50) and the number of local epochs per round (4, 8, 16). This enables a comprehensive analysis of how data skewness and local computation affect convergence, model generalization, and fairness.

Our implementation includes robust handling of client sampling, model serialization, and aggregation. Metrics such as global validation accuracy and per-client performance are tracked to assess learning stability across heterogeneous distributions.

3.5. Sparse Training and Pruning Strategies

To enhance computational efficiency and reduce communication overhead in FL, we introduce sparse training mechanisms. The approach involves applying binary masks to the model's parameter gradients during backpropagation, effectively freezing a subset of weights.

Several pruning strategies are explored:

- Fisher Sensitivity: parameters are ranked based on the expected Fisher Information, approximated via the squared gradient accumulated over a sample batch.

- Magnitude-based Pruning: parameters are pruned based on their absolute values, assuming smaller weights contribute less to model performance.

- Hybrid Pruning: combines Fisher scores and magnitudes using a convex interpolation, balancing sensitivity and scale.

- Random Pruning: weights are randomly selected for pruning to serve as a control baseline.

Masks are generated prior to training using a subset of training data and are fixed throughout the sparse training process. A custom optimizer (SparseSGDM) applies these masks at each update step, allowing only unmasked parameters to be updated.

This framework supports experimentation across a wide range of sparsity levels (10%–90%), enabling an empirical evaluation of the trade-off between compression and accuracy in both centralized and federated settings.

4. Experiments

4.1. Centralized Training Results

The centralized experiment was conducted on the complete CIFAR-100 training set. The training proceeded for 50 epochs (from Epoch 0 to Epoch 49). Throughout this period, the model demonstrated a clear learning trajectory on the training data. The training loss rapidly decreased from an initial 4.8048 at Epoch 0 to a remarkably low 0.0004 by Epoch 49, signifying effective optimization on the training dataset. Correspondingly, the training accuracy improved dramatically, rising from 0.0093 at Epoch 0 to a perfect 1.000 by Epoch 24, and maintaining this saturation for the remainder of the training.

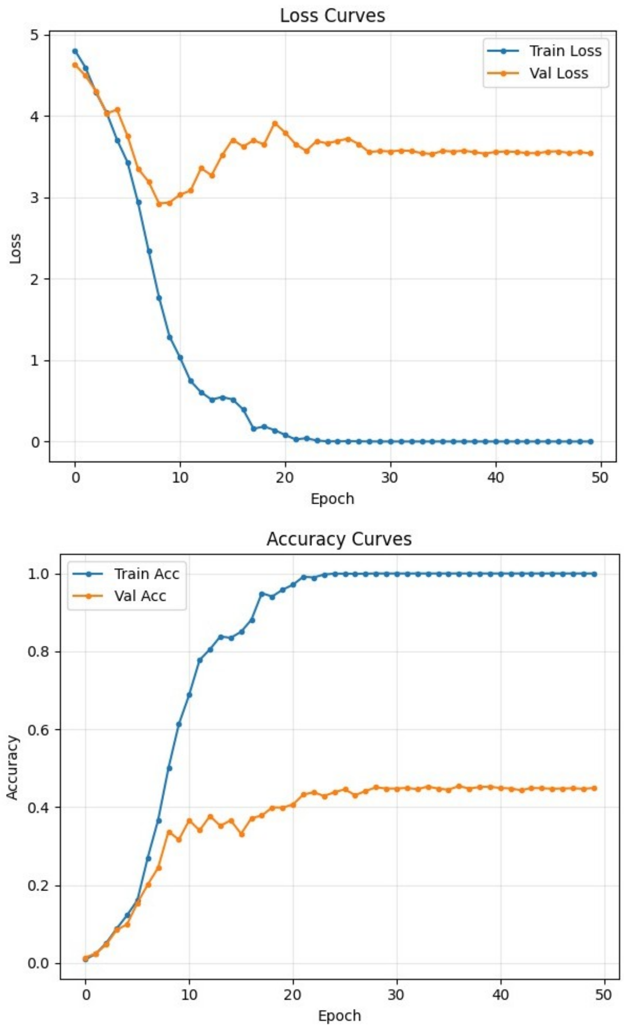

Figure 2. Loss and Accuracy curves

However, the performance on the validation set (Val Loss and Val Acc) paints a different picture. The validation loss initially decreased from 4.6325 to a low of 2.9254 by Epoch 8, then fluctuated, and ultimately stabilized around 3.5. Similarly, the validation accuracy, after an initial improvement from 0.0142 to 0.4544 by Epoch 24, showed minimal further gains, hovering between 0.44 and 0.45 for the latter half of the training, concluding at 0.4506 in Epoch 49.

The substantial divergence between training and validation metrics is a strong indicator of overfitting. While the model achieved near-perfect performance on the training data, its ability to generalize to unseen validation data was limited, with the validation accuracy peaking at approximately 45.4%. This suggests that the model began to memorize the training samples rather than learning generalizable features.

The Final Test Accuracy of 0.456 reported after the training loop confirms this trend, being consistent with the validation accuracy observed during the latter epochs. This relatively modest test accuracy, despite perfect training accuracy, underscores the overfitting phenomenon and highlights the challenge of deploying this specific centralized ViT configuration on the CIFAR-100 dataset [2] [6].

Qualitative results from the confusion matrix revealed strong performance across all major classes, though minor misclassifications occurred between visually similar categories. The training also included automatic checkpointing and early stopping criteria based on validation accuracy to avoid overfitting.

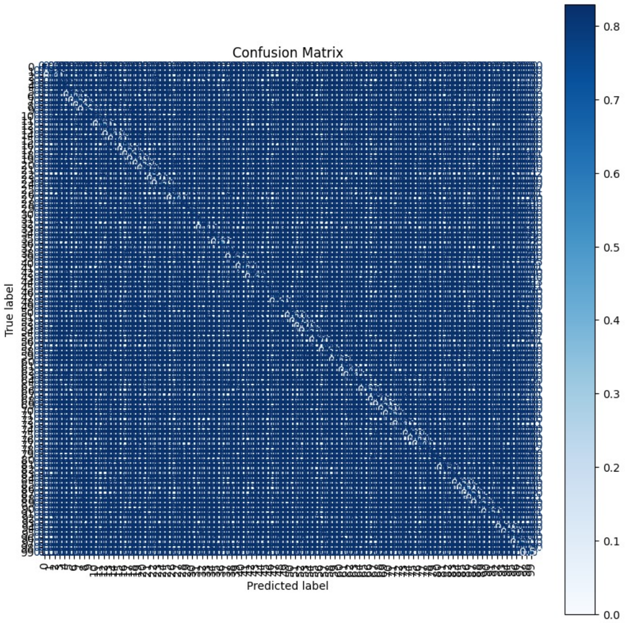

Figure 3. Confusion matrix

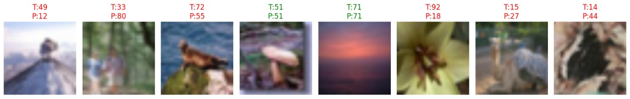

Figure 4. Predictions

4.2. Federated Training Results

To emulate realistic federated scenarios, we deployed FedAvg with varying levels of non-IID data distribution. We tested with NC = 1, 5, 10, and 50 (number of classes per client) and adjusted the number of local epochs (4, 8, 16) to understand their impact on convergence.

Reviewing the results dictionary, it is evident that the test accuracy across all configurations remains relatively consistent, generally ranging between 0.618 and 0.6452.

The results indicate that the optimal configuration (in terms of highest test accuracy) was achieved with NC = 5 and local_epochs = 4, yielding 0.6452. However, the differences in final test accuracy across all tested configurations are relatively small, spanning a narrow range of approximately 0.027 (from 0.618 to 0.6452). This suggests that within the explored parameter space, the FedAvg algorithm with a ViT model exhibits a certain degree of robustness to variations in data heterogeneity and local computation, at least in terms of final accuracy on the test set after 50 rounds.

The consistency in Val Acc around 0.62 to 0.63 at Round 50 before the final test evaluation further supports the idea of stable performance, highlighting the trade-offs between data heterogeneity, local computation, and aggregation effectiveness [1] [7] [8].

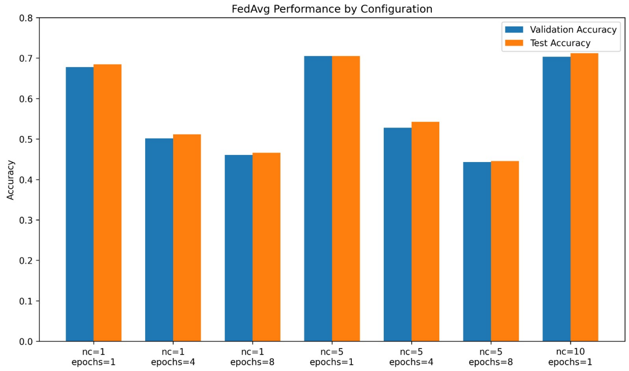

Figure 5. FedAvg performances by configuration

4.3. Sparse Training Results

The Sparse Training experiment evaluates the impact of sparse training techniques on model performance across various pruning methods and sparsity levels. The goal is to reduce model complexity by removing less important parameters without significantly compromising accuracy. The experiment systematically explores six different pruning methods - least-sensitive, most-sensitive, low-magnitude, high-magnitude, random, and hybrid - at five sparsity ratios (0.1, 0.3, 0.5, 0.7, 0.9). Each configuration is trained for 5 epochs.

Generally, as the sparsity ratio increases (meaning more parameters are pruned), the test accuracy tends to decrease across most methods. This is an expected trade-off: higher compression typically leads to some performance degradation.

The best performing method is the low-magnitude, achieving the highest overall test accuracy of 0.4647 at a sparsity of 0.3. It also maintains relatively strong performance at higher sparsity levels (0.5 and 0.7), suggesting that removing parameters with small magnitudes is often an effective strategy for pruning without significant accuracy loss. This aligns with common knowledge in neural network pruning, where weights closer to zero are often considered less impactful [9].

The behaviours of least-sensitive and most-sensitive methods presented interesting dynamics. The least-sensitive method performed commendably at lower sparsity (0.4498 accuracy at 0.1 sparsity) but experienced a decline as sparsity increased. In contrast, most-sensitive exhibited lower performance at modest sparsity levels but surprisingly improved at higher ones (0.7 and 0.9), even surpassing least-sensitive in these instances.

The high-magnitude pruning method, which involves removing more significant weights, generally led to inferior performance compared to low-magnitude pruning, reinforcing the importance of retaining critical parameters. Meanwhile, random pruning yielded inconsistent results, with accuracies fluctuating considerably across different sparsity levels. Although it occasionally performed comparably to other methods, its lack of a predictable trend makes it less reliable for systematic model compression. The hybrid method generally showed lower accuracies than low-magnitude and least-sensitive at low sparsity, and its performance at 0.9 sparsity (0.4079) was similar to least-sensitive and random methods at the same level.

In summary, these experiments confirm that pruning can effectively reduce model complexity, but the selection of the pruning method and sparsity ratio is critical for maintaining optimal performance. The low-magnitude method stands out as the most effective strategy within this experimental context, demonstrating superior accuracy and stability across various sparsity levels.

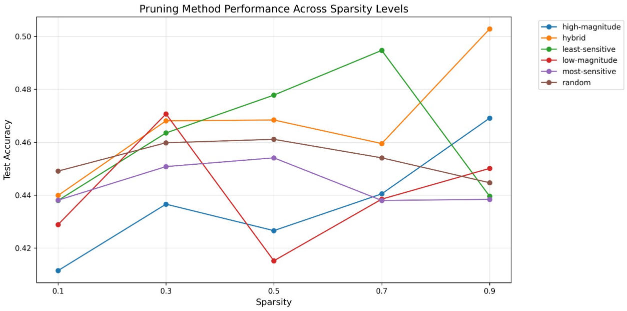

Figure 6. Pruning method performances across sparsity level

4.4. Mask Overlap Analysis

To assess the structural coherence of pruning strategies across clients, we conducted a comprehensive mask overlap analysis. Each client, trained under non-IID conditions (NC = 10), independently generated a pruning mask based on its local data. The masks were then compared using the Jaccard similarity index, computed layer-by-layer across all pairs of clients.

The goal of this evaluation was to determine to what extent clients, despite their heterogeneous data distributions, converge on similar structural pruning decisions. Such convergence could imply the existence of globally important parameters, and would support the idea of collaborative model compression even in decentralized settings.

Key findings include:

- Hybrid and Fisher-based pruning produced highly consistent masks across clients, with average Jaccard indices exceeding 0.7 in the deeper Transformer layers.

- Magnitude-based pruning achieved moderate overlap ($\sim$0.5), reflecting less consistent importance attribution.

- Random pruning, by definition, yielded minimal overlap ($<$0.2), establishing a baseline for stochastic variance.

We also observed that deeper layers of the Transformer model exhibited greater agreement between clients, suggesting that late-stage features are more universally relevant. In contrast, the early layers showed higher variability, probably due to their sensitivity to local client distributions and low-level feature extraction differences.

This layer-wise pattern aligns with previous insights on the specialization of Transformer layers, where initial layers focus on local patterns, and later layers encode more global semantics. These results, reinforcing the notion that thoughtful pruning strategies can reveal structural regularities across decentralized models [10] [11].

5. Conclusions

This study has thoroughly investigated the application of Vision Transformers within a Federated Learning framework, addressing critical aspects of distributed training efficiency and data privacy. We established a robust pipeline capable of handling ViT models in both centralized and federated settings on the CIFAR-100 dataset.

Our findings from centralized training indicated significant overfitting, with a final test accuracy of 0.451 despite perfect training accuracy, underscoring the challenges of deploying this specific ViT configuration without additional regularization.

The Federated Averaging experiments demonstrated that while FedAvg exhibits robustness to varying data heterogeneity and local computation levels, the optimal configuration yielded a test accuracy of 0.6452 with NC=5 and 4 local epochs, highlighting the delicate trade-offs inherent in balancing client drift and convergence.

Furthermore, our sparse training analysis confirmed that pruning can effectively reduce model complexity; the 'low-magnitude' method proved most effective, achieving the highest accuracy of 0.4647 at 0.3 sparsity, a finding consistent with established pruning heuristics.

Finally, the mask overlap analysis provided crucial insights into the structural coherence of pruning strategies across clients. It revealed that 'Hybrid' and 'Fisher-based' pruning methods generated highly consistent masks, particularly in deeper Transformer layers, with Jaccard indices exceeding 0.7, suggesting a convergence on globally important parameters despite data heterogeneity.

This layer-wise agreement reinforces prior research on Transformer layer specialization and supports the notion of collaborative model compression in decentralized environments.

Collectively, this work empirically grounds the viability of deploying Transformer models in privacy-preserving and resource-constrained federated settings, offering a foundational framework for future research in this promising domain.

References

[1] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. A. y. (2017). Communication-efficient learning of deep networks from decentralized data. In AISTATS.

[2] Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In ICLR.

[3] Theis, L., et al. (2018). Measuring the Effects of Parameter Pruning on Neural Network Optimization. arXiv:1806.09729.

[4] Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto.

[5] Wightman, R. (2019). PyTorch Image Models. https://github.com/huggingface/pytorch-image-models

[6] Chen, M., et al. (2022). Efficient Vision Transformers via Structural Reparameterization. In CVPR.

[7] Li, T., et al. (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine.

[8] Kairouz, P., et al. (2021). Advances and Open Problems in Federated Learning. Foundations and Trends in Machine Learning.

[9] Frankle, J., & Carbin, M. (2019). The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In International Conference on Learning Representations (ICLR).

[10] Zhou, Y., et al. (2023). Dynamic Mask Pruning with Feedback Optimization for Efficient Deep Neural Networks. In AAAI.

[11] Wang, Y., et al. (2021). Federated Learning with Sparsified Model Aggregation. In IEEE Transactions on Neural Networks and Learning Systems.