Federated Learning & Optimization

Jawad Saeed Muhammad Saad Haroon Daanish Uddin Khan

Abstract

Federated learning has emerged as a transformative approach for distributed training of machine learning models, addressing privacy and communication constraints. This report delves into the theoretical and empirical analysis of six algorithms: Centralized Training, FedSGD, FedAvg, SCAFFOLD, FedGH, and FedSAM, with a focus on handling data heterogeneity, communication efficiency, and convergence rates. Each algorithm is rigorously evaluated under varying levels of client heterogeneity, using the MNIST dataset partitioned via Dirichlet distributions. Key insights include the robustness of SCAFFOLD in mitigating client drift, FedGH's efficacy in resolving gradient conflicts, and FedSAM's superior generalization through sharpness-aware minima. Comparative experiments highlight the critical trade-offs and merits of these algorithms, offering valuable guidelines for federated learning deployment in real-world scenarios.

Project Repository : Link

1. Introduction

Federated learning (FL) represents a paradigm shift in machine learning by enabling decentralized training without sharing raw data, thus preserving privacy and reducing communication costs. However, FL introduces unique challenges, particularly when client data distributions are heterogeneous (non-IID), resulting in client drift and reduced global model performance. Addressing these challenges requires novel algorithms that balance convergence, communication efficiency, and performance under diverse conditions.

In this report, we analyze six prominent algorithms in federated learning: **Centralized Training**, **FedSGD**, **FedAvg**, **SCAFFOLD**, **FedGH**, and **FedSAM**. Each algorithm is evaluated for its ability to handle data heterogeneity, communication efficiency, and convergence rates, with experiments conducted on the MNIST dataset partitioned using Dirichlet distributions. The findings aim to provide actionable insights into the design and deployment of FL systems.

2. Methodology

2.1. Task 1

This task analyzes the theoretical equivalence between Federated Stochastic Gradient Descent (FedSGD) and a centralized training scenario using full-batch gradient descent. The equivalence is explored by tracking the magnitude of gradients across multiple communication rounds for both setups. The following methodology was adopted:

2.1.1. Problem Setup

- **Data Partitioning:** The dataset was partitioned among $M = 5$ clients using a Dirichlet distribution with $\alpha = 0.8$ to simulate varying levels of heterogeneity. Each client was assigned a unique subset of data.
- **Training Configuration:**
  - **Batch Size:** $B = 1$ to minimize floating-point discrepancies.
  - **Optimizer:** Stochastic Gradient Descent (SGD) with a learning rate $\eta = 10^{-4}$.
  - **Model Architecture:** A simple convolutional neural network (`SimpleCNN`).

2.1.2. FedSGD Training Procedure

FedSGD was implemented as follows:

- **Local Updates:** Each client i computes a gradient update Δw_i on its local dataset D_i for one epoch:

$$\Delta w_i = \nabla L(w; D_i),$$

  where $L(w; D_i)$ represents the loss function for client i.

- **Aggregation:** The global model updates are computed by averaging the client gradients:

$$\Delta w_G = \frac{1}{M} \sum_{i=1}^{M} \Delta w_i,$$

  where M is the number of clients. The global model weights are updated as:

$$w_G^{(t+1)} = w_G^{(t)} - \eta \Delta w_G.$$

2.1.3. Centralized Training Procedure

Centralized training aggregates all client datasets and performs full-batch gradient descent:

- **Gradient Calculation:** The centralized gradient is computed as:

$$\Delta w_C = \nabla L(w; D),$$

  where $D = \bigcup_{i=1}^{M} D_i$ is the combined dataset.

- **Parameter Update:** The centralized model weights are updated as:

$$w_C^{(t+1)} = w_C^{(t)} - \eta \Delta w_C.$$

2.1.4. Evaluation Metrics

The primary metric is the gradient magnitude, computed as:

$$GradientMagnitude = \sum_{i=1}^{P} \|\Delta w_i\|_2,$$

where P is the number of parameters in the model. Gradient magnitudes were compared across rounds to identify any divergence.

2.1.5. Code Implementation

The provided Python code implemented the following:

- `fedsgd_training`: Simulates FedSGD by averaging client updates.
- `centralized_training_updates`: Implements centralized training with full-batch gradient descent.
- `Gradient Comparison`: Magnitudes were plotted for both methods to visually inspect equivalence.

2.2. Task 2

This study investigates the performance of *Federated Averaging (FedAvg)* under varying conditions of label heterogeneity. The task comprises two key components: **FedAvg under Varying Levels of Heterogeneity** as the main focus, and **FedAvg with Extreme Heterogeneity** as an exploratory subtask. Below, we describe the dataset preparation, model configurations, and experimental setups for each component.

2.3. Dataset Preparation

The MNIST dataset was used for all experiments, with an 80% training and 20% testing split. Label heterogeneity was simulated using Dirichlet distributions, varying the concentration parameter α to control the degree of skew in the label distribution. For the extreme heterogeneity setup, data was partitioned such that each client received examples from a single class only.

Key Parameters: For the main part of Task 2, training data was distributed among 5 clients, while the exploratory part involved 10 clients. Both components used a batch size of 128, 20 local epochs per client, and a learning rate of 0.001. Cross-entropy loss and the Adam optimizer were used for training.

2.4. FedAvg under Varying Levels of Heterogeneity (Main Task)

The primary focus of this task was to evaluate the impact of different levels of label heterogeneity on FedAvg's performance. We used Dirichlet distributions to partition the data across 5 clients, with α values of $\{2.0, 0.5, 0.1\}$ representing decreasing levels of homogeneity. Each client trained a local model on its assigned dataset, and global updates were aggregated using FedAvg. After each communication round, the global model was evaluated on the centralized test set.

2.5. FedAvg with Extreme Heterogeneity (Exploratory Subtask)

In this scenario, each client was assigned data from only one class, representing an extreme case of label skew. Training data was distributed among 10 clients, with each client training locally on its respective single-class dataset. The global model was updated using FedAvg and evaluated on the centralized test set to analyze its generalization ability across unseen classes.

Assumptions: All clients participated in every communication round. We assumed a stable connection and no client or communication dropout during training.

2.6. Evaluation Metrics

The primary metric for both components was classification accuracy on the centralized test set. For the main task, we compared accuracy trends across different α values to assess the impact of label skew. For the exploratory subtask, we focused on the global model's ability to generalize to multiple classes despite training on highly skewed data.

2.7. Task 3

This task involved the implementation and testing of the **SCAFFOLD** algorithm for varying levels of heterogeneity. The task setup is defined as follows:

- **Data Preparation:** The task required using the provided MNIST dataset, which was configured to produce federated training and testing data loaders for varying degrees of heterogeneity.

- **Model Configuration:** The model configuration remained the same as that of the previous tasks where the provide **SimpleCNN** model was used as both the local and global model.

- **Assumptions:** This task was done under the assumption that the structure of the SCAFFOLD algorithm will be the same as that of FedAVG with the addition of control variates and their update equations.

- **Task Setup:** Same task setup as the previous task where the training and testing suite was called for varying alpha values to observe the algorithm's performance in different scenarios.

2.8. Task 4

This task involved the observation of gradient conflicts that occur in a federating learning scenario alongside the implementation of the **Gradient Harmonization Algorithm (FedGH** for varying levels of alpha.

- **Data Preparation:** The task required using the provided MNIST dataset, which was configured to produce federated training and testing data loaders for varying degrees of heterogeneity.

- **Model Configuration:** The model configuration remained the same as the previous tasks where the provide **SimpleCNN** model was used as both the local and global model.

- **Assumptions:** This task involved updating the FedAVG implementation to count the gradient conflicts and resolve them as the client models were trained in each round.

- **Task Setup:** The task was set up in two different parts.
  - The first part involved observing the gradient conflict using the provided **gradient_conflict_counter** function to see how the value of alpha affects the number of gradient conflicts across rounds.
  - The second part involved the implementation of the FedGH algorithm using the FedAVG algorithm as a base to resolve the gradient conflicts in each round as a result of the local training of all the clients.

2.9. Task 5

This task focuses on implementing the Federated Sharpness-Aware Minimization (FedSAM) algorithm to improve generalization in federated learning under heterogeneous data distributions. FedSAM introduces a perturbation step during local client training to guide the optimization process towards flatter minima, which are associated with better generalization. The methodology is outlined as follows:

2.9.1. FedSAM Algorithm

FedSAM modifies the conventional federated learning pipeline by incorporating sharpness-aware perturbations during local client training. The key steps are described below:

Perturbation Step For a given model w and loss function $L(w)$, the sharpness-aware perturbation is computed as:

$$w' = w + \rho \frac{\nabla L(w)}{\|\nabla L(w)\| + \epsilon},$$

where:

- ρ is the perturbation radius, controlling the extent of perturbation.
- $\nabla L(w)$ is the gradient of the loss function with respect to model parameters.
- ϵ is a small constant to ensure numerical stability.

Re-optimization The perturbed weights w' are then used to compute a new gradient:

$$g' = \nabla L(w'),$$

which is applied to update the model:

$$w = w - \eta g',$$

where η is the learning rate.

2.9.2. Local Training Procedure

The FedSAM algorithm modifies the local training process on each client as follows:

1. Compute the gradient of the loss function at the current model parameters.
2. Apply the SAM perturbation to obtain perturbed weights.
3. Compute the loss and gradient at the perturbed weights.
4. Revert the perturbation and apply the updated gradient to the original model.

The parameter update for client i after E local epochs is given by:

$$\Delta w_i = \frac{1}{E} \sum_{e=1}^{E} \nabla L(w_i'; D_i),$$

where D_i is the local dataset for client i.

2.9.3. Federated Training with FedSAM

The global model is updated by aggregating client updates:

$$\Delta w_G = \frac{1}{M} \sum_{i=1}^{M} \Delta w_i,$$

where M is the number of clients. The global model parameters are then updated as:

$$w_G = w_G + \Delta w_G.$$

2.9.4. Experimental Setup

The experiments were conducted under the following settings:

- **Dataset:** The MNIST dataset was partitioned among $M = 5$ clients using a Dirichlet distribution to simulate varying degrees of data heterogeneity. Three values of the Dirichlet parameter α ($\alpha = 2.0, 0.5, 0.1$) were used.
- **Model:** A simple convolutional neural network (`SimpleCNN`) was used as the local and global model.
- **Hyperparameters:**
  - Number of communication rounds: $R = 3$
  - Local epochs: $E = 20$
  - Learning rate: $\eta = 0.001$
  - Batch size: $B = 128$
  - Perturbation radius: $\rho = 0.001$
- **Sharpness Analysis:** To analyze the impact of FedSAM, the following experiments were conducted:
  - Comparison of accuracy across different Dirichlet parameters ($\alpha = 2.0, 0.5, 0.1$).
  - Analysis of the effect of varying ρ values ($\rho = 0.0025, 0.005, 0.0075$) on model accuracy.

2.9.5. Performance Metrics

The following metrics were used to evaluate the effectiveness of FedSAM:

- **Model Accuracy:** Measured on a centralized test set after each communication round.
- **Sharpness Measures:** The loss landscape was analyzed using three sharpness measures:
  1. **First-Order Approximation:** Gradient norm-based measure as described in Task 3.
  2. **Random-Directional Sharpness:** Maximum change in loss under random perturbations.
  3. **Hessian-Based Sharpness:** Trace of the Hessian matrix, approximated using Hutchinson's method:

$$Tr(\nabla^2 L(w)) \approx E_v \left[v^\top (\nabla^2 L(w)) v \right].$$

3. Results

3.1. Task 1

The gradient magnitudes across rounds for both FedSGD and centralized training are shown in Figure 1.

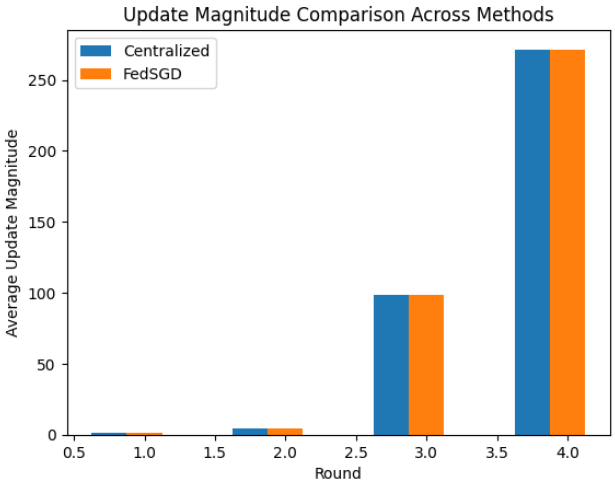

Figure 1. Comparison of Gradient Magnitudes for FedSGD and Centralized Training Across Rounds. Both methods show near-identical gradient magnitudes, confirming theoretical equivalence.

- **Observation:** As shown in Figure 1, the average gradient magnitudes for FedSGD and centralized training are nearly identical across all communication rounds.
- **Key Results:**
  - At Round 1, both methods recorded a gradient magnitude of approximately 0.5.
  - By Round 4, the gradient magnitudes reached a value of approximately 250 for both methods.
- **Conclusion:** The results confirm that FedSGD and centralized training yield equivalent gradient updates under the provided setup, as expected theoretically.

3.2. Task 2

This section presents the results for **FedAvg under Varying Levels of Heterogeneity** and the exploratory **FedAvg with Extreme Heterogeneity**. Key findings are illustrated using both visualizations and summary tables.

3.3. FedAvg under Varying Levels of Heterogeneity

We varied the Dirichlet parameter α across $\{2.0, 0.5, 0.1\}$ to simulate different levels of label heterogeneity. Figure 2 shows the progression of total accuracy across 3 communication rounds for each α. Additionally, Figure 3 provides a detailed view of client-specific accuracies across rounds for the different α values.

Observations: - For $\alpha = 2.0$, representing a near-uniform data distribution, accuracy increased steadily to a final value of **59.13%**. - As α decreased (higher skew), performance dropped significantly. For $\alpha = 0.5$, the final accuracy was **60.23%**, while for $\alpha = 0.1$, it was **38.77%**. - **Higher heterogeneity levels reduced model generalization**, as evident from lower accuracies for smaller α values. - Client-specific accuracies (Figure 3) show greater variability as α decreases, highlighting the uneven learning across clients in highly skewed scenarios.

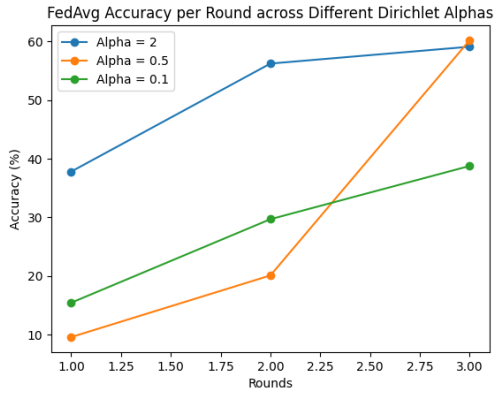

Figure 2. Task 2: FedAvg Accuracy per Round across Different Dirichlet Alphas. Higher α values correspond to less label skew.

3.4. FedAvg with Extreme Heterogeneity

The exploratory subtask demonstrates the severe limitations of FedAvg in scenarios with extreme non-IID data. Table 3.4 highlights the following: - Certain clients, such as Client 1, achieved **100%** accuracy on their local datasets, reflecting overfitting to their single-class data. - However, most clients failed to learn meaningful representations, resulting in near-zero accuracy for many classes. - The global model's overall accuracy remained consistently low, peaking at **12.97%** in Round 1 and declining to **11.07%** by Round 3.

Key Insight: In extreme heterogeneous setups, FedAvg fails to aggregate meaningful updates due to conflicting contributions from local models. The global model struggles to generalize beyond the biased and disjoint client data.

Client ID	Round 1	Round 2	Round 3
0	0.00	0.17	0.50
1	77.36	99.37	100.00
2	0.00	0.00	0.00
3	0.00	0.00	0.32
4	0.00	0.00	0.00
5	51.87	9.30	4.19
6	0.00	0.00	0.00
7	0.00	0.00	0.00
8	0.00	0.00	0.00
9	0.35	0.26	0.00
Total	**12.97**	**11.43**	**11.07**

Table 2. Task 2: Client and Total Accuracy (%) for Extreme Heterogeneity Scenario.

3.5. Task 3

Following the implementation of the **SCAFFOLD** algorithm, it was tested across varying levels of heterogeneity to analyze the robustness of the algorithm. Figure 4 illustrates the results we obtained from testing.

From the graph, it is visible that following three rounds of communication, the algorithm achieves greater accuracies for all levels of Alpha as compared to **FedAVG**. The difference in performance is particularly noticeable at lower alpha values. For example, at $\alpha = 0.1$, FedAVG achieves an accuracy of **39.70%** following three rounds of communication. In contrast, for the same experimental settings, SCAFFOLD exhibits an accuracy of **64.87%**, representing a 24% in accuracy for the two algorithms.

Similar results are observed for $\alpha = 0.5$ and $\alpha = 0.2$ where SCAFFOLD outperforms achieving accuracies of **81.55%** and **85.38%** respectively.

3.6. Task 4

This task involved the implemented of the **Gradient Harmonization** algorithm (FedGH) and testing it for varying levels of heterogeneity. Figure 5 illustrates the number of gradient conflicts across all layers as we vary the α parameter. From the figure it is visible that as the heterogeneity of data increases the number gradient conflicts also increases.

Following the implementation of the **FedGH** algorithm we visualized the number of gradient conflicts pre-harmonization and post-harmonization to if the algorithm

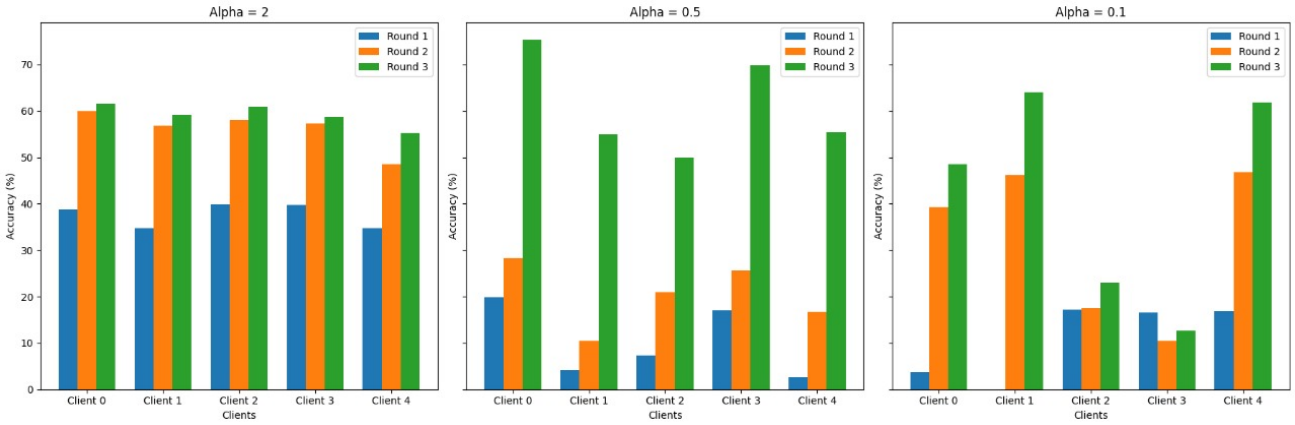

Figure 3. Task 2: Client-specific accuracy per round for $\alpha = 2.0$, $\alpha = 0.5$, and $\alpha = 0.1$. The variability in performance increases as α decreases, indicating higher heterogeneity.

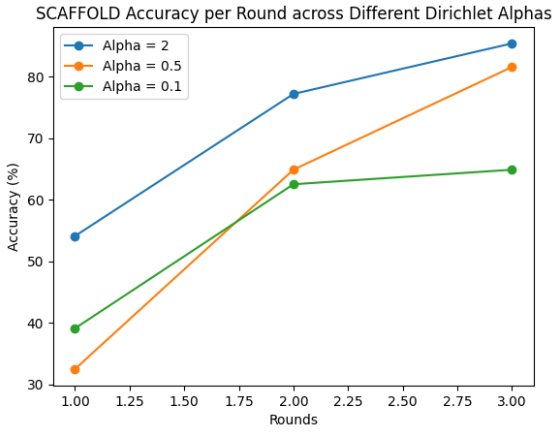

Figure 4. SCAFFOLD Accuracies for varying degrees of heterogeneity

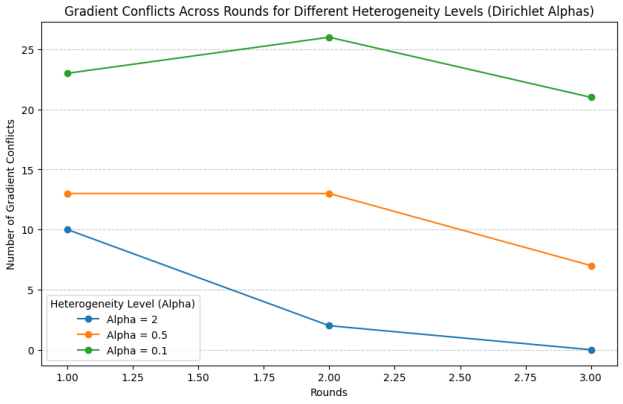

Figure 5. Gradient Conflicts for varying degrees of heterogeneity

was effective. Fig 6 illustrates these results.

From Figure 6 it is visible that the algorithm is effective in mitigating conflicts at each round which is visible by the lower number of conflicts across all all the rounds post-harmonization.

Lastly, Figure 7 illustrates the model accuracies across the varying levels of α. The trend indicates that overall the accuracy increases across all rounds or communication, and greater accuracy is observed in scenarios with lesser heterogeneity.

3.7. Task 5

The results for Task 5 are detailed below, showcasing the performance and sharpness analysis of FedSAM under varying experimental conditions. The findings are presented in the order of graphs provided.

3.7.1. FedSAM Performance Across Varying Levels of Heterogeneity

Figure 8 illustrates the accuracy of FedSAM across communication rounds for varying Dirichlet parameters ($\alpha = 2.0, 0.5, 0.1$). The following observations were made:

- Higher α values (e.g., $\alpha = 2.0$) led to more uniform data distributions, achieving higher accuracy and faster convergence.
- Lower α values (e.g., $\alpha = 0.1$) simulated higher data heterogeneity, resulting in slower convergence, but FedSAM still demonstrated significant improvements across rounds.

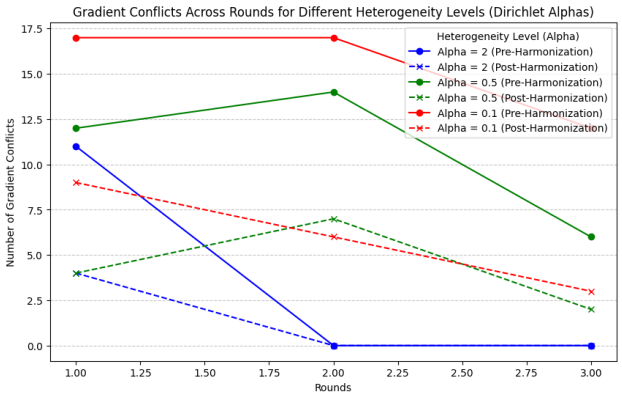

Figure 6. Pre-harmonization and Post-harmonization conflicts for varying degrees of heterogeneity

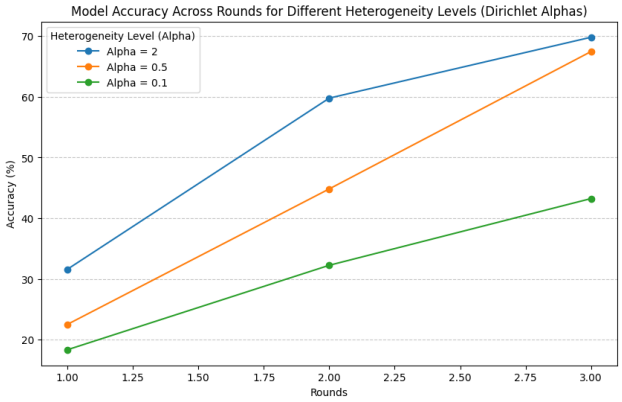

Figure 7. FedGH accuracies for varying degrees of heterogeneity

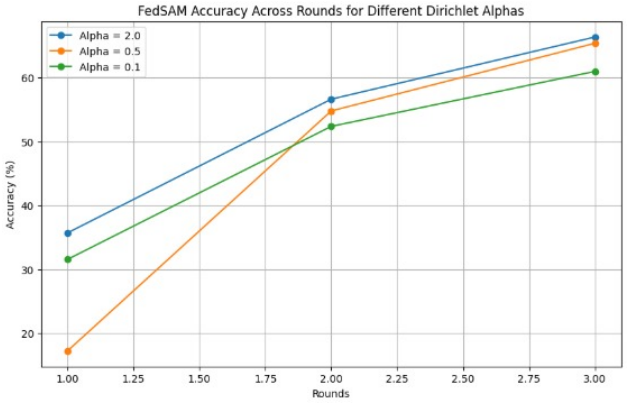

Figure 8. FedSAM Accuracy Across Rounds for Different Dirichlet Alphas.

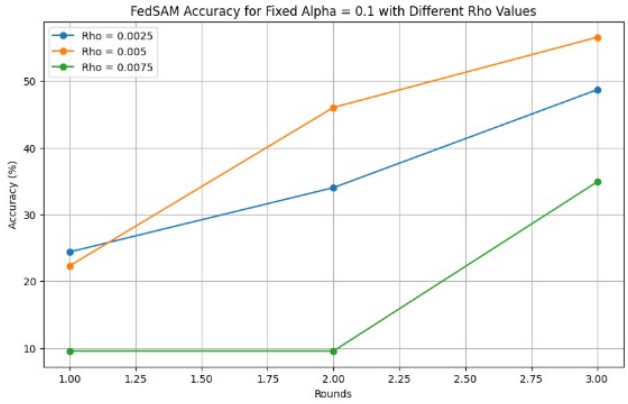

Figure 9. FedSAM Accuracy for Fixed $\alpha = 0.1$ with Different ρ Values.

3.7.2. FedSAM Accuracy for Varying Perturbation Radii (ρ)

Figure 9 shows the effect of varying perturbation radii ($\rho = 0.0025, 0.005, 0.0075$) on FedSAM performance for $\alpha = 0.1$. Key insights include:

- An optimal ρ value of 0.005 yielded the best performance, achieving the highest accuracy after three rounds.
- Larger ρ values (e.g., $\rho = 0.0075$) caused over-perturbation, slowing down convergence and leading to reduced final accuracy.
- Smaller ρ values (e.g., $\rho = 0.0025$) resulted in suboptimal sharpness-aware adjustments, limiting accuracy improvements.

3.7.3. FedSAM vs. FedAvg Performance

Figure 10 compares the performance of FedSAM and FedAvg across communication rounds for varying Dirichlet parameters. Key findings include:

- FedSAM consistently outperformed FedAvg across all α values, demonstrating its ability to handle data heterogeneity more effectively.
- The performance gap was most pronounced for $\alpha = 0.1$, with FedSAM achieving a final accuracy of 60.99% compared to 38.77% for FedAvg.
- For more homogeneous data distributions ($\alpha = 2.0$), FedSAM and FedAvg showed similar trends, but FedSAM still achieved a marginally higher accuracy.

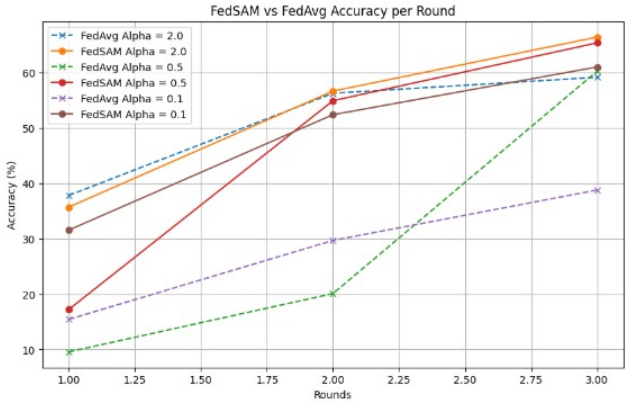

Figure 10. FedSAM vs. FedAvg Accuracy per Round for Different Dirichlet Alphas.

3.7.4. Sharpness Measure Comparisons

Figure 11 provides a comparison of the normalized sharpness values for three measures: First-Order Approximation, Random-Directional Sharpness, and Hessian-Trace Sharpness. Key observations include:

- The **First-Order Approximation** measure showed a steady decline, indicating reduced sensitivity to gradient perturbations over rounds.
- The **Random-Directional Sharpness** measure initially decreased, followed by an increase, reflecting dynamic changes in the loss landscape.
- The **Hessian-Trace Sharpness** measure exhibited an increasing trend, highlighting growing curvature in the loss surface as training progressed.

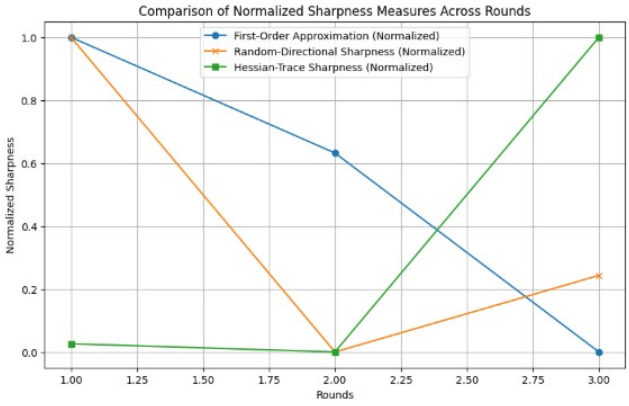

Figure 11. Comparison of Normalized Sharpness Measures Across Rounds.

4. Discussion

4.1. Task 1

4.1.1. Theoretical Equivalence

The experimental results confirm the theoretical equivalence between FedSGD and centralized training under ideal conditions. Both methods calculate gradients over the same underlying dataset, leveraging the linearity of gradient computation:

$$\nabla L(w; D) = \sum_{i=1}^{M} \nabla L(w; D_i),$$

where $D = \bigcup_{i=1}^{M} D_i$ represents the aggregated dataset.

4.1.2. Observed Erratic Behavior

Despite the theoretical equivalence, this task exhibited unpredictable behavior across multiple re-runs. In some runs, the gradient magnitudes for FedSGD and centralized training aligned closely, while in others, discrepancies were observed. This inconsistency stems from the data distribution process, which introduces variability during each execution. Specifically:

- **Dirichlet Sampling:** Data is distributed among clients using a Dirichlet distribution. This process assigns varying proportions of each class to clients based on the α parameter.
- **Client Data Diversity:** As the Dirichlet distribution generates new splits in each re-run, the heterogeneity of client datasets differs, directly influencing the local gradients computed during FedSGD.
- **Impact on Gradients:** Variability in data distributions causes differences in the local objective functions $L(w; D_i)$ for each client, leading to potential gradient divergence between FedSGD and centralized training.

4.1.3. Insights from Data Partitioning

The data partitioning process, as shown below, demonstrates the inherent randomness introduced by Dirichlet sampling:

- Each class's indices are split among clients based on random proportions sampled from a Dirichlet distribution with $\alpha = 0.8$.
- This process ensures diverse data distributions but leads to inconsistency across runs as proportions and splits vary.

For instance, given the same dataset, the Dirichlet-based partitioning may result in one client receiving a majority of a specific class in one run but a more balanced distribution

in another. This variability directly affects the computed updates.

4.1.4. Key Insights

- **FedSGD Robustness:** While FedSGD achieves theoretical equivalence in idealized scenarios, its robustness depends heavily on consistent data distributions across runs.
- **Practical Implications:** In real-world applications, where client data distributions are fixed, such variability may not occur. However, it highlights the importance of carefully managing client data splits during simulation-based studies.

4.1.5. Limitations and Future Considerations

- The observed variability underscores the importance of controlling randomness in federated learning simulations, such as fixing seeds during Dirichlet sampling.
- In practical scenarios, where client data distributions are often static, such variability is unlikely to arise. However, further studies should explore the sensitivity of FedSGD to differing levels of heterogeneity in real-world datasets.

4.2. Task 2

The results from Task 2 reveal significant insights into the behavior of FedAvg under varying levels of heterogeneity and extreme heterogeneity scenarios. Additionally, the theoretical discussion on permutation invariance provides important considerations for the practical application of FedAvg.

4.3. FedAvg under Varying Levels of Heterogeneity

The primary findings for this task are visualized in Figure 2 and summarized in Table **??**. As α decreases, representing higher levels of label skew, the global model's ability to generalize is significantly reduced. For example: - At $\alpha = 2.0$, the final accuracy reached **59.13%**, indicating good generalization under near-uniform data distribution. - At $\alpha = 0.1$, the final accuracy dropped to **38.77%**, reflecting poor performance in the presence of extreme heterogeneity.

Key Insight: The performance of FedAvg is directly correlated with the level of heterogeneity in client data. More homogeneous distributions (higher α) lead to better global models, while higher heterogeneity (lower α) creates challenges in aggregating complementary updates.

4.4. FedAvg with Extreme Heterogeneity

The exploratory subtask demonstrates the severe limitations of FedAvg in scenarios with extreme non-IID data. Table 3.4 highlights the following: - Certain clients, such as Client 1, achieved **100%** accuracy on their local datasets, reflecting overfitting to their single-class data. - However, most clients failed to learn meaningful representations, resulting in near-zero accuracy for many classes. - The global model's overall accuracy remained consistently low, peaking at **12.97%** in Round 1 and declining to **11.07%** by Round 3.

Key Insight: In extreme heterogeneous setups, FedAvg fails to aggregate meaningful updates due to conflicting contributions from local models. The global model struggles to generalize beyond the biased and disjoint client data.

4.5. Theoretical Discussion: Permutation Invariance in FedAvg

The debate between Yann LeCun and Mustafa Siddiqui highlights the theoretical concern of *permutation invariance* in neural networks and its potential impact on FedAvg. Yann argues that weight averaging may yield meaningless results if corresponding parameters are misaligned due to permutation invariance. While this is a valid concern, Mustafa counters with key practical considerations that mitigate the issue.

Key Points from Mustafa's Rebuttal: 1. *Shared Initialization:* All clients start with the same global model, ensuring that their parameter spaces are aligned initially. 2. *Gradient Descent:* Gradients align learned features across clients during training, reducing the impact of permutation invariance. 3. *Empirical Success:* FedAvg has been shown to work effectively in practice, even under moderate non-IID conditions.

Conclusion: While permutation invariance is a valid theoretical limitation, it is largely mitigated in practice due to shared initialization and the aligning effect of gradient descent. These factors enable FedAvg to empirically perform well in many real-world scenarios.

4.6. Overall Takeaways

From the experimental results and theoretical analysis, the following conclusions can be drawn: 1. FedAvg performs well in moderately heterogeneous setups but struggles in extreme heterogeneity, where client updates conflict. 2. Local models in extreme heterogeneity learn class-specific features but fail to generalize, leading to poor global model performance. 3. Theoretical concerns such as permutation invariance are important to consider but are mitigated in practical implementations of FedAvg.

4.7. Task 3

Based on the results observed in 4, we can infer the following:

- As the α value increases, the overall accuracy increases, which is visible from the increase observed going from values of 0.1 to 2. This is because higher values of **Dirichlet Alpha** result in less heterogeneous settings where the data is distributed more evenly among the clients. Due to even distribution, each client tends to perform better, and hence, the overall accuracy increases.
- At lower values of alpha, we simulate a more realistic federated learning scenario where each client mimics a different distribution. Even in such scenarios we can see that SCAFFOLD performs well indicating that the control variate correction tends to optimize the global loss objective instead of the local objective losses of the clients.

4.7.1. Theoretical Answers:

Suppose someone modified the SCAFFOLD algorithm so that, at the start of each communication round, each client sets its local control variate c_i equal to the global control variate c received from the server (i.e., $c_i = c$), instead of maintaining a client-specific c_i across rounds. How would this modified version of SCAFFOLD compare to FedAvg?

Based on the modifications suggested to the SCAFFOLD algorithm, we propose that the following will happen:

Since the global and the local control variates are the same, the equation for adjusting the model parameters becomes,

$$w_i \leftarrow w_i - \eta \left(g_i(w_i) + c - c\right)$$

We can see that this reduces to the equation for gradient descent as follows,

$$w_i \leftarrow w_i - \eta \left(g_i(w_i)\right)$$

Now, since the variates are the same after the local epochs, the control variate update equation at the client reduces to,

$$c_i^+ \leftarrow \frac{1}{K\eta}(w_G - w_i)$$

Examining the equations, this reveals that the training loop is essentially gradient descent, which is the same as that of FedAVG.

The only difference that is visible is the update of local client variate which is essentially updated based on a scaled difference between the initial global model and the updated local model.

The following describes what these modifications achieve:

- **Reduction of Client Drift:** In the original SCAFFOLD algorithm, c_i^+ helps counteract the client drift caused by non-IID data distributions. However, in the modified version, where $c_i = c$, this correction loses its client-specific nature.
- **Scaling of Updates**: The update still scales the discrepancy between w_i and w_G, but this becomes more of a global adjustment rather than a personalized correction.

4.8. Task 4

Based on the results, in Fig 5 we can see that an increase in heterogeneity corresponds to increased gradient conflicts throughout the client models. This makes sense as in such settings the clients tend to overfit on their local data which results in a **local drift**. Since each client focuses on the minimization of its own local loss, this results in conflicting gradients in an increased non-IID scenario.

Based on the results in Fig 6 we can see that FedGH decreases gradient conflicts indicating its effectiveness. Across the three rounds it is visible that at each stage the number of conflicts reduces resulting an an overall decrease in the number of conflicts. However the conflicts are not always zero which can be due to a number of reasons as follows:

- **Incomplete Conflict Resolution:** FedGH reduces conflicts by aligning gradients, but the process is not perfect. Some residual misalignment remains due to the complexity of balancing global and local objectives.
- **Intrinsic Heterogeneity:** In scenarios with highly heterogeneous client data distributions (e.g., low Dirichlet α values), clients have fundamentally conflicting objectives, making complete resolution difficult.
- **Gradient Magnitude Variability:** Clients with larger gradients may dominate the harmonization process, leaving weaker gradients unresolved. Noise in gradients can also contribute to residual conflicts.
- **Suboptimal Harmonization Process:** Limited steps or suboptimal hyperparameters in the harmonization process may prevent full resolution of conflicts during each round.
- **Sequential Aggregation Effects:** Gradients from earlier rounds may introduce cumulative inconsistencies, causing conflicts to persist across rounds despite harmonization.
- **Non-Convex Global Objective:** The global federated learning objective is non-convex, and conflicts are a symptom of the optimization complexity that FedGH cannot entirely eliminate.

4.8.1. THEORETICAL ANSWERS:

1. Is it guaranteed that this new, harmonized gradient direction will lead to a minimization step towards the client's local objective? Or could it possibly increase the local objective's loss for some clients?

No, the harmonized gradient does not guarantee a minimization step towards each client's local objective. The primary goal of gradient harmonization is to reduce conflicts across client updates and steer the global model toward a consensus direction that balances the contributions of all clients. This may not always align with minimizing the local objective for every individual client. Since FedGH is a problem that focuses on global optimization instead of local optimization it is very likely that the local objective loss for some clients increases in order to satisfy the requirements of the global objective loss.

The following are two possible explanation for why this might occur:

- **Gradient Alteration:** The projection step modifies the gradient direction for conflicting clients. This might reduce the component of the gradient that minimizes the local objective for a specific client in favor of improving alignment with other clients.
- **Trade-off Between Clients:** Federated learning involves a trade-off between optimizing individual client objectives and the global objective. By harmonizing gradients, FedGH prioritizes the global objective over local objectives, which could lead to suboptimal updates for some clients.

Thus, while harmonization aims to improve overall convergence and mitigate gradient conflicts, it might increase the local loss for some clients in exchange for better global performance.

2. Consider an extreme scenario where each client's data distribution is vastly different, causing each client to learn a highly unique model during local training. If all clients' gradients are heavily conflicted, how might this affect the magnitude of the final harmonized gradient? Do you think it would be large, small, or unpredictable, and what implications might this have on the effectiveness of the global model update?

In scenarios with highly non-IID data, where clients' data distributions differ significantly, the harmonized gradient may become small or unpredictable in magnitude. The following explains reasons for this:

- **Opposing Directions:** If most client gradients conflict (i.e., they point in nearly opposite directions), the projection step will remove large components of the gradients to align them. This can result in a reduction of magnitude, as much of the conflicting information is discarded.
- **Loss of Information:** The harmonized gradient represents a compromise and may fail to capture strong signals from any individual client. This can lead to slower convergence or less effective updates to the global model.

Implications for the global model:

- **Smaller Gradient Magnitude:** The global update may become ineffectively small, slowing down convergence. In extreme cases, updates could stall if the harmonization excessively suppresses gradient magnitude.
- **Reduced Adaptability:** In diverse datasets, the global model might fail to generalize well to all client distributions because it suppresses significant gradients from individual clients.
- **Potential for Bias:** Clients with gradients that align with the majority might dominate the update, while outlier clients contribute less. This could lead to a global model biased toward clients with more similar distributions.

4.9. Task 5

The results of Task 5 reveal critical insights into the performance of FedSAM, its comparison with FedAvg, and the analysis of sharpness measures in Federated Learning.

4.10. Conclusions and Key Insights

- **FedSAM Performance Across Varying Levels of Heterogeneity:**
  FedSAM demonstrated robust performance under different levels of data heterogeneity, as shown by its consistent accuracy improvements across rounds for all Dirichlet alphas. The method's ability to achieve higher accuracy in highly heterogeneous setups ($\alpha = 0.1$) highlights its effectiveness in mitigating the challenges posed by non-IID data distributions.
- **FedSAM Accuracy for Varying Perturbation Radii (ρ):**
  The perturbation radius ρ was a critical hyperparameter in determining the balance between effective sharpness-aware adjustments and over-perturbation. The optimal $\rho = 0.005$ provided the best trade-off, with excessive or insufficient perturbation radii leading to degraded performance.

- **FedSAM vs. FedAvg Comparison:**
  FedSAM consistently outperformed FedAvg in all heterogeneity settings, particularly for highly non-IID distributions ($\alpha = 0.1$), where it achieved a significant improvement of 22.22% in final accuracy. This performance gap underscores FedSAM's ability to enhance model generalization by effectively navigating the sharpness of the loss landscape.
- **Sharpness Measure Comparison:**
  The comparison of sharpness measures revealed key insights into the relationship between loss landscape flatness and model performance:
  - The **First-Order Approximation**, while computationally efficient, lacked precision in capturing the curvature of the loss landscape.
  - The **Random-Directional Sharpness** provided a robust middle ground by balancing computational feasibility and accuracy.
  - The **Hessian-Based Sharpness**, although computationally expensive, offered the most precise measure of curvature and flatness, making it ideal for theoretical insights.

4.11. Answers to Discussion Questions

Q1: Why might a flatter minimum lead to a more generalizable solution than a sharp minimum?
A flatter minimum indicates that the loss function is relatively stable within a neighborhood around the minimum. This stability implies the model is less sensitive to small perturbations in the input data or weights. The following points highlight why flatter minima improve generalization:

- **Robustness to Variations:** Flatter minima reduce the risk of overfitting, enabling the model to learn features that generalize well to unseen data.
- **Lower Sensitivity to Noise:** Models at sharp minima are highly sensitive to noise in input or optimization, leading to poor generalization on new datasets.
- **Gradient Landscape:** In flatter regions, gradients are less steep, enabling stable performance on unseen distributions.

Q2: In what ways does this property help address the challenges of data heterogeneity in federated learning?
Data heterogeneity in federated learning arises when client data distributions are not identical (non-IID). Flatter minima help address this challenge as follows:

- **Mitigates Local Bias:** Flatter minima prevent overfitting to biased client-specific data distributions, resulting in a more robust global model.
- **Improved Aggregation Stability:** Stable minima ensure that updates from heterogeneous clients do not cause large fluctuations in global model weights, enhancing convergence.
- **Better Generalization Across Clients:** Flatter minima promote solutions that perform well across diverse client distributions, ensuring higher average accuracy.
- **Reduced Sensitivity to Local Perturbations:** Heterogeneous data leads to conflicting gradients. Flatter regions reduce the effects of these conflicts, promoting smoother global updates.

Q3: Explore an Alternate Measure of Sharpness and Compare it with the First-Order Approximation
Three sharpness measures were implemented to evaluate the loss landscape:

- **First-Order Approximation:** Computationally efficient but lacks precision in capturing loss curvature.
- **Random-Directional Sharpness:** Perturbs model weights in random directions, offering a robust estimate that balances cost and accuracy.
- **Hessian-Based Sharpness:** Most precise, capturing second-order curvature, but computationally expensive. Hutchinson's approximation was used to estimate the trace of the Hessian efficiently.

The results showed that random-directional sharpness is practical for balancing computational feasibility and accuracy, while Hessian-based methods provide deeper theoretical insights.

4.12. Task 6

This section presents a comparative analysis of the algorithms implemented in this assignment—**Centralized Training**, **FedSGD**, **FedAvg**, **SCAFFOLD**, **FedGH**, and **FedSAM**. The comparison focuses on three critical criteria: handling data heterogeneity, communication efficiency, and convergence rates. Insights are drawn from empirical observations and theoretical understanding.

4.13. Handling Increased Levels of Data Heterogeneity

Federated learning often encounters non-IID data across clients, leading to challenges like client drift and reduced global model performance.

- **Centralized Training**:
  - **Merit**: Avoids the problem of data heterogeneity as all data is centralized, ensuring uniformity in training.

Dirichlet Alpha	Round 1	Round 2	Round 3
2.0	37.81	56.25	59.13
0.5	9.56	20.08	60.23
0.1	15.43	29.68	38.77

Table 3. FedAvg Accuracy (%) for Different Dirichlet Alphas Across Rounds

Dirichlet Alpha	Round 1	Round 2	Round 3
2.0	35.70	56.64	66.38
0.5	17.22	54.90	65.35
0.1	31.55	52.38	60.99

Table 4. FedSAM Accuracy (%) for Different Dirichlet Alphas Across Rounds

Dirichlet Alpha	Round 1	Round 2	Round 3
2.0	54.09	77.18	85.38
0.5	32.48	64.88	81.55
0.1	39.10	62.51	64.87

Table 5. SCAFFOLD Accuracy (%) for Different Dirichlet Alphas Across Rounds

Dirichlet Alpha	Round 1	Round 2	Round 3
2.0	31.61	59.80	69.53
0.5	22.53	44.84	67.47
0.1	18.34	32.26	43.25

Table 6. FedGH Accuracy (%) for Different Dirichlet Alphas Across Rounds

  - **Demerit**: Impractical in privacy-sensitive scenarios where data cannot be shared or aggregated.

- **FedSGD**:
  - **Merit**: Achieves equivalence to centralized training under ideal conditions (e.g., single gradient step per communication round). Performs poorly in heterogeneous settings due to client drift.
  - **Demerit**: Ineffective in handling heterogeneity with multiple local updates.

- **FedAvg**:
  - **Merit**: Handles mild heterogeneity better than FedSGD by averaging local updates after multiple local steps.
  - **Demerit**: Suffers as heterogeneity increases due to client drift and weight permutation issues.

- **SCAFFOLD**:
  - **Merit**: Addresses client drift explicitly through control variates, improving performance in heterogeneous settings.
  - **Demerit**: Adds computational and storage overhead by maintaining control variates.

- **FedGH**:
  - **Merit**: Resolves gradient conflicts among clients, leading to effective updates in highly heterogeneous scenarios.
  - **Demerit**: Reduces client-specific learning, potentially affecting local objectives.

- **FedSAM**:
  - **Merit**: Ensures better generalization through flatter minima, even in heterogeneous settings.
  - **Demerit**: Increases local training complexity due to perturbation and evaluation steps.

4.14. Communication Efficiency

Efficient communication is vital for scalable federated learning systems.

- **Centralized Training**:
  - **Merit**: No communication overhead in a federated sense.
  - **Demerit**: Not feasible for distributed privacy-preserving setups.

- **FedSGD**:
  - **Merit**: Minimal local computation, but requires frequent server communication.
  - **Demerit**: Highly communication-intensive and inefficient for large-scale scenarios.

- **FedAvg**:
  - **Merit**: Reduces communication by performing multiple local updates before synchronization.
  - **Demerit**: Efficiency gains are limited under severe heterogeneity.

- **SCAFFOLD**:
  - **Merit**: Balances communication via control variates, reducing misaligned updates.
  - **Demerit**: Slightly increases communication overhead due to variates.

- **FedGH**:
  - **Merit**: Maintains communication efficiency similar to FedAvg.
  - **Demerit**: Slightly higher overhead for gradient harmonization.

- **FedSAM**:

  - **Merit**: Communication-efficient as perturbations occur locally without extra server communication.
  - **Demerit**: Increased local computation might indirectly affect overall efficiency.

5. Conclusion

This report provides a comprehensive analysis of six federated learning algorithms, highlighting their performance across critical metrics such as data heterogeneity handling, communication efficiency, and convergence rates. **Centralized Training** serves as a benchmark, while **FedSGD** and **FedAvg** offer baseline approaches with limitations under high heterogeneity. Advanced methods like **SCAFFOLD** and **FedGH** address client drift and gradient conflicts, excelling in heterogeneous settings. **FedSAM** emerges as a promising approach, leveraging sharpness-aware minima for improved generalization at the cost of increased local computation. The results underscore the importance of algorithm selection based on deployment-specific requirements, providing valuable guidelines for the practical implementation of federated learning in real-world scenarios.

6. Contributions

- **Muhammad Saad Haroon**: Task 1 & Task 5 & task 6
- **Jawad Saeed**: Task 3 & Task 4.
- **Daanish ud Din**: Task 2 & Task 6.

References

[1] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. *SCAFFOLD: Stochastic Controlled Averaging for Federated Learning*. In Proceedings of the 37th International Conference on Machine Learning (ICML), pages 5132–5143. PMLR, 2020.

[2] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. *Federated Optimization: Distributed Machine Learning for On-Device Intelligence*. arXiv preprint arXiv:1610.02527, 2016.

[3] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. *Communication-Efficient Learning of Deep Networks from Decentralized Data*. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), pages 1273–1282. PMLR, 2017.

[4] Z. Qu, X. Li, R. Duan, Y. Liu, B. Tang, and Z. Lu. *Generalized Federated Learning via Sharpness Aware Minimization*. In Proceedings of the 39th International Conference on Machine Learning (ICML). PMLR, 2022.

[5] X. Zhang, W. Sun, and Y. Chen. *Tackling the Non-IID Issue in Heterogeneous Federated Learning by Gradient Harmonization*. IEEE Signal Processing Letters, 2024.