Enhancing Federated Learning Through Low-Rank Adaptation and Singular Value Decomposition

by

Yixuan Chen

Thesis submitted to the
Deanship of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the degree in
ML

Department of Machine Learning
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

Supervisor(s): Dr. Samuel Horvath
Assistant Professor, Dept. of ML,
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)
Dr. Martin Takac
Associate Professor , Dept. of ML,
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

Internal Member: Dr. Bin Gu
Assistant Professor, Dept. of ML,
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis investigates an innovative direction in the field of federated learning (FL) by enhancing model communication efficiency and handling data heterogeneity through the integration of Low-Rank Adaptation (LoRA) and Singular Value Decomposition (SVD) techniques. As distributed data sources surge, especially in today's era where privacy preservation is increasingly emphasized, FL offers an effective framework for training models without centralizing data. However, one of the main challenges FL faces is efficiently handling non-independent and identically distributed (non-IID) data while minimizing the communication overhead during model updates.

To address these challenges, this study proposes a federated learning method that combines truncated SVD with LoRA. This method aims to optimize the model parameter update and transmission process through low-rank matrix techniques, thereby reducing communication costs and enhancing model adaptability. Experimental validation on the MNIST and CIFAR-10 datasets demonstrates the effectiveness of the proposed method in dealing with data heterogeneity and compares its performance with traditional FL algorithms, such as FedAvg and FedProx.

The results indicate that, compared to standard FL algorithms, our method shows improved performance under various settings, especially in environments with highly heterogeneous data distributions. Moreover, by adjusting the degree of low-rank adaptation, our method can find an optimal balance between model complexity and accuracy across different data distribution scenarios, thus mitigating the risk of overfitting and enhancing the model's generalizability.

The contribution of this research lies in proposing a FL framework that combines SVD and LoRA, offering a new perspective and approach for model optimization in federated learning. This method not only provides solutions to the current challenges of communication efficiency and data heterogeneity in FL but also opens up a new pathway for optimizing distributed machine learning models using low-rank matrix techniques.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Dr. Samuel Horvath, for his invaluable guidance and insightful advice throughout the course of my master's studies. Dr. Horvath has not only been a mentor of high calibre but also a great inspiration through his rigorous academic standards and profound dedication to excellence in research.

His meticulous attention to detail and supportive nature have been instrumental in overcoming the challenges encountered during my research. The profound insights and constructive suggestions provided by Dr. Horvath have significantly enriched the content of my thesis, making my work more comprehensive and profound.

Furthermore, I am thankful for all my peers and friends who participated in this research, whose advice and assistance were crucial to the completion of this thesis. I am also grateful for my family, whose understanding and support have allowed me to balance the demands of my studies with personal responsibilities.

Once again, I extend my thanks to Dr. Horvath and everyone who has supported me on this journey. Thank you all!

Dedication

To my loving family, for their unwavering support and endless patience throughout my academic journey.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence.

Chapter 1

Introduction

Over the last decade, machine learning has evolved from an academic theory into a powerful tool that drives innovation in nearly every field. This significant growth is attributed to the explosion in data generation and substantial advancements in computing power. Deep learning, a subset of machine learning, has enabled computers to identify patterns and make decisions with minimal human intervention. However, this rapid growth has not come without its challenges. Beyond a few industries that can fully leverage big data to drive the development of AI technologies, most sectors still grapple with limited data availability and poor data quality.[13] This situation restricts the implementation and further development of AI technologies, especially in those professional fields where high quality and quantity of data are critically required [18].

Particularly noteworthy is the professional field of healthcare, where there is an urgent demand for high-quality annotated data, yet obtaining such data is exceptionally challenging and costly. Not only does annotating medical data require the precious time of medical professionals, but it also involves privacy and accuracy requirements far beyond those of other fields. [24] It is estimated that outsourcing medical data annotation to third-party companies would require tens of thousands of people and a decade to collect sufficient valid data. This estimate highlights the severe data scarcity even in scenarios where efforts to expand data collection are made. Moreover, the issue of data silos further complicates the utilization of data. In most industries, even within different departments of the same company, there are significant barriers to data integration, including industry competition, privacy and security protection, and complex administrative procedures. [9] Not to mention the integration of data across institutions and regions, which is almost an impossible task or one that would require immense costs.

With the further development of big data, data privacy and security issues have become a global concern. Every public data breach, such as the Facebook data leakage case, attracts significant attention from the media and public. At the same time, governments worldwide are strengthening protections for data security and privacy, as evidenced by the implementation of laws and regulations like the General Data Protection Regulation (GDPR) [1] by the European Union and the Cybersecurity Law by China. These legal and regulatory measures pose unprecedented challenges to the AI field, involving not only the legality of data collection but also compliance in the processing, transfer, and use of data.

Given these challenges, traditional data processing and AI model training methods have clearly reached a bottleneck, especially when involving data sharing and use across entities, often conflicting with increasingly strict data protection regulations. [34] Hence, there is an urgent need for a new technological solution that can both protect data privacy and security and efficiently utilize scattered data resources.

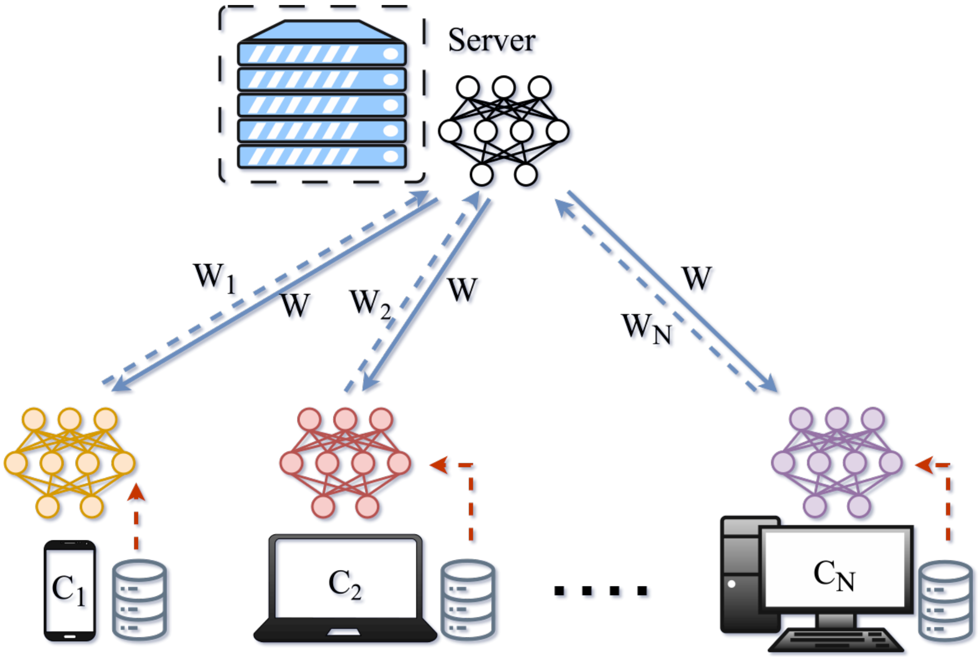

Figure 1.1: Illustration of Federated Learning

In response to these issues, federated learning, proposed by McMahan et al. [21], has emerged as an innovative approach. It allows the training of machine learning models across numerous decentralized devices (such as smartphones or laptops), while keeping the training data on the device, as Figure 3.1. The concept has been further broadened by Kairouz et al. [19] to encompass a wider range of collaborative environments. This method offers significant advantages in terms of privacy and data security because sensitive data does not need to be centralized or shared. Federated learning decentralizes computation, shifting the training burden to edge devices. [19] As the computational capacity of edge devices improves, both the application range and efficiency of federated learning have significantly increased.

In recent years, federated learning has proven its worth as a privacy-preserving machine learning approach, finding applications across diverse fields. In healthcare, initiatives like the 4CE Consortium leverage federated learning to analyze global electronic health records of COVID-19 patients [28], safeguarding privacy while enhancing disease diagnostics and treatment strategies. Similarly, in finance, projects such as FATE [20] enable collaboration among institutions for risk management and credit assessment models without compromising sensitive data. The adoption of federated learning extends to personal devices, with

Google's Gboard [32] utilizing on-device federated learning to improve word prediction and intelligent input features across millions of smartphones while respecting user privacy. Moreover, in the automotive industry, efforts like Volkswagen's HEAT project [25] employ federated learning to accelerate the development of autonomous driving technology by sharing learned experiences among vehicles without exchanging raw data, ensuring safety and privacy on the roads.

However, federated learning introduces new challenges, such as ensuring consistent model performance across diverse data distributions and managing efficient communication between devices. [30] Detailed show in 1.2.

1.1 Motivation

Concurrently, there has been a significant trend towards developing larger, more complex machine learning models. These models, especially in deep learning domains like natural language processing and computer vision, have set new benchmarks for accuracy and performance. However, their size and complexity come with substantial computational costs and a need for extensive training data. This poses a challenge for deployment in environments with constrained resources. Moreover, these large models often require significant fine-tuning when applied to specific tasks or datasets, further increasing the computational burden.

Low-Rank Adaptation (LoRA) becomes pivotal at this juncture. LoRA is an innovative approach that efficiently adapts large, pre-trained models for specific tasks. It achieves this by applying low-rank matrices to modify the model's weights, a method that reduces the resource requirements typically associated with training these models from scratch. The adaptability of LoRA to different tasks and settings makes it an attractive solution for optimizing large models, particularly in resource-constrained environments. [16]

Inspired by LoRA[16] and benefiting from its low-rank nature, integrating LoRA into Federated Learning (FL) with some enhancements has been proposed to address the aforementioned challenges. Although LoRA was originally designed for fine-tuning large language models (LLMs), its core concept involves integrating low-order parameter matrices with fewer trainable parameters into pre-trained models to adapt to downstream tasks. Therefore, the amalgamation of FL with LoRa presents several advantages. Firstly, the low-rank structure allows clients to focus their learning efforts on the most critical parts of local knowledge, reducing the potential for overfitting to local knowledge, especially when client data is very limited. Secondly, by significantly reducing the number of trainable parameters, LoRa helps to retain most of the general knowledge acquired from other clients, thereby enhancing generalization. Concurrently, reducing the number of trainable parameters implies a decrease in communication parameters, which can substantially improve communication efficiency.

1.2 Federated Challenges

1.2.1 Data heterogeneity

Data heterogeneity in the context of federated learning refers to the phenomenon where different clients (which could be devices or data centers participating in the training process) possess data distributions that are not identical.[12] This inconsistency can manifest across multiple dimensions, including but not limited to, the volume of data, distribution of features, and distribution of labels. Data heterogeneity can be further categorized into:

1. **Feature Heterogeneity**: Different devices may collect or focus on different sets of features. For example, in a medical federated learning project, different hospitals might use different tests to diagnose the same disease, therefore the feature sets they collect could vary.

2. **Label Heterogeneity**: In some cases, even for the same set of features, the data on different devices may have different label distributions. For example, in a federated learning model for loan approval, banks in different regions might have different approval criteria, leading to label distribution heterogeneity.

3. **Sample Size Heterogeneity**: In federated learning, different devices may have varying amounts of data samples. Some devices might have a large amount of data, while others may only have a few samples. This kind of heterogeneity can lead to imbalances during model training.

4. **Data Distribution Heterogeneity**: Even if the features, labels, and number of samples are similar, the data distribution on different devices may significantly differ. This difference could be caused by factors such as geographic location, user behavior, and device type. Data distribution heterogeneity could impact the model's generalization ability.

The presence of data heterogeneity poses multiple challenges for federated learning.[19, 22] Firstly, it can lead to inconsistent model performance across different datasets, with the model potentially overfitting on certain types of data and underperforming on others. This inconsistency exacerbates the issue of model generalizability, as finding a single model that performs well across all clients becomes increasingly difficult. Moreover, the significant differences in data among clients can slow down the model training process, requiring more time for convergence, especially when some clients have considerably more data than others. Lastly, data heterogeneity can introduce fairness issues, where the model might exhibit biases towards the data of certain clients, potentially correlating with inherent socio-economic or demographic characteristics within the data.

To tackle the issue of data heterogeneity, researchers have proposed a variety of strategies:

1. **Personalized Models**[4, 27, 26]: Tailoring or adjusting the model for each client to fit its specific data distribution, thereby improving the model's performance on the client-side.

2. **Meta-learning and Transfer Learning**[10, 17, 3, 7]: Utilizing meta-learning or transfer learning approaches to aid the model in better transitioning and generalizing across different data distributions.

3. **Aggregation Algorithm Optimization**[5, 29]: Optimizing the model aggregation phase algorithms (e.g., using weighted averaging or more complex aggregation strategies), taking into account the data distribution and contributions of different clients to enhance the overall model performance.

4. **Data Augmentation and Synthesis**[6, 11]: Employing data augmentation or synthesis techniques, without compromising privacy, to simulate various data distributions and boost the model's generalization capabilities.

1.2.2 Communication efficiency

Communication efficiency issues refer to the performance bottlenecks and challenges encountered during the transmission of data or model parameters between nodes in a federated learning system. These problems are generally related to the time required for data transmission, bandwidth utilization, and the ability to perform effective communication under limited resource conditions. In many cases, communication efficiency problems lead to slower system operation, increased energy consumption, and difficulties in meeting performance requirements in real-time or low-latency application scenarios.[33]

The impact of model size on communication efficiency is a significant issue. This is because federated learning involves exchanging model parameters or gradient updates between the server and participating clients. The larger the model, the more data needs to be transmitted with each communication, leading to increased demands on communication bandwidth, higher energy consumption, and delays in updates.[15]

To enhance communication efficiency and address the challenges associated with model size, a core principle in federated learning is executing multiple training steps (i.e., local iterations) on local devices, followed by sending only the updated model parameters back to the central server for aggregation. This approach, initially employed by the Federated Averaging algorithm (FedAvg) [21] proposed by McMahan et al., significantly reduces communication costs by minimizing the amount of data that needs to be transmitted over the network. In each communication round, selected clients independently train the model on their local data for several iterations and then send their model updates back to the server for aggregation. By doing so, it is possible to effectively reduce the latency and energy consumption associated with data transmission without sacrificing too much in terms of model performance.

However, even with the adoption of the FedAvg algorithm, the size of the model remains a critical factor impacting communication efficiency. Despite completing multiple iterations locally, transmitting the parameters of a large model in each communication round still presents a substantial challenge. To address the issue of model size, researchers and engineers have proposed several solutions, mainly including the following strategies:

1. **Model Compression**[14]: Reducing the size of the model through techniques such as parameter pruning and knowledge distillation. Parameter pruning involves removing unimportant weights from the model, while knowledge distillation is about transferring the knowledge from a larger model to a smaller model to reduce its size.

2. **Quantization**[2]: Quantizing model parameters and gradients to a lower precision (e.g., reducing from 32-bit floating point numbers to 8-bit or lower). This not only reduces the amount of data needed for communication but can also maintain model performance to some extent.

    - **Uniform Quantization** One of the simplest forms of quantization that maps a continuous range of values onto a finite number of discrete levels. Given a real number x, uniform quantization can be expressed as:

    $$Q(x) = \Delta \cdot \left\lfloor \frac{x}{\Delta} + \frac{1}{2} \right\rfloor$$

    where Δ is the quantization step size, and $\lfloor \cdot \rfloor$ denotes the floor operation.

3. **Sparse Representation**[31, 23]: Reducing the amount of data for communication by only transmitting non-zero parameters in updates. This approach assumes that most gradient updates are close to zero and therefore can be ignored.

    - **Top-k Sparsification** Only the top k gradients with the largest absolute values are transmitted. The mathematical expression involves selecting a set S that contains the indices of the top k absolute values of all gradients, then only updating these k gradients:

    $$g_i' = \begin{cases} g_i & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$$

    where g_i' is the sparsified gradient, and g_i is the original gradient.

4. **Parameter Decomposition**[8]: Applying mathematical methods such as Singular Value Decomposition (SVD) to reduce the number of model parameters, thus decreasing the required communication bandwidth by transmitting a low-rank approximation of the parameters.

1.3 Structure of the Thesis

This thesis is organized into five chapters that systematically explore the integration of Low-Rank Adaptation (LoRA) and Singular Value Decomposition (SVD) within Federated Learning (FL) frameworks. The following sections provide a detailed blueprint of the thesis' composition and focal points.

Chapter 1: Introduction

This chapter introduces Federated Learning, highlighting its significance in the era of data privacy and the need for efficient computational methods due to the challenges posed by data heterogeneity. The motivation behind integrating LoRA and SVD to enhance FL is discussed, emphasizing the potential improvements in communication efficiency and model performance. The objectives of the research are outlined, focusing on the demonstration of the proposed method's effectiveness in handling non-IID data across distributed networks.

Chapter 2: Problem Formulation

A detailed theoretical background on the challenges of Federated Learning is provided, including the mathematical formulations that underpin the model aggregation issues in FL environments. This chapter reviews existing FL algorithms such as FedAvg and FedProx, exploring their limitations in dealing with data heterogeneity and communication inefficiencies. The need for novel methodologies that can address these shortcomings is articulated, setting the stage for the subsequent chapters.

Chapter 3: Methodology

The novel Federated Learning method combining truncated SVD with LoRA is introduced. This chapter details the algorithmic structure developed for integrating these techniques within the FL framework, including pseudocode and mathematical validations. The methodologies for model compression and efficient data transmission, which are critical for enhancing FL system scalability and efficiency, are thoroughly discussed.

Chapter 4: Experiments

This chapter describes the experimental setup, including the datasets used (MNIST and CIFAR-10), configuration details, and the rationale behind the choice of metrics and data. It presents a comprehensive analysis of the experimental results, comparing the performance of the proposed FL method against traditional FL methods. The strengths and potential limitations of the LoRA and SVD integration approach are critically analyzed.

Chapter 5: Conclusion and Future Work

The final chapter summarizes the key findings of the research, emphasizing the improvements enabled by the proposed method in FL systems' capability. The theoretical and practical contributions of the thesis to the fields of machine learning and federated learning are discussed. Future research directions are proposed, suggesting potential methodological enhancements and exploration of additional datasets and environments.

Chapter 2

Problem Formulation

The goal of federated learning is to train a global model by learning from decentralized data located on multiple clients. This is achieved without directly sharing the data among clients or with a central server. Formally, the problem can be stated as follows:

Given a set of N clients, each with its own local dataset D_i, the objective is to train a global model with parameters $\mathbf{w}$ that minimizes the global loss function $F(\mathbf{w})$, which is defined as a weighted sum of the clients' local loss functions $f_i(\mathbf{w})$:

$$\min_{\mathbf{w}} F(\mathbf{w}) = \min_{\mathbf{w}} \sum_{i=1}^{N} q_i f_i(\mathbf{w}), \tag{2.1}$$

where

- $\mathbf{w}$ represents the parameters of the global model,
- $f_i(\mathbf{w})$ is the local loss function for client i which measures the prediction error of the model on the local dataset D_i,
- q_i is the weight of client i's loss in the global loss function, often chosen to reflect the size of D_i relative to the total data size, such that $\sum_{i=1}^{N} q_i = 1$. One common approach is to determined by the proportion of the size of the local data size, i.e.,

$$q_i = \frac{|D_i|}{\sum_{j=1}^{N} |D_j|}, \quad \text{for } i = 1, 2, \ldots, N. \tag{2.2}$$

2.1 FedAvg

The Federated Averaging Algorithm (FedAvg), proposed by McMahan et al. [21], represents a cornerstone in federated learning, aiming to collaboratively train a global model across multiple clients while minimizing communication overhead. Each client i holds a local dataset D_i, and the objective is to train a global model with parameters $\mathbf{w}$ by leveraging the decentralized datasets without compromising data privacy.

The FedAvg algorithm operates by distributing the current global model parameters $\mathbf{w}$ to a subset of clients in each training round. These clients update the model parameters locally using their datasets D_i for a predefined number of epochs E. The local update rule on client i for epoch t can be expressed mathematically as:

$$\mathbf{w}_i^{t+1} = \mathbf{w}_i^t - \eta \nabla F_i(\mathbf{w}_i^t),$$

where η denotes the learning rate and $\nabla F_i(\mathbf{w}_i^t)$ represents the gradient of the loss function F_i with respect to the model parameters $\mathbf{w}_i^t$ on client i's local data. After performing E epochs of local updates, the clients compute the difference in model parameters $\Delta \mathbf{w}_i = \mathbf{w}_i^{t+1} - \mathbf{w}_i^t$ and send these updates back to the server.

The server then aggregates these updates from all participating clients and updates the global model parameters according to:

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \frac{1}{K} \sum_{i=1}^{K} \Delta \mathbf{w}_i,$$

where K is the number of clients that participated in the training round. This process iteratively continues, with the global objective of minimizing the aggregated loss function defined as:

$$F(\mathbf{w}) = \sum_{i=1}^{N} \frac{n_i}{n} F_i(\mathbf{w}),$$

with $n_i = |D_i|$ representing the number of samples in client i's dataset, and $n = \sum_{i=1}^{N} n_i$ denoting the total number of samples across all clients.

Despite its success in reducing communication overhead, the limitations of FedAvg in handling data heterogeneity have become increasingly apparent.

Data heterogeneity, or the inconsistency in data distributions across different clients, poses a significant challenge in federated learning. In practice, the environment and methods by which each client collects and processes data can vary, leading to substantial differences in local data distributions. This inconsistency can increase gradient variance during model training, affecting training efficiency and the final performance of the model.

Specifically, in each round of training, the FedAvg algorithm begins by distributing the current global model to a subset of selected clients. These clients perform multiple rounds of training locally using their own data, then compute and send the updates of model parameters back to the server. The server collects all updates from participating clients, calculates the average, and uses it to update the global model. However, due to data heterogeneity among clients, the parameter updates calculated by different clients can vary significantly, causing the direction of the global model update to potentially be suboptimal. Moreover, performing multiple rounds of local updates can lead to model parameters "overfitting" to the specific data distribution of individual clients, thereby increasing the bias of the global model across different clients.

Besides data heterogeneity, FedAvg also faces a trade-off between communication efficiency and model performance. While reducing the number of communication rounds can lower communication costs, selecting too few clients per round or setting an inappropriate number of local training rounds can affect the convergence speed and ultimate performance of the model.

Therefore, although the FedAvg algorithm has provided an important foundation for research and application in federated learning, it still faces significant challenges in handling data heterogeneity and balancing the trade-off between communication efficiency and model performance. These challenges have motivated researchers to explore new algorithms and techniques to improve the adaptability and efficiency of federated learning in complex data environments.

2.2 SCAFFOLD

In the federated learning environment, data heterogeneity leads to decreased training efficiency and final model performance. To address this issue, the SCAFFOLD (Stochastic Controlled Averaging Federated Learning) algorithm was proposed. Its core idea is to introduce control variables at each client and the server to reduce the variance of gradients, thereby improving training efficiency and model accuracy.

Specifically, suppose there is a set of clients N, each client i owns its local dataset D_i. The goal of federated learning is to collaboratively train a global model across all clients, with its parameters denoted as $\mathbf{w}$. In the SCAFFOLD algorithm, besides maintaining local model parameters $\mathbf{w}_i$, each client i also introduces a local control variable c_i. The server, on the other hand, maintains global model parameters $\mathbf{w}$ and a global control variable c.

At the beginning of the training process, the server initializes the global model parameters $\mathbf{w}$ and the global control variable c. In each training round, the server distributes the current global model parameters $\mathbf{w}$ and the global control variable c to selected clients. Each receiving client i computes the gradient $\nabla f_i(\mathbf{w}_i)$ based on its local data, where f_i is the loss function defined on client i. Then, the client adjusts its gradient using the following formula:

$$\tilde{\nabla} = \nabla f_i(\mathbf{w}_i) - c_i + c$$

Here, $\tilde{\nabla}$ represents the gradient adjusted by the control variables. The client updates its local model parameters based on the adjusted gradient:

$$\mathbf{w}_i' = \mathbf{w}_i - \eta\tilde{\nabla}$$

where η is a predefined learning rate.

After completing the local update, the client calculates the change in model parameters $\Delta\mathbf{w}_i = \mathbf{w}_i' - \mathbf{w}_i$ and the change in the control variable Δc_i, and then sends these changes back to the server. The server collects updates from all participating clients and aggregates the global model parameters and global control variable using the following formulas:

$$\mathbf{w} = \mathbf{w} + \frac{1}{N}\sum_{i=1}^{N} \Delta\mathbf{w}_i$$

$$c = c + \frac{1}{N}\sum_{i=1}^{N} \Delta c_i$$

Initialization of control variables is pivotal for the efficient convergence of SCAFFOLD. Optimal strategies may include zero initialization or small random values, particularly tailored to the data's distribution characteristics across clients.

Theoretical underpinnings demonstrate that SCAFFOLD achieves variance reduction in gradient estimates by effectively compensating for the biased gradients due to non-IID data distributions. This is achieved through the control variables that align local updates more closely with the global optimal gradient direction.

Through this method, the SCAFFOLD algorithm effectively solves the gradient variance problem caused by data heterogeneity in federated learning, improving the efficiency and accuracy of model training.

Although the SCAFFOLD algorithm has significant advantages in reducing gradient variance and improving training efficiency, it also introduces additional computational and communication overheads. Each client and the server need to compute, update, and synchronize control variables, which not only increases the computational burden but also the communication load. Moreover, the performance of the algorithm highly depends on the appropriate setting of hyperparameters, such as the choice of the learning rate η, requiring careful adjustment and optimization in practical applications. Despite these challenges, the SCAFFOLD algorithm remains a powerful tool for improving the overall performance of federated learning systems while protecting privacy.

Exploring the integration of differential privacy with SCAFFOLD could offer enhanced privacy guarantees. This integration, however, might affect the convergence and efficiency of the algorithm, necessitating a balanced approach to privacy and performance.

2.3 FedProx

FedProx is an extension of the Federated Averaging (FedAvg) algorithm, designed to address the challenges posed by data heterogeneity across clients in federated learning environments. FedProx introduces a proximal term to the objective function that each client optimizes, allowing for more flexible local updates that can better accommodate the variance in local data distributions.

The core idea of FedProx is to mitigate the adverse effects of non-IID data distributions by slightly modifying the local objective functions. This modification encourages local models to stay closer to the global model, which can help in reducing the variance among the updates submitted by different clients.

In FedProx, similar to FedAvg, the global model is iteratively updated by aggregating local model updates from a subset of participating clients during each training round. However, the key difference lies in the local training objective on each client i, which is given by:

$$F_i(\mathbf{w}_i) = f_i(\mathbf{w}_i) + \frac{\mu}{2}\|\mathbf{w}_i - \mathbf{w}\|^2$$

Here, $F_i(\mathbf{w}_i)$ represents the local loss function based on the client's data, $\mathbf{w}_i$ denotes the local model parameters for client i, and $\mathbf{w}$ represents the current global model parameters. The term $\frac{\mu}{2}\|\mathbf{w}_i - \mathbf{w}\|^2$ is the proximal term added to the local objective function, where $\mu \geq 0$ is a tunable hyperparameter that controls the strength of regularization. This proximal term penalizes large deviations between the local model parameters and the global model parameters, encouraging more conservative updates that are likely to be more in line with the global model.

The proximal term effectively acts as a regularizer that minimizes the squared Euclidean distance between the local and global parameters. This regularization not only mitigates the skew induced by non-IID data but also theoretically improves the Lipschitz continuity of the loss function, which is crucial for enhancing convergence rates under variable data conditions across clients.

Under reasonable assumptions about the smoothness and convexity of the local loss functions, FedProx can achieve a convergence rate of $O(1/\sqrt{KT})$, where K is the number of clients and T is the number of communication rounds. This rate is particularly effective in scenarios where data distributions are highly skewed, making it a preferable choice over FedAvg, which might not converge under such conditions.

The inclusion of the proximal term in the local objective function allows FedProx to handle the statistical heterogeneity of local data distributions more gracefully than FedAvg. This approach can lead to improved convergence properties and more stable training across a wide range of federated learning scenarios, especially those characterized by significant data heterogeneity among clients.

Despite its advantages, FedProx also introduces challenges, primarily related to the selection of the hyperparameter μ. The optimal value of μ can vary significantly across different applications and datasets, requiring careful tuning to balance the trade-off between allowing meaningful local updates and ensuring these updates do not diverge too much from the global model. Furthermore, like all federated learning algorithms, FedProx must address issues related to communication efficiency and privacy preservation, especially as the proximal term introduces additional dependencies on the global model parameters.

Selecting the optimal μ involves trade-offs: a higher μ may overly constrain local updates, potentially leading to underfitting on local datasets, while a too-low μ might not sufficiently mitigate the variance caused by non-IID data. Practical strategies for tuning μ include using validation sets or adaptive methods that adjust μ based on the observed divergence of local updates from the global model during training.

In summary, FedProx provides a robust framework for federated learning in heterogeneous environments by incorporating a proximal term into the local training objective. This modification helps in mitigating the challenges posed by non-IID data distributions, leading to more stable and efficient training of global models. However, the effectiveness of FedProx depends on the careful selection of hyperparameters and the specific characteristics of the federated learning setup, including the degree of data heterogeneity and the computational capabilities of the clients.

2.4 Fine-tuning

Fine-tuning, as an efficient transfer learning strategy that addresses the challenge of adapting large pre-trained models to specific tasks, especially in scenarios where data scarcity is prevalent. It substantially reduces the computational resources and time costs associated with training large models from scratch. By starting with a model that has been pre-trained on a large and diverse dataset, fine-tuning allows for the model to quickly adapt to new tasks by making minimal adjustments to the model's parameters, thereby retaining much of the learned features that are general across tasks while adjusting subtle features specific to the new task.

Fine-tuning in large models typically involves adjusting the deeper layers of the model while keeping the initial layers fixed. This method is particularly beneficial because the initial layers capture universal features like edges and textures, which are useful across various tasks, while the deeper layers capture more specific features that are crucial for performance on particular tasks. This selective adjustment helps in mitigating overfitting—a common challenge with large models especially when the target dataset is small. Moreover, fine-tuning enhances model performance significantly by leveraging pre-learned representations, thus reducing the need for extensive computational power and data requirements that are typically necessary for training large models from scratch.

Federated learning, as an innovative distributed learning paradigm, aims to solve how to share the fruits of model learning among multiple data holders without directly sharing sensitive data, thus overcoming data privacy and data silo problems. Observing these two technologies, it's not difficult to find their common ground in terms of data utilization and model generalization.

Specifically, both fine-tuning techniques and federated learning strive to effectively leverage available data to improve the adaptability and performance of models in new environments. This commonality provides a theoretical basis for applying fine-tuning methods to federated learning. The integration is particularly seamless as fine-tuning can be adapted to the federated setting by updating the global model locally on each client using their specific data. This approach not only customizes the model for individual data distributions but also maintains overall model robustness and generalization across all clients.

Especially in a federated learning environment, fine-tuning techniques can help the model better adapt to each participant's specific data distribution and solve problems such as slow transmission speeds and limited computing power on clients' devices by reducing the number of model parameters and enhancing the model's computational efficiency. For instance, through technologies like Low-Rank Adaptation (LoRA), it's possible to significantly reduce the dimensionality of model parameters without significantly sacrificing model performance, making the model more efficient in federated learning environments, while also easing the computational and storage burdens on participants' devices.

Furthermore, applying fine-tuning techniques to federated learning brings additional benefits. It not only enhances the model's generalization ability in the face of diverse data but also improves the flexibility and efficiency of model training, enabling models to be trained and deployed on various devices, including those with limited resources. This practical application extends the application scope of federated learning technologies, making them more serviceable in real-world scenarios such as mobile computing and the Internet of Things, where data privacy and device computing power often pose constraints.

In summary, by integrating fine-tuning techniques with federated learning, we solve problems related to data privacy protection and efficient resource utilization, while also achieving rapid adaptation and optimization of models in different environments. This fusion provides new insights and possibilities for the development and application of deep learning technologies, proving the enormous potential of cross-domain technology integration and opening new directions for future deep learning research and practice.

Chapter 3

Methodology

This section presents the methodology adopted for enhancing federated learning with an innovative approach inspired by Low-Rank Adaptation (LoRA). Building upon the shortcomings identified in applying LoRA directly to federated learning (as detailed in Section 3.1), we propose a novel method that integrates truncated Singular Value Decomposition (SVD) to efficiently compress updates during the learning process. This integration serves to reconcile the benefits of LoRA's rank adaptation with the communication efficiency imperative in a federated context.

The core of our methodology is the introduction of an auxiliary weight matrix, denoted as $\boldsymbol{\sigma} \in \mathbb{R}^{m\times n}$, which parallels the dimensionality of the original model weights, $\mathbf{w} \in \mathbb{R}^{m\times n}$. This auxiliary matrix is initialized to zero, ensuring that the initial state of the federated model remains unaffected by the low-rank approximation inherent to our approach. Clients are tasked with updating $\boldsymbol{\sigma}$ through local iterations, applying gradient descent steps with a learning rate η, followed by a truncation step that employs SVD. This truncation selectively retains the dominant components of the weight matrix, thus reducing the volume of data transmitted between clients and the server and addressing one of the primary bottlenecks in federated learning: communication overhead.

In discussing communication overhead, our focus extends beyond merely reducing the volume of data transmitted between clients and the server through compressed updates to enhance communication efficiency. We also consider how our method mitigates overfitting to local data. Overfitting occurs when a model performs too well on training data, losing its ability to generalize to new data. In the context of federated learning, due to data heterogeneity, there is a possibility that the model may over-train on local data from certain clients, leading to decreased performance on the global dataset.

To address this issue, our method involves executing truncated Singular Value Decomposition (SVD) after completing local training iterations but before transmitting local updates to the global server. The transmitted updates consist of the compressed auxiliary matrix $\boldsymbol{\sigma}$ and the local data size, which the server uses to compute the aggregation weight for each client update. This step selectively retains the dominant components of the weight matrix, not only helping to reduce the volume of data needed for communication but also aiding the model in maintaining its generalization capability over local data. Truncated

SVD, by discarding smaller singular values, effectively removes noise and redundant information in the model that could lead to overfitting, thereby focusing the model more on the core structures and patterns of the data. This approach decreases the model's sensitivity to anomalies or noise in individual client data, thus reducing the risk of overfitting on the global model.

Furthermore, during weighted aggregation on the server side, we allocate weights based on the size of the client's dataset, further ensuring that each client's contribution to the global model is proportional to their data volume. This weight distribution mechanism not only fairly reflects the importance of data from each client but also mitigates the impact of data idiosyncrasies from a single client on the global model, helping to enhance the model's generalization performance across diverse client data. The result is an updated global model that embodies the essence of federated learning—collaboratively learning without centralizing data, thereby maintaining privacy and security.

Figure 3.1 illustrates the sequence of operations within our federated learning methodology. The diagram delineates the communication flow between the server and clients, highlighting the local computation, compression step, and the weighted aggregation that results in the global model update.

Algorithm 1 Federated LoRA with Truncated SVD

```
Require: local iteration step $\tau$, learning rate $\eta$
Ensure: Global model $\mathbf{w}(T, 0)$
Initialize: Global model $\mathbf{w}(0, 0)$, $\boldsymbol{\sigma}(0, 0) = \mathbf{0}$
For $t = 0, \ldots, T - 1$ communication rounds do:
    Global server do:
        Select $m$ clients for $S^{(t,0)}$ uniformly at random
        Send $\mathbf{w}^{(t,0)}$, $\boldsymbol{\sigma}(t, 0) = \mathbf{0}$ to client in $S^{(t,0)}$
    Client $k \in S^{(t,0)}$ in parallel do:
        Set $\boldsymbol{\sigma}_i^{(t,0)} = \boldsymbol{\sigma}(t, 0)$
        For $r = 0, \ldots, \tau - 1$ local iterations do:
            $\boldsymbol{\sigma}_i^{(t,r+1)} \leftarrow \boldsymbol{\sigma}_i^{(t,r)} - \eta \nabla f_i(\boldsymbol{\sigma}_i^{(t,r)})$
            Update $\boldsymbol{\sigma}_i^{(t,r+1)} \leftarrow$ truncated SVD$(\boldsymbol{\sigma}_i^{(t,r+1)})$
        EndFor
        Send $\boldsymbol{\sigma}_i^{(t,\tau)}$ and local data size $\|D_i\|$ to the server
    Global server do:
    Update global model with $\mathbf{w}^{(t+1,0)} = \mathbf{w}^{(t,0)} + \sum_{i \in S^{(t,0)}} q_i \boldsymbol{\sigma}_i^{(t,\tau)}$
EndFor
```

where $\boldsymbol{\sigma}$ has the same dimensions as $\mathbf{w}$, initialized as zero.

In Algorithm 1, we formalize the steps of our proposed methodology, delineating the iterative process from initialization to the final global model update. The algorithm elucidates the sequence of operations that begin with the global server disseminating the global model to selected clients and culminate with the aggregation of low-rank updates to form the federated model. The description of each step in the algorithm is meticulously de-

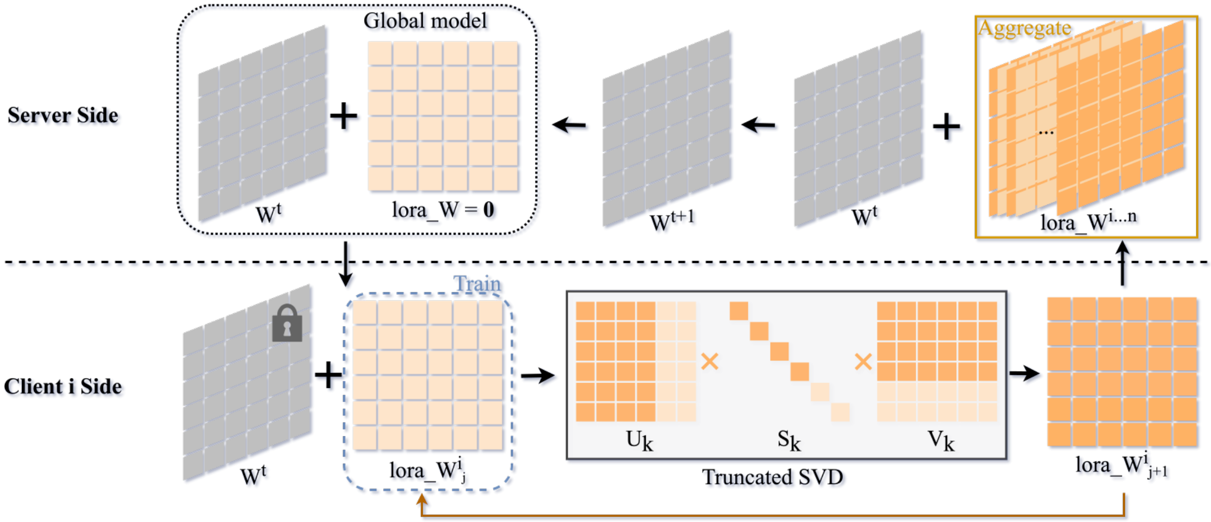

Figure 3.1: Illustration of Method

tailed to facilitate replication and to enhance understanding of the method's application to federated learning scenarios.

The proposed methodology, with its roots in the principles of LoRA, stands as a testament to the adaptability of low-rank matrix approximation techniques in distributed learning environments. The implementation of this methodology is expected to pave the way for more efficient federated learning processes, ultimately contributing to the broader adoption of privacy-preserving machine learning models.

3.0.1 Truncated SVD

Given that $\mathbf{w} \in \mathbb{R}^{m\times n}$ is the original model weight matrix, our goal is to update $\mathbf{w}$ by adding a low-rank adaptation $\boldsymbol{\sigma}$, where $\boldsymbol{\sigma}$ has the same dimensions as $\mathbf{w}$.

- **For fully connected layers** (assumed to be two-dimensional weight matrices)

  Truncated SVD: First, perform a truncated singular value decomposition on $\boldsymbol{\sigma}$, selecting the top k largest singular values ($k < \min(m,n)$), to obtain $\mathbf{w} \approx U_k S_k V_k^T$, where $U_k \in \mathbb{R}^{m\times k}$, $S_k \in \mathbb{R}^{k\times k}$, and $V_k^T \in \mathbb{R}^{k\times n}$. Then, use the truncated decomposition components to reconstruct $\boldsymbol{\sigma}$, which can be achieved by appropriately adjusting and fine-tuning the singular values S_k or the left and right singular vectors U_k, V_k.

- **For convolutional layers** (assumed to be four-dimensional weight matrices)

  Tucker Decomposition: Given a four-dimensional convolutional kernel $\boldsymbol{\sigma} \in \mathbb{R}^{c_{\text{out}}\times c_{\text{in}}\times h\times w}$, perform Tucker decomposition to obtain $\mathbf{w} \approx \mathcal{G} \times_1 A_{\text{mode1}} \times_2 A_{\text{mode2}} \times_3 A_{\text{mode3}} \times_4 A_{\text{mode4}}$, where $\mathcal{G}$ is the core tensor, and A_{modei} are the mode matrices. After that, use the results of the Tucker decomposition to reconstruct $\boldsymbol{\sigma}$, by adjusting the core tensor $\mathcal{G}$ and the mode matrices A_{modei}.

In our approach, integrating truncated Singular Value Decomposition (SVD) for model updates allows us to compress data to reduce communication overhead while also tuning the model's complexity by carefully selecting the number of singular values retained. When we opt to retain all singular values, our method essentially reverts to the Federated Averaging (FedAvg) algorithm, as this means no compression or simplification of model parameters has taken place, and all information in every update round is preserved and transmitted. This aspect reveals the flexibility of our approach, where adjusting the number of singular values retained in the truncated SVD allows us to find a balance between reducing communication costs and maintaining model performance.

More specifically, applying truncated SVD and retaining all singular values does not alter the rank of the weight matrix, hence the model's representational capacity remains unchanged. This equates to transmitting the full model parameters in each round of the federated learning process, operationally consistent with the FedAvg algorithm. In this scenario, our method provides a validation of the standard federated averaging approach, demonstrating that our framework can naturally degrade to FedAvg when no low-rank approximation is introduced.

However, when we choose to truncate and retain only the top k singular values, our method begins to demonstrate its advantages. Through this approach, we can remove redundant information and noise in the model that might lead to overfitting or diminish the model's generalization ability on unseen data. Moreover, by reducing the amount of data that needs to be transmitted, we can significantly lower communication costs, making model updates more efficient. This trade-off provides a mechanism to finely adjust the number of singular values retained to accommodate different communication constraints and data heterogeneity needs.

3.1 Directly combined with LoRA

3.1.1 Introduction to LoRA

LoRA, which stands for Low-Rank Adaptation, is a parameter adaptation technique designed for deep learning models, especially large-scale pre-trained models such as Transformer models. The purpose of LoRA is to enable effective fine-tuning of models without significantly increasing the number of parameters. This method is particularly suitable for scenarios where computational resources are limited.

The core idea of LoRA is to adjust the model efficiently by adapting its weight matrices through low-rank matrices, avoiding direct fine-tuning of all model parameters. Instead of updating the weight matrix $\mathbf{w} \in \mathbb{R}^{m \times n}$ of a linear layer directly, LoRA introduces two smaller matrices $A \in \mathbb{R}^{r \times n}$ and $B \in \mathbb{R}^{m \times r}$, where $r \ll \min(m, n)$ denotes the low rank. These matrices adjust $\mathbf{w}$ indirectly by adapting it through the product of A and B.

Assuming the original weight matrix is $\mathbf{w}$, LoRA adapts it by introducing two low-rank matrices B and A, with dimensions $m \times r$ and $r \times n$ respectively, where r is much smaller than both m and n, indicating a low rank. The adaptation process of LoRA can be expressed as:

$$\mathbf{w}_{\text{adapted}} = \mathbf{w} + \Delta\mathbf{w} = \mathbf{w} + B \times A$$

Here, $\mathbf{w}_{\text{adapted}}$ represents the adapted weight matrix, and $\Delta\mathbf{w} = B \times A$ signifies the adjustment to the original weight matrix $\mathbf{w}$.

In practice, A and B are learned through training data, while $\mathbf{w}$ remains unchanged. Thus, most of the model's parameters (i.e., $\mathbf{w}$) do not need to be updated; only the relatively few parameters in A and B are adjusted to adapt the model, thereby improving the model's adaptability and performance with a minimal increase in parameters.

3.1.2 Federated with LoRA

The integration of Low-Rank Adaptation (LoRA) into the federated learning paradigm offers an innovative mechanism for reconciling the efficiency of model updates with personalized adaptation. The procedure begins with the initialization of the global model and the iterative refinement of low-rank matrices A and B, as outlined in the pseudocode depicted in Algorithm 2. At the commencement of each communication round, the global server selects a subset of clients at random and distributes the global model weights $\mathbf{w}(t, 0)$, along with the initialized low-rank matrices $A(t, 0)$ and $B(t, 0)$.

Each client computes their local model weights by incorporating the received global weights and low-rank matrices. They proceed to perform multiple iterations on their local dataset to update the matrices A and B. After completing the predetermined number of local iterations, the clients transmit the revised low-rank matrices $A^{(\tau)}$ and $B^{(\tau)}$, together with the dataset size, back to the global server. The server conducts a weighted aggregation of these updates to refine the global model weights, scaling each client's contribution by a factor q_i, which may correspond to the size of the client's dataset, as delineated in line 14 of the algorithm.

Figure 3.2 further illuminates the schematic diagram of this integrative process. Throughout each communication round, the global model $\mathbf{w}^t$ is updated by the additive adaptation through the product of low-rank matrices A and B, initially setting B to zero. The clients' computed matrices A_i and B_i are sent back to the server after each local iteration for aggregation. Post-aggregation, the server incorporates these updates into the original global model W^t, resulting in a newly adapted global model W^{t+1}. This strategy significantly reduces the volume of data transferred between clients and the server and enables the model to adapt flexibly to the data characteristics of disparate clients. This is particularly invaluable for federated learning implementations across real-world applications with non-uniform data distributions.

Thus, the amalgamation of LoRA within the federated learning framework facilitates coordinated model updates among clients while adhering to data privacy norms, ensuring the model updating process is both efficient and personalized. The employment of low-rank

Algorithm 2 Federated Update with LoRA

```
Require: local iteration step $\tau$, learning rate $\eta$
Ensure: Global model $\mathbf{w}^{(T,0)}$
Initialize: Global model $\mathbf{w}^{(0,0)}$, $A^{(0,0)}$, $B^{(0,0)} = \mathbf{0}$
For $t = 0, \ldots, T-1$ communication rounds do:
    Global server do:
        Select $m$ clients for $S^{(t,0)}$ uniformly at random
        Send $\mathbf{w}^{(t,0)}$, $A^{(t,0)}$, $B^{(t,0)} = \mathbf{0}$ to client in $S^{(t,0)}$
    Client $k \in S^{(t,0)}$ in parallel do:
        Set $\mathbf{w}_i^{(t,0)} = \mathbf{w}^{(t,0)} + B^{(t,0)} \times A^{(t,0)}$
        For $r = 0, \ldots, \tau - 1$ local iterations do:
            $A_i^{(t,r+1)} = A_i^{(t,r)} - \eta \nabla_B f_i(A_i^{(t,r)}, B_i^{(t,r)})$
            $B_i^{(t,r+1)} = B_i^{(t,r)} - \eta \nabla_A f_i(A_i^{(t,r)}, B_i^{(t,r)})$
        EndFor
        Send $A_i^{(t,\tau)}$, $B_i^{(t,\tau)}$ and local data size $\|D_i\|$ to the server
    Global server do:
    Update global model with $\mathbf{w}^{(t+1,0)} = \mathbf{w}^{(t,0)} + \sum_{i \in S^{(t,0)}} q_i (B_i^{(t,\tau)} \times A_i^{(t,\tau)})$
    or $\mathbf{w}^{(t+1,0)} = \mathbf{w}^{(t,0)} + (\sum_{i \in S^{(t,0)}} q_i B_i^{(t,\tau)}) \times (\sum_{i \in S^{(t,0)}} q_i A_i^{(t,\tau)})$
EndFor
```

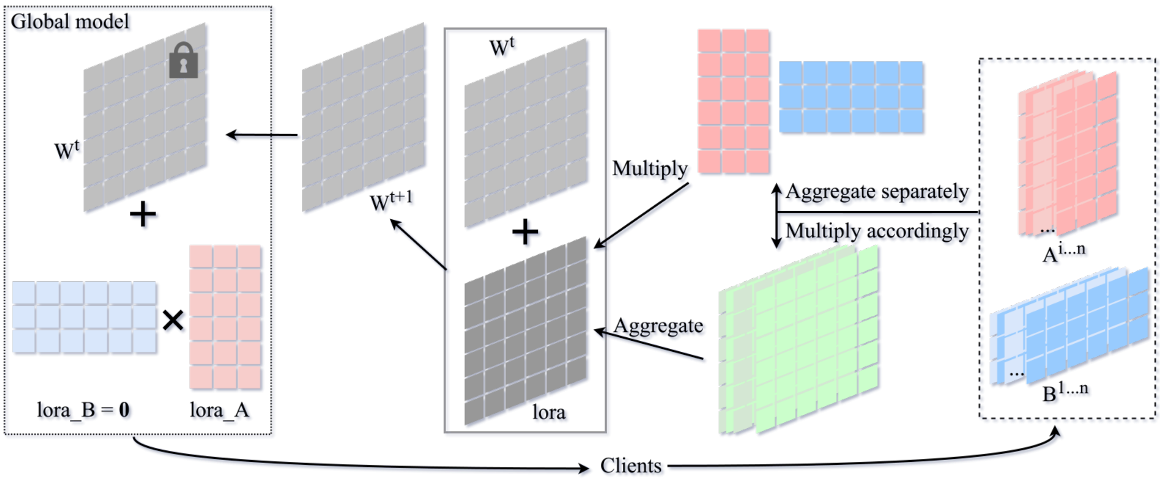

Figure 3.2: Illustration of Method

matrices as a medium for model adaptation alleviates the computational and communicational load, particularly on resource-constrained client devices, hence presenting extensive applicative potential within the distributed machine learning domain.

Despite the initial intent of Low-Rank Adaptation (LoRA) in federated learning to reduce communication overhead without sacrificing model performance, and theoretically, to reduce overfitting by simplifying the model complexity, thereby enhancing the model's generalizability, our experimental results have shown that even when retaining 100% of the model parameters, the federated learning method combined with LoRA (which will be detailed in Chapter 4) does not outperform the traditional Federated Averaging (FedAvg). Moreover, even with a reduction in model size by half, which theoretically should further improve generalization and accuracy, the expected outcomes were not achieved. This suggests a need to re-evaluate the application strategy of LoRA in federated learning and its complex relationship with model size and overfitting. The observed phenomenon may be due to various factors, including the heterogeneity of data distribution, the inadaptability of model update strategies, or potential limitations of low-rank matrices in capturing key features of distributed data. Further research should focus on optimizing the structure and update mechanisms of these low-rank matrices to more accurately reflect and utilize the local data characteristics of clients, ultimately enhancing the overall performance of federated learning models.

Based on the experimental outcomes and in-depth analysis of the LoRA method, we recognized the need for an improved strategy to address the limitations of LoRA's application in federated learning. Therefore, we propose a new method, "LoRA with SVD". Theoretically, when retaining 100% of the model parameters, this approach should not be inferior to the Federated Averaging (FedAvg) method, as in this case, the two are Identity. Moreover, "LoRA with SVD" promises to retain the advantages of the original LoRA method, such as reducing communication overhead and enhancing the model's personalization capabilities. By utilizing SVD, we can capture and compress the model's weight information in a more refined and elegant manner, which not only helps to maintain model performance while reducing storage and transmission size but also theoretically provides a more effective parameter adaptation strategy.

Chapter 4

Experiments

4.1 Experiment Setup

In the experimental setup of our federated learning framework, two distinct datasets, MNIST and CIFAR10. Specifically, for the MNIST dataset, which contains 28x28 grayscale images of handwritten digits, a simpler architecture, MNISTNet, was employed. In contrast, the CIFAR10 dataset, comprising 32x32 color images across 10 different classes, was processed using a Convolutional Neural Network (CNN) model, adept at handling the complexity and spatial hierarchy of high-dimensional image data.

The dataset preparation phase was meticulously designed to emulate a real-world federated learning scenario. We initiated by setting a seed for reproducibility purposes, ensuring identical initialization across runs. The dataset for each experiment was transformed using standard procedures, including tensor conversion and normalization. The CIFAR10 and MNIST datasets were then subdivided into training and testing sets, which were further distributed among 100 simulated clients using a Dirichlet distribution with a specified alpha parameter. This distribution method allowed for creating heterogeneous data partitions among clients, reflecting the diverse data distributions one might encounter in real-world federated learning environments.

For each training round, a subset of clients, precisely 10%, was randomly selected for model training, ensuring each client's dataset's unique characteristics were captured. The model updates from these clients were then aggregated to improve the global model iteratively. In parallel, 20% of clients were randomly chosen for model evaluation, offering insights into the global model's generalization capabilities across different data distributions. This approach highlights the federated learning paradigm's capacity to harness decentralized data while preserving privacy and reducing data centralization needs.

Our federated learning experiments were facilitated by the Flower framework, a flexible tool for developing and deploying federated learning systems. Flower's robustness and adaptability made it an ideal choice for managing the complexities of our experimental setup, including client management, model aggregation, and communication protocols. Through this experimental design, we aimed to explore the nuances of federated learning across different datasets and models, providing valuable insights into the applicability and performance of federated learning strategies in diverse data environments.

In our experimental design, the baseline comparisons were established using two prominent federated learning algorithms: FedAvg and FedProx. To ensure the fairness and integrity of the comparative analysis, both methods were implemented using the Stochastic Gradient Descent (SGD) optimizer across all experiments. A meticulous grid search approach was employed to identify the optimal hyperparameters and learning rates for each algorithm. This rigorous optimization process was essential for neutralizing potential biases in the experimental setup and allowing a fair and direct comparison of the algorithms' performance. By adopting this approach, we aimed to isolate the effects of the federated learning algorithms themselves, rather than the influence of differing hyperparameters or optimization strategies. This methodology underscores our commitment to providing an equitable and comprehensive evaluation of FedAvg and FedProx's effectiveness within our federated learning framework.

4.2 Experiment Results

4.2.1 Federated with LoRA

In our federated learning experiment using the MNIST dataset and the MNISTNet model, we conducted a detailed comparison between the Low-Rank Adaptation (LoRA) method and the conventional Federated Averaging (FedAvg) algorithm. The results were showcased at various local update steps (Epoch 1, Epoch 5, and Epoch 10) and under different levels of data distribution heterogeneity (α values of 0.1 and 0.5). The table 4.1 illustrates that in the initial local update steps, FedAvg demonstrated higher test accuracy under both α settings, indicating its robustness during the early model training phase. However, as the local update steps progressed, the test accuracy of different LoRA variants displayed diversity. Notably, under certain conditions, the 50% LoRA accuracy surpassed that of the 100% LoRA, suggesting that a moderate level of adaptation might reduce overfitting and thereby improve the model's generalizability.

	Epoch 1		Epoch 5		Epoch 10	
Algorithm	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.5$
FedAvg	0.955	0.972	0.959	0.978	0.950	0.979
2% LoRA	0.734	0.912	0.791	0.948		
10% LoRA	0.777	0.953	0.885	0.972	0.851	0.967
50% LoRA	0.841	0.961	0.887	0.974	0.889	0.976
100% LoRA	0.850	0.957	0.924	0.971	0.917	0.971

Table 4.1: Test accuracy for FedAvg and Federated with LoRA algorithms at different epochs and alpha values of MNIST dataset.

Despite incremental improvements across local update steps, the performance of 100% LoRA consistently did not exceed that of FedAvg, indicating limitations within the full adaptation scope of the LoRA method. The performance of the 50% LoRA exceeding that of the 100% LoRA in some settings highlights a significant phenomenon: reduced adaptation not only has the potential to lessen model complexity but also to improve accuracy, potentially due to mitigating overfitting and enhancing generalization.

These experimental findings point to a potential research direction: if the performance of the fully adaptive method (100% LoRA) could at least match that of FedAvg, then smaller models (like 50% LoRA) might not only reduce the number of model parameters but also unexpectedly increase accuracy. This insight provides a theoretical foundation for introducing our next method. With this approach, we aim to find a more efficient balance between model size and accuracy. The practical application of this strategy in model design may help economize on computational resources in federated learning scenarios while potentially enhancing the model's generalization capability across distributed data. Therefore, these results not only highlight the importance of exploring model adaptability within federated learning but also offer a new direction for future research: improving learning outcomes in federated environments by judiciously adjusting model complexity.

4.2.2 Federated LoRA with SVD

In this study, we evaluated the performance of the Singular Value Decomposition Low-Rank Adaptation (SVD-LoRA) method in a federated learning context using the CIFAR10 dataset, comparing it against the standard Federated Averaging (FedAvg) and FedProx algorithms over 500 communication rounds, with each algorithm undergoing a single local epoch.

Table 4.2 reveals that in an extreme non-independent and identically distributed (non-IID) data environment ($\alpha = 0.1$), the 50% SVD-LoRA model outperformed all other models, including the standard FedAvg and FedProx, in test accuracy. As the data distribution heterogeneity decreased (increasing α value), the performance of 50% SVD-LoRA diminished slightly but remained comparable to the fully adapted 100% SVD-LoRA, suggesting that moderate low-rank adaptation helps in reducing model complexity while maintaining accuracy.

Model	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1.0$
FedAvg	0.662	0.729	0.745
FedProx	0.672	0.735	0.742
100% SVD-LoRA	0.661	0.728	0.744
50% SVD-LoRA	0.665	0.723	0.735
20% SVD-LoRA	0.611	0.686	0.689

Table 4.2: Test accuracy for FedAvg and SVD-LoRA models at different values of α with local epoch 1.

Table 4.3 tracks the performance change of the models with increasing local training steps (Epoch 5 and Epoch 10) under the setting of $\alpha = 0.5$. The test accuracy for all models gradually decreased with continued training, which might reflect overfitting in the context of limited local training data. However, 100% SVD-LoRA demonstrated better performance retention at Epoch 10 compared to FedAvg, while 50% SVD-LoRA showed promise in the early training phase (Epoch 5) but experienced a decline in performance after longer training periods (Epoch 10).

Model	Epoch 5	Epoch 10
FedAvg	0.680	0.632
100% SVD-LoRA	0.698	0.647
50% SVD-LoRA	0.676	0.634
20% SVD-LoRA	0.625	0.593

Table 4.3: Test accuracy for FedAvg and different configurations of the SVD-LoRA model at various epochs with $\alpha = 0.5$

Synthesizing these observations, we conclude that SVD-LoRA exhibits significant potential in federated learning environments, particularly in dealing with non-IID data. By adjusting the degree of SVD adaptation, we can find an effective balance between model complexity and accuracy. This offers a new perspective on optimizing federated learning models based on specific data distributions. Notably, the potential of a moderate degree of low-rank adaptation, such as 50% SVD-LoRA, in enhancing model generalizability provides a valuable direction for further research and experimentation.

4.3 Limitations

Despite the innovative approach of integrating truncated Singular Value Decomposition (SVD) and Low-rank Adaption (LoRA) within the federated learning framework, our empirical findings highlight several notable limitations that must be addressed. These limitations not only challenge the practical deployment of our methodology but also provide a valuable direction for future research enhancements.

- **Balancing Model Complexity and Accuracy**
  The primary challenge in applying truncated SVD to federated learning is the difficulty in maintaining an optimal balance between model complexity and accuracy. This balance is particularly problematic in scenarios involving highly heterogeneous data distributions across different clients. Low-rank adaptations tend to underfit when faced with diverse data distributions. This occurs because such adaptations, by design, reduce the model's capacity to capture complex data patterns, leading to a loss in essential details that are critical for model accuracy.

- **Efficiency of Local Iterations**
  An increase in the number of local iterations does not always correlate with improved

performance, contrary to our initial expectation: Excessive local training can cause models to become overly fitted to their specific local data characteristics. This overspecialization restricts the model's ability to generalize well on unseen global data, ultimately diminishing the overall performance across the network.

- **Communication Overheads and Operational Costs**
  While our method aims to reduce communication costs by compressing updates, the actual implementation still incurs considerable overheads: Despite the theoretical reduction in the volume of data to be transmitted, the practical implementation shows that communication still poses a significant bottleneck. This issue is exacerbated when merely increasing the number of local steps, as it does not proportionally decrease the overall operational costs due to the inherent inefficiencies in the communication protocols used.

Chapter 5

Conclusion

This thesis embarked on exploring the efficacy of federated learning (FL) frameworks enhanced by Low-Rank Adaptation (LoRA) and Singular Value Decomposition (SVD) techniques, aiming to tackle the challenges posed by data heterogeneity and communication efficiency. The investigation was rooted in the motivation to harness the burgeoning data generated across decentralized environments without compromising privacy or data integrity, steering towards an innovative FL approach that amalgamates the benefits of LoRA and SVD.

The research meticulously formulated the problem, grounding the need for an advanced FL methodology that could adeptly handle the intricacies of diverse data distributions while minimizing communication overheads. The methodology proposed—integrating truncated SVD with LoRA—emerged as a promising solution, theoretically poised to refine communication efficiency and model adaptability in a federated context.

Experimental validation was conducted across two distinct datasets, MNIST and CIFAR-10, employing models of varying complexities to simulate real-world federated learning scenarios. The experiments were thoughtfully designed to mirror practical FL challenges, including non-IID data distributions and the necessity for efficient communication strategies.

The results revealed a nuanced landscape of FL performance, with the SVD-LoRA methodology showcasing notable potential in addressing non-IID challenges and achieving significant accuracy improvements over traditional FL methods like FedAvg and FedProx, especially in highly heterogeneous data environments. The experiments underscored the importance of optimizing the degree of low-rank adaptation, highlighting that a moderate level of adaptation (50% SVD-LoRA) could effectively balance model complexity and performance, thereby reducing the risk of overfitting and enhancing generalizability.

In conclusion, this thesis contributes a novel perspective to the federated learning domain, proposing an enhanced FL framework that leverages the strengths of LoRA and SVD to improve model performance and communication efficiency. The findings advocate for further exploration into the integration of low-rank matrix techniques within federated learning, suggesting a promising avenue for future research aimed at optimizing distributed machine learning models in an increasingly data-driven world.

5.1 Future Directions

The integration of SVD and LoRA into the federated learning framework has surfaced promising avenues for enhancing model efficiency and communication efficacy. However, the nuanced challenges unearthed through our experimentation reveal significant opportunities for deeper exploration and refinement.

Dynamic rank adaptation emerges as a crucial area for future investigations. Given the static nature of rank settings in our current methodology, adjusting the rank dynamically in response to the varying complexities and characteristics of data across different clients could potentially enhance model performance. This adaptation could leverage real-time performance metrics to optimize the balance between model complexity and the risk of overfitting, employing adaptive algorithms or autonomous machine learning techniques such as reinforcement learning.

Furthermore, the local training protocols warrant a reevaluation. The inefficacy of merely increasing local iteration counts suggests the need for advanced training strategies that mitigate over-specialization to local datasets. Incorporating regularization techniques tailored for federated contexts or alternating training modalities could provide mechanisms to balance local and global learning objectives effectively.

On the communication front, despite reductions in data transmission volumes, the overheads associated with frequent updates underscore the necessity for more sophisticated data compression methods and asynchronous communication protocols. Such strategies could minimize the latency issues and optimize the bandwidth usage, making the federated learning process more seamless and efficient.

In summary, the prospective research directions outlined herein not only highlight the potential for substantial methodological advancements but also set a foundation for making federated learning a more viable and robust solution for real-world applications. These endeavors will undoubtedly propel the domain forward, bridging the gap between theoretical research and practical implementation.

References

[1] Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (general data protection regulation), 2018. European Union.

[2] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in neural information processing systems*, 30, 2017.

[3] Omid Aramoon, Pin-Yu Chen, Gang Qu, and Yuan Tian. Meta-federated learning. In *Federated Learning*, pages 161–179. Elsevier, 2024.

[4] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[5] Muhammad Asad, Ahmed Moustafa, and Takayuki Ito. Fedopt: Towards communication efficiency and privacy preservation in federated learning. *Applied Sciences*, 10(8):2864, 2020.

[6] Haokun Chen, Ahmed Frikha, Denis Krompass, Jindong Gu, and Volker Tresp. Fraug: Tackling federated learning with non-iid features via representation augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4849–4859, 2023.

[7] Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems*, 35(4):83–93, 2020.

[8] Moming Duan, Duo Liu, Xinyuan Ji, Renping Liu, Liang Liang, Xianzhang Chen, and Yujuan Tan. Fedgroup: Efficient federated learning via decomposed similarity-based clustering. In *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 228–237. IEEE, 2021.

[9] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature Medicine*, 25(1):24–29, 2019.

[10] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

[11] Chenyou Fan and Ping Liu. Federated generative adversarial learning. In *Pattern Recognition and Computer Vision: Third Chinese Conference, PRCV 2020, Nanjing, China, October 16–18, 2020, Proceedings, Part III 3*, pages 3–15. Springer, 2020.

[12] Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

[13] Ibrar Yaqoob Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. The rise of “big data” on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, 2015.

[14] Samuel Horvóth, Chen-Yu Ho, Ludovit Horvath, Atal Narayan Sahu, Marco Canini, and Peter Richtárik. Natural compression for distributed deep learning. In *Mathematical and Scientific Machine Learning*, pages 129–141. PMLR, 2022.

[15] Seyyedali Hosseinalipour, Sheikh Shams Azam, Christopher G Brinton, Nicolo Michelusi, Vaneet Aggarwal, David J Love, and Huaiyu Dai. Multi-stage hybrid federated learning over large-scale d2d-enabled fog networks. *IEEE/ACM transactions on networking*, 30(4):1569–1584, 2022.

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[17] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

[18] Michael I. Jordan and Tom M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[19] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[20] Ivan Kholod, Evgeny Yanaki, Dmitry Fomichev, Evgeniy Shalugin, Evgenia Novikova, Evgeny Filippov, and Mats Nordlund. Open-source federated learning frameworks for iot: A comparative review and analysis. *Sensors*, 21(1):167, 2020.

[21] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282, 2017.

[22] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8397–8406, 2022.

[23] Emre Ozfatura, Kerem Ozfatura, and Deniz Gündüz. Time-correlated sparsification for communication-efficient federated learning. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 461–466. IEEE, 2021.

[24] Wullianallur Raghupathi and Viju Raghupathi. Big data analytics in healthcare: promise and potential. *Health Information Science and Systems*, 2(1):3, 2014.

[25] Momina Shaheen, Muhammad Shoaib Farooq, Tariq Umer, and Byung-Seo Kim. Applications of federated learning; taxonomy, challenges, and research trends. *Electronics*, 11(4):670, 2022.

[26] Yiqing Shen, Yuyin Zhou, and Lequan Yu. Cd2-pfed: Cyclic distillation-guided channel decoupling for model personalization in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10041–10050, 2022.

[27] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020.

[28] Akhil Vaid, Suraj K Jaladanki, Jie Xu, Shelly Teng, Arvind Kumar, Samuel Lee, Sulaiman Somani, Ishan Paranjpe, Jessica K De Freitas, Tingyi Wanyan, et al. Federated learning of electronic health records improves mortality prediction in patients hospitalized with covid-19. *MedRxiv*, 2020.

[29] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

[30] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

[31] Hang Xu, Kelly Kostopoulou, Aritra Dutta, Xin Li, Alexandros Ntoulas, and Panos Kalnis. Deepreduce: A sparse-tensor communication framework for federated deep learning. *Advances in Neural Information Processing Systems*, 34:21150–21163, 2021.

[32] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018.

[33] Won Joon Yun, Yunseok Kwak, Hankyul Baek, Soyi Jung, Mingyue Ji, Mehdi Bennis, Jihong Park, and Joongheon Kim. Slimfl: Federated learning with superposition coding over slimmable neural networks. *IEEE/ACM Transactions on Networking*, 2023.

[34] Shen Zhe and Zachary C. Lipton. Deep learning for privacy-preserving data release. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 645–654. ACM, 2018.