Theoretical Convergence Guaranteed Resource-Adaptive Federated Learning with Mixed Heterogeneity

Yangyang Wang
Shandong University
Qingdao, China

Xiao Zhang∗
Shandong University
Qingdao, China

Mingyi Li
Shandong University
Qingdao, China

Tian Lan
George Washington University
Washington, United States

Huashan Chen
Chinese Academy of Sciences
Beijing, China

Hui Xiong
Hong Kong University of Science and Technology
Guangzhou, China

Xiuzhen Cheng
Shandong University
Qingdao, China

Dongxiao Yu∗
Shandong University
Qingdao, China

ABSTRACT

In this paper, we propose an adaptive learning paradigm for resource-constrained cross-device federated learning, in which heterogeneous local submodels with varying resources can be jointly trained to produce a global model. Different from existing studies, the submodel structures of different clients are formed by arbitrarily assigned neurons according to their local resources. Along this line, we first design a general resource-adaptive federated learning algorithm, namely RA-Fed, and rigorously prove its convergence with asymptotically optimal rate $O(1/\sqrt{T+Q})$ under loose assumptions. Furthermore, to address both submodels heterogeneity and data heterogeneity challenges under non-uniform training, we come up with a new server aggregation mechanism RAM-Fed with the same theoretically proved convergence rate. Moreover, we shed light on several key factors impacting convergence, such as minimum coverage rate, data heterogeneity level, submodel induced noises. Finally, we conduct extensive experiments on two types of tasks with three widely used datasets under different experimental settings. Compared with the state-of-the-arts, our methods improve the accuracy up to 10% on average. Particularly, when submodels jointly train with 50% parameters, RAM-Fed achieves comparable accuracy to FedAvg trained with the full model.

CCS CONCEPTS
• Computing methodologies → Machine learning; • Computer systems organization → Distributed architectures.

∗Corresponding authors. Email: {xiaozhang, dxyu}@sdu.edu.cn

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD ’23, August 6–10, 2023, Long Beach, CA, USA
© 2023 Copyright held by the owner/authors(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0103-0/23/08...$15.00
https://doi.org/10.1145/3580305.3599521

1 INTRODUCTION

In recent years, with the promulgation of kinds of data regulations such as GDPR and individuals’ awareness of privacy data protection, federated learning has drawn rapidly growing interest from both academia and industry. The classical federated learning is centralized with a parameter server, in which model parameters can be learned from local dispersed datasets and then sent to the server for aggregation without sharing local data. Particularly, with the rapid increase of the volume of data generated by massive mobile and IoT devices [11–13, 24], the cross-device federated learning [8, 22] has become a popular distributed computing paradigm.

In real-world cross-device federated learning scenarios, mobile devices are usually equipped with limited resources for computation and communication which seriously restrict the convergence performance of the federated learning algorithms. It would be difficult and unaffordable for the resource-constrained clients to run the full model for coordination in federated learning, especially for the arising large models like ChatGPT [6]. Therefore, kinds of technologies such as model compression [19], model pruning [5], splitting learning [20] have been introduced to reduce model size or communication cost to facilitate cross-device federated learning feasible. For instance, PruneFL [5] selects the important parameters to train with adaptive pruning. SplitFL [20] combined splitting learning with federated learning to split the full model into smaller parts and train them on a server, and distributed clients separately.

In this work, we consider a novel learning paradigm in resource-limited federated learning. Different from the traditional federated learning in which each client needs to update the full model in each


global epoch, different clients can train different submodels according to their own resource constraints. Thus the resource-adaptive learning paradigm aims to train heterogeneous local submodels with varying resources and still produce a single global inference model. Recently, independent subnet training (IST) [26] belongs to this kind of learning paradigm with a strong assumption that hidden neurons are all random uniformly assigned to disjoint computing nodes. Literature [30] achieves this goal by adaptively pruning the shared global full model and establishing sufficient conditions for the heterogeneous submodels to converge. In this work, we consider more general cases without these strong assumptions where existing works would become special cases of our proposed learning paradigm. An example is shown in Fig. 1, the submodel structures of different clients are formed by arbitrarily assigned neurons according to their local resources. As the training continues, the submodel structure within the same client could also change continuously due to the changing resources.

Thus in order to achieve this goal, several non-trivial challenges arise. (1) **Submodel heterogeneity.** The arbitrary submodels training induced uncontrollable noises compared with the full model training, which would affect the performance of federated models. (2) **Non-uniform training.** Due to the arbitrarily constructed submodels, it is obvious that not all the neurons of the whole network can be trained in each round. As shown in the depicted Fig. 1, neuron region A is never trained by any clients in this training epoch. The insufficient training would make the convergence of the federated model difficult, which has never been addressed by existing IST and literature [30]. (3) **Data heterogeneity** denotes one same neuron region might be trained in different clients whose data distributions could be not independent and identically distributed (data heterogeneity) [27, 28]. Taking neuron region B in Fig. 1 as an example, which is trained by submodel 2, 3, 4 simultaneously in a certain training round. Especially mixing with submodels heterogeneity, different neuron regions of the full models might be trained by different subsets of clients, which further exacerbates slow convergence [23] and has also been ignored by existing IST and literature [30]. (4) **Theoretical guarantee.** Under the arbitrarily assigned neurons training paradigm, arising with the submodel heterogeneity, non-uniform training, and data heterogeneity challenges, how to theoretically guarantee the convergence rate of our proposed algorithm is unprecedentedly challenging. Little is known about whether such algorithms can converge like standard federated learning methods.

Along this line, we first propose a general resource-adaptive federated learning framework, namely RA-Fed, under arbitrary neuron assignments. Within every training round, the server sends the global model to all clients, different clients leverage adaptive online masks to train heterogeneous submodels with varying neuron regions, and then the server receives and aggregates accumulated local updates for each neuron. We give detailed convergence analysis with loose assumptions (e.g., remove bounded gradient and assume the biased mask and compression), which can achieve asymptotically optimal rate $O(1/\sqrt{\Gamma^*T_Q})$, where $Q$ is the number of communication rounds, $T$ is the number of local iterations and $\Gamma^*$ is the minimum coverage rate defined in Section 3. Moreover, to mitigate the effects of both submodel heterogeneity and data heterogeneity under non-uniform training, we further proposed RAM-Fed with a new server aggregation mechanism, in which the server stores the latest updates for different regions of global model in each client, and reuses it as an approximation for current regions update. We also prove that RAM-Fed can achieve the same convergence rate with RA-Fed. Finally, extensive experiments are conducted on two widely used datasets, both RA-Fed and RAM-Fed demonstrating its superiority over other baselines. Our source code is available on github1.1 The main contributions of this paper are summarized as follows:

- To the best of our knowledge, we are the first to propose the arbitrarily assigned neurons based resource-adaptive federated learning paradigm. The heterogeneous local submodels with varying resources can be jointly trained to produce a single global model.
- We design a general resource-adaptive learning algorithm RA-Fed under arbitrary neuron assignments. We give detailed convergence analysis with loose assumptions to prove RA-Fed can achieve asymptotically optimal rate $O(1/\sqrt{\Gamma^*T_Q})$, which can achieve speedup with coverage level $\Gamma^*$.
- Furthermore, in order to mitigate the effects of both submodel heterogeneity and data heterogeneity under non-uniform training, we further propose RAM-Fed with a new server aggregation mechanism. We also theoretically prove the RAM-Fed can also converge with $O(1/\sqrt{\Gamma^*T_Q})$.
- Based on the theoretical convergence analysis, we investigate several key factors impacting convergence rate, such as the minimum coverage rate $\Gamma^*$, data heterogeneity level, submodel induced noises.
- We perform extensive experiments on two different tasks with three datasets by comparing with state-of-the-art algorithms under different experimental settings. Our algorithms improve 10% accuracy compared with the optimal results in baselines. Particularly, RAM-Fed with 50% model achieves comparable accuracy to FedAvg trained with the full model.

---

1https://github.com/wyy-123-xyy/RA-Fed
In summary, the proposed novel resource-adaptive learning paradigm provides a new insight and rigorously theoretical guarantee for the real-world deployment of arising large models on massive resource-limited devices. Moreover, existing studies would become special cases of our learning paradigm. When $\Gamma^* = N$, RAM-Fed achieves the same convergence rate $O(1/\sqrt{NTQ})$ as the vanilla FedAvg [16, 25]. When $\Gamma^* = 1$, RAM-Fed achieves the same convergence rate $O(1/\sqrt{NTQ})$ as OAP$^2$ [30].

2 RELATED WORK

In traditional federated learning [22], FedAvg [16] is the widely used aggregation algorithm, which achieves $O(1/\sqrt{NTQ})$ convergence rate with training full global model in each client. However, with the popularity of large models, it would be difficult for devices with limited resources to run the full model under classic federated learning. In recent years, kinds of approaches [14, 15, 21, 26, 30] has been proposed to address the resource-constrained problem. For example, literature [30] focuses on training heterogeneous models with online global model pruning and achieves convergence with strong assumptions (e.g. bounded gradient). IST [26] is proposed by decomposing the fully connected neural network into multiple subnetworks with the same depth. HeteroFL [1] designs a stable framework to train heterogeneous fixed sub-network without theoretical convergence analysis. In addition, to address the limited communication problem, several works [17–19] are proposed. DGC [10] combines gradient sparsity and multiple optimization technologies to greatly reduce communication costs with comparable accuracy. CHOCO-SGD [9] is proposed to realize arbitrary compression level with theoretical convergence on non-convex assumption. Different from any existing studies, in this work, we consider more general cases by proposing a resource-adaptive federated learning paradigm under arbitrarily assigned neurons. We also demonstrate theoretical convergence analysis for the proposed algorithms, existing studies would become special cases of our learning paradigm.

3 PRELIMINARIES

Given the resource-constrained cross-device federated learning paradigm, there exist $N$ clients, and all clients collaboratively learn a single global inference model with parameter $\theta$. The goal is to optimize the empirical risk minimization like traditional setting:

$$\min_{\theta \in \mathbb{R}^d} F(\theta) := \frac{1}{N} \sum_{n=1}^{N} F_n(\theta)$$

where $F_n(\theta) := \mathbb{E}_{\xi_n \sim D_n}[F_n(\theta, \xi_n)]$ is the local loss function of client $n$ on local dataset $D_n$.

**Definition 1.** Neuron regions. The global inference model contains $|\mathcal{K}|$ neuron regions with varying number of neurons. In extreme cases, each model neuron can be regarded as a separate region.

In our proposed resource-adaptive learning paradigm, due to the arbitrarily assigned neurons, each client can train a submodel with multiple varying neuron regions according to their own online heterogeneous resource constraints. Specifically, the adaptive online mask strategy is utilized to obtain the submodel for each client.

### Table 1: Frequently used notations

<table>
<thead>
<tr>
<th>Notations</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$|\cdot|$</td>
<td>the vector $l_1$ norm or the matrix spectral norm depending on the argument</td>
</tr>
<tr>
<td>$\mathcal{K}$</td>
<td>the set of all neuron regions</td>
</tr>
<tr>
<td>$S_q$</td>
<td>the trained neuron regions set in round $q$</td>
</tr>
<tr>
<td>$</td>
<td>S_q</td>
</tr>
<tr>
<td>$S^*$</td>
<td>minimum number of trained neuron regions: $S^* = \min</td>
</tr>
<tr>
<td>$N^i_q$</td>
<td>the set of clients training neuron region $i$ in round $q$</td>
</tr>
<tr>
<td>$\Gamma^i_q$</td>
<td>the number of clients in $N^i_q$</td>
</tr>
<tr>
<td>$\Gamma^*$</td>
<td>minimum coverage rate: $\Gamma^* = \min_{q, i} \Gamma^i_q, i \in S_q, \forall q$</td>
</tr>
<tr>
<td>$\Delta_{q,n}$</td>
<td>the accumulated local updates from client $n$ on itself submodel in round $q$</td>
</tr>
<tr>
<td>$\Delta_{i,n}$</td>
<td>the accumulated local updates from client $n$ on neuron region $i$ in round $q$</td>
</tr>
<tr>
<td>$m_{q,n}$</td>
<td>the mask of client $n$ in round $q$</td>
</tr>
<tr>
<td>$u_{q+1,n}$</td>
<td>the latest update from client $n$ on neuron region $i$ in round $q+1$</td>
</tr>
<tr>
<td>$\theta_q^i$</td>
<td>the neuron region $i$ of global model in round $q$</td>
</tr>
<tr>
<td>$\mathcal{C}(\cdot)$</td>
<td>the arbitrary compressor</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>the step size (learning rate)</td>
</tr>
</tbody>
</table>

## 4 ALGORITHM DESIGN

In this section, we design a novel resource-adaptive learning paradigm in cross-device federated learning scenarios. Due to the limited and continuously changing resources in device clients, different submodels can be trained with arbitrarily varying neuron regions
**Algorithm 1: RA-Fed**

1. **Initialize:** subdataset $D_i$ on $N$ clients, mask policy $P$, $\theta_1$ for $q = 1$ to $Q$

2. for $n = 1$ to $N$ (all workers in parallel)

3. Generate mask $m_{q,n} = P(\bar{C}(\theta_q))$

4. Generate submodel $\theta_{q,n,0} = C(\theta_q) \odot m_{q,n}$

5. # Update local submodel with multiple neuron regions:

   for epoch $t = 1$ to $T$ do

   1. $\theta_{q,n,t} = \theta_{q,n,t-1} - \gamma \nabla F_n(\theta_{q,n,t-1}, \bar{S}_{n,t-1}) \odot m_{q,n}$

   endfor

6. $\Delta_q = \frac{\theta_{q,n,T} - \theta_{q,n,T-1}}{\gamma}$

7. endfor

8. # Update all neuron regions of global model:

9. for region $i = 1$ to $K$

10. Find $N_q^i = \{ n : m_{q,n}^i = 1 \}$

11. if $\Gamma_q^i = 0$ then

12. Update $\theta_{q,n}^i = \theta_q^i$

13. else

14. Update $\theta_{q,n}^i = \theta_{q,n}^i - \gamma \frac{1}{\gamma} \sum_{n \in N_q^i} \Delta_q^i$

15. end

16. endfor

17. $\theta_{q+1} = \sum_{i=1}^{Q} \theta_{q+1}^i$

according to their own resource constraints in our work. The non-uniform training leads to that not all the neuron regions of the full model can be trained in each round. In addition, one same neuron region might be trained by different clients in each round, which might face high data heterogeneity. Therefore, we first propose a general resource-adaptive learning algorithm, namely RA-Fed to address the arising challenges. Moreover, to further mitigate the effects of both submodel heterogeneity and data heterogeneity, we propose RAM-Fed with a new server aggregation mechanism. The details are shown as follows.

### 4.1 RA-Fed Algorithm

In order to achieve resource-adaptive learning, we propose the RA-Fed algorithm, whose training process is shown in Algorithm 1. First, the server sends the globally full model to all clients, different clients leverage adaptive online masks to train heterogeneous submodels with varying neuron regions, and then the server receives and aggregates accumulated local updates for each neuron. The details of RA-Fed in the $q$-th round are described as follows:

- **Mask generation:** Online mask $m_{q,n}$ would be generated according to its resource constraints within each client.

- **Submodel construction:** Each client $n$ leverage adaptive online mask $m_{q,n}$ to generate heterogeneous local submodel $\theta_{q,n,0}$ with multiple neuron regions.

- **Local submodel update:** Each client $n$ calculates local gradients and update local submodel with $T$ iterations: $\theta_{q,n,t} = \theta_{q,n,t-1} - \gamma \nabla F_n(\theta_{q,n,t-1}, \bar{S}_{n,t-1}) \odot m_{q,n}$.

- **Uploading local updates:** Each client $n$ calculates accumulated local updates on local submodel: $\Delta_q = \frac{\theta_{q,n,T} - \theta_{q,n,T-1}}{\gamma}$.

- **Neuron regions aggregation:** For each neuron region $i$, the server calculates the number of clients which local submodels contains neuron region $i$: $\Gamma_q^i$. If $\Gamma_q^i = 0$, neuron region $i$ in round $q$ is not trained: Update $\theta_{q,n}^i = \theta_q^i$. Otherwise, the neuron region $i$ is trained by at least one client: Update $\theta_{q,n}^i = \theta_{q,n}^i - \gamma \frac{1}{\gamma} \sum_{n \in N_q^i} \Delta_q^i$.

- **Full global model generation:** The full model is constructed based on all neuron regions: $\theta_{q+1} = \sum_{i=1}^{Q} \theta_{q+1}^i$.

The convergence of the RA-Fed algorithm is theoretically proved in Sec. 5.1.

### 4.2 RAM-Fed Algorithm

Except for the submodel heterogeneity and the non-uniform training, data heterogeneity also seriously restricts the convergence and performance of the proposed resource-adaptive learning algorithms. The core challenge of the mixed heterogeneity is that one neuron region might be trained by partial clients simultaneously, or even not be trained in one round due to arbitrariness. Obviously, the key is to ensure each neuron region can be updated by all clients in each round. Inspired by the idea of memorized latest updates $[2, 4]$, we further propose the RAM-Fed algorithm with a new server aggregation mechanism to further mitigate the effects of the mixed heterogeneity.

In RAM-Fed, the server stores the latest updates from all clients on each neuron region. Specifically, in Algorithm 2, $\Lambda_q^i$, represents the current updates from client $n$ ($n \in N_q^i$) on neuron region $i$ in round $q$. To maintain the latest updates from client $n$ ($n \in N$), after each round, we perform the following step for all clients:

$$u_{q,n}^i = \begin{cases} \Lambda_q^i & \text{if } n \in N_q^i \\ u_{q,n}^i & \text{if } n \notin N_q^i \end{cases}$$

By this way, $u_{q+1,n}^i$ maintains the latest updates from all clients on neuron region $i$ in round $q$.

Then, when updating the neuron region $i$, we can use the latest updates $u_{q,n}^i (n \in N)$ in round $q - 1$ and current updates $\Lambda_q^i$ ($n \in N_q^i$) in round $q$ to compute an approximation aggregated update $v_{q,n}^i$ from all clients. Specifically, in round $q$, if the neuron region $i$ is not trained, then the neuron region $i$ will be updated by the average latest update from all clients: $v_{q,n}^i = \frac{1}{N} \sum_{n=1}^{N} u_{q,n}^i$. Otherwise, if the neuron region $i$ is trained by clients $N_q^i$, then the neuron region $i$ will be updated by $\Delta_q^i$ ($n \in N_q^i$) and $u_{q,n}^i$ ($n \notin N_q^i$): $v_{q,n}^i = \frac{1}{N} \sum_{n=1}^{N} u_{q,n}^i + \frac{1}{\gamma} \sum_{n \in N_q^i} \Delta_q^i - u_{q,n}^i$. Noted that we give higher weight to current client updates $\Lambda_q^i$ ($n \in N_q^i$) as compared to previous client updates $u_{q,n}^i$ ($n \notin N_q^i$) following FedVARP $[4]$. Thus, this can correct the update bias (only partial clients update error compared with all clients update) in neuron regions using the latest updates from all clients in each round. The convergence of the RAM-Fed algorithm is theoretically proved in Sec. 5.2.
Algorithm 2: RAM-Fed

1 Initialize: subdataset \( D_n \) on \( N \) clients, mask policy \( p, \theta_1 \),
\( u_{i,n}, (n = 1, \ldots, N) \)

2 for \( q = 1 \) to \( Q \) do

3 for \( n = 1 \) to \( N \) (all workers in parallel) do

4 Generate mask \( m_{q,n} = P(\mathcal{C}(\theta_q), n) \)

5 Generate submodel \( \theta_{q,n,0} = \mathcal{C}(\theta_q) \otimes m_{q,n} \)

6 # Update local submodel with multiple neuron regions:

7 for epoch \( t = 1 \) to \( T \) do

8 \( \theta_{q,n,t+1} = \theta_{q,n,t} - \gamma \nabla F_n(\theta_{q,n,t}; \xi_{n,t-1}) \otimes m_{q,n} \)

9 endfor

10 \( \Delta_{q,n} = \frac{\theta_{q,n,T} - \theta_{q,n,0}}{T} \)

11 endfor

12 # Store the latest update for each client:

13 for \( n = 1 \) to \( N \) do

14 \( u^{i}_{q,n} = \Delta_{q,n} \) if \( n \in N^i_q \)

15 \( \theta_{q+1} = \sum_{i=1}^{K} \theta^i_{q+1} \)

16 endfor

5 CONVERGENCE ANALYSIS

In the section, we show the convergence rate of our proposed RA-Fed and RAM-Fed algorithms. Firstly, we give some commonly used assumptions in federated learning:

**Assumption 2.** Lipschitzian Condition: Every function \( F_n(\cdot) \) is with L-Lipschitzian gradient: \( \forall n \in [N], \theta, \varphi \in \mathbb{R}^d \)

\[
\| \nabla F_n(\theta) - \nabla F_n(\varphi) \| \leq L \| \theta - \varphi \| \tag{5}
\]

**Assumption 3.** Bounded compression: An operator \( \mathcal{C} : \mathbb{R}^d \rightarrow \mathbb{R}^d \) is a w-approximate compressor over \( w_2 \) for \( w_2 \in (0, 1) \) if

\[
\mathbb{E}\| \mathcal{C}(\theta) - \theta \| ^2 \leq w_2^2 \mathbb{E}\| \theta \| ^2. \quad \forall \theta \in \Omega \tag{6}
\]

**Assumption 4.** Bounded variance: There exists \( \sigma > 0 \):

\[
\mathbb{E}_{\xi_{n,t}-D_n}\| \nabla F_n(\theta_{q,n,t}; \xi_{n,t}) - \nabla F_n(\theta_{q,n,t}) \|^2 \leq \sigma^2, \quad \forall q,n,t \tag{7}
\]

\( \sigma > 0 \) bounds the variance of stochastic gradient.

**Assumption 5.** Bounded data heterogeneity level: There exists \( \delta > 0 \):

\[
\| \nabla F_n(\theta_q) - \nabla F(\theta_q) \|^2 \leq \delta^2 \tag{8}
\]

\( \delta > 0 \) bounds the effect of heterogeneous data.

5.1 Convergence analysis of RA-Fed

**Lemma 1.** Deviation of local submodel and global model:

Let all assumptions hold.

\[
1 \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\| \theta_{q,n,t-1} - \theta_q \|^2 \leq 4y^2T^2\sigma^2 + 32y^2T^2\delta^2 + 32y^2T^2\sum_{i \in S_q} \mathbb{E}\| \nabla^2 F_i(\theta_i) \|^2 + 4w_2^2 \mathbb{E}\| \theta_q \|^2 \tag{9}
\]

Lemma 1 bounds the difference between local submodel and global model. It indicates that the effects of local submodel training: \( \theta_{q,n,t-1} - \theta_{q,n,0} \) and mask and compression error: \( \theta_{q,n,0} - \theta_q \). Note that \( \theta_{q,n,0} - \theta_q \) can be split into mask error \( \mathcal{C}(\theta_q) \otimes m_{n,q} \) and compression error \( \mathcal{C}(\theta_q) - \theta_q \).

**Theorem 1.** Let all assumptions hold. Suppose that the step size \( \gamma \) satisfies the following relationships:

\[
\begin{align*}
8y^2T^2 \leq \frac{1}{\gamma} & \Rightarrow \gamma \leq \frac{1}{8y^2T^2} \\
32y^2T^2 \frac{\Gamma L}{\sqrt{N}} \leq \frac{1}{\gamma} & \Rightarrow \gamma \leq \frac{1}{32y^2T^2 \frac{\Gamma L}{\sqrt{N}}} \\
96L^3y^3T^3 \frac{1}{2} \gamma^2 & \leq \frac{1}{\gamma} \Rightarrow \gamma \leq \frac{1}{96L^3y^3T^3} \\
\frac{3}{2} \gamma^2 T^2 \leq \frac{1}{8} & \Rightarrow \gamma \leq \frac{1}{12T^2} 
\end{align*}
\]

Therefore, the step size \( \gamma \) is defined as:

\[
0 \leq \gamma \leq \min \left\{ \frac{1}{12T^2}, \sqrt{\frac{\Gamma}{16L^2Y}} \frac{(\gamma^*)^{-1}}{16L^2 N}, \frac{\Gamma^*}{768LTN^{\frac{1}{2}}} \right\}
\]

Then, for all \( Q \geq 1 \), we have:

\[
\frac{1}{Q} \sum_{q=1}^{Q} \mathbb{E}\| \nabla F_i(\theta_i) \|^2 \leq \frac{8\mathbb{E}[F(\theta_1)]}{TYQ} + \left( 64w_2^2 \frac{N}{\Gamma^2} L^2 + 96L^3yT \frac{N}{\Gamma^2} w_2^2 \right) \frac{1}{Q} \sum_{q=1}^{Q} \mathbb{E}\| \theta_q \|^2 \\
+ \frac{8N}{\Gamma T} \left( 32y^2T^2L^2 + 1 + 96L^3y^3T^3 + 3LyT \right) \delta^2 \\
+ \gamma L \frac{8N}{\Gamma T} (4LyT + \frac{3}{2} + 12L^2y^2T^2) \sigma^2
\]

where \( 2w_2^2 + 2w_2^2 + w_2^2 = w^2 \)

Theorem 1 shows the convergence rate of algorithm RA-Fed by giving the upper bound on the average gradient of all clients for all trained neuron regions.

**Remark 1** Impact of the number of trained neuron regions \( |S_q| \).

Our algorithm is novel with stronger generalization, in which not all neuron regions can be trained in each round. Specifically, regions \( (K - S_q) \) can not be trained in round \( q \). It is obvious that in identical settings, the larger \( |S_q| \), the more neuron regions trained, the more gradients on neuron regions can be bounded, the better the convergence rate. Furthermore, to reduce the impact of partial neuron regions not being updated in some rounds on the convergence rate, we design a new server aggregation mechanism in Algorithm 2, which achieves the same convergence rate and ensures that all neuron regions can be updated in each round.

**Remark 2:** Impact of the mask-induced noise \( w_1 \) and compression noise \( w_2 \).
Our convergence result shows that the smaller noises \(w^2 = 2w_1^2w_2^2 + 2w_1^2 + w_2^2\) would lead to a faster convergence rate and better performance in federated learning. Besides, it is worth noting that the mask-induced noise is also highly related to \(S_q\) and \(\Gamma^*\).

**Remark 3:** Impact of the data heterogeneity level \(\delta\).

In our learning paradigm, we consider heterogeneous data distributions in real-world scenarios. The larger \(\delta\) denotes the higher the data heterogeneity level and the slower convergence rate. Therefore, to reduce the impact of data heterogeneity on learning performance, we propose the RAM-Fed shown in Algorithm 2.

Next, by choosing the appropriate convergence rate \(\gamma\) and the parameters representing data heterogeneity levels \(\delta\), we can obtain the following corollary.

**Corollary 1.** Let all assumptions hold. Supposing that the step size \(\gamma = O(\sqrt{\Gamma^*})\) and that \(\delta = O(\frac{1}{\sqrt{\Gamma^*}})\), when the constant \(C > 0\) exists, the convergence rate can be expressed as follows:

\[
\frac{1}{Q} \sum_{q=1}^{Q} \sum_{i \in S_q} E[\|\nabla F_i(\theta_q)\|^2] \leq C(\frac{1}{\sqrt{\Gamma^*}} + \frac{1}{\Gamma^*} + \frac{1}{\gamma^2} + \frac{1}{\Gamma^*} + \frac{1}{Q^2} + \frac{1}{Q^2} + \frac{1}{Q^2})
\]

(Corollary 1 indicates that when \(Q\) is sufficiently large, the term \(O(1/\sqrt{\Gamma^*TQ})\) will dominate the convergence rate and the convergence increases with the number of \(\Gamma^*\).

The detailed theoretical proof of Theorem 1 and Corollary 1 are provided in Supplement.

**Remark 4:** Impact of the minimum coverage rate \(\Gamma^*\).

The Corollary 1 demonstrates that our proposed RA-Fed algorithm can converge to \(O(1/\sqrt{\Gamma^*TQ})\) under arbitrary adaptive online mask. Except for the non-trained neuron regions from the global model, others can be trained by at least \(\Gamma^*\) submodels in each round. Intuitively when fixing other impacting factors, as \(\Gamma^* \) increases, the more frequently the neuron region can be trained, so the faster RA-Fed can converge to a stationary point.

### 5.2 Convergence analysis of RAM-Fed

**Assumption 6.** Number of continuously non-trained rounds: We define the total number of rounds that client \(n\) has not trained neuron region \(i\) continuously as \(r_{i,n}\):

\[
\tau_q = \max_{n,i} r_{i,n}, n \in N, i \in K
\]

**Assumption 7.** Bounded gradient: In algorithm 2, the expected squared norm of stochastic gradients is bounded uniformly, for constant \(G > 0\) and \(\forall n, q, i:

\[
E[\|\nabla F_n(\theta_{q,n,t}, \xi_{n,t-1})\|^2] \leq G.
\]

**Lemma 2.** Deviation of average submodel stochastic gradient between round \(q\) and round \(q - \tau_q\): Let all assumptions hold.

\[
\sum_{i \in S_q} \sum_{n \in N_q} \sum_{T=1}^{T} \frac{1}{T} \nabla F_i(\theta_{q,n,T-1}, \xi_{n,T-1}) - \frac{1}{\Gamma^*} \sum_{n \in N_q} \sum_{T=1}^{T} \nabla F_i(\theta_{q-n,T-1}, \xi_{n,t-1})^2
\leq 6 \frac{N}{\Gamma^*} \sigma^2 + 18 \frac{N}{\Gamma^*} L^2(4\gamma^2 T \sigma^2 + 32\gamma^2 T^2 \delta^2 + 32\gamma^2 T^2 G)
\]

**Remark 5** Impact of the maximum number of continuously non-trained rounds \(\tau_q\).

In our convergence analysis, we need to satisfy: \(q - \tau_q > 0 \Rightarrow \tau_q < q\). Recall that \(\tau_q = \max_{n,i} r_{i,n}\) means until round \(q\), the maximum number of non-trained for all neuron regions on all clients. Therefore, in our algorithm, we can only ensure that all neuron regions are trained on all clients in the first round. In this case, inequality \(\tau_q < q\) always holds. Moreover, the result indicates that the larger \(\tau_q\), the worse the convergence rate.

Next, by choosing the appropriate convergence rate \(\gamma\), we can obtain the following corollary.

**Corollary 2.** Let all assumptions hold. Supposing that the step size \(\gamma = O(\sqrt{\Gamma^*})\) and \(\sigma\) is sufficiently small, when the constant \(C > 0\) exists, the convergence rate can be expressed as follows:

\[
\frac{1}{Q} \sum_{q=1}^{Q} E[\|\nabla F(\theta_q)\|^2] \leq C(\frac{1}{\sqrt{\Gamma^*TQ}} + \frac{1}{\Gamma^*Q} + \frac{1}{\gamma^2} + \frac{1}{\Gamma^*} + \frac{1}{Q^2})
\]
Table 2: Performance comparison on MLP-MNIST and CNN-CIFAR10. '-' means this method doesn’t work under corresponding mask level setting. Bold is the optimal result except for FedAvg with full model training, underlined is the suboptimal result.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Mask level</th>
<th>MLP-MNIST (Accuracy %)</th>
<th>CNN-CIFAR10 (Accuracy %)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$a = 0.01$</td>
<td>$a = 0.05$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a = 0.1$</td>
<td>$a = 0.15$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a = 0.2$</td>
<td>$a = 0.3$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a = 0.4$</td>
<td></td>
</tr>
<tr>
<td>FedAvg</td>
<td>Full</td>
<td>87.9</td>
<td>91.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>94.2</td>
<td>96.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>61.9</td>
<td>64.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70.6</td>
<td>70.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75.8</td>
<td></td>
</tr>
<tr>
<td>SplitFL</td>
<td>SameStr.</td>
<td>52.2</td>
<td>69.2</td>
</tr>
<tr>
<td></td>
<td>Arb.</td>
<td>72.4</td>
<td>67.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>37.6</td>
<td>45.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50.1</td>
<td>63.8</td>
</tr>
<tr>
<td>IST</td>
<td>U.A.</td>
<td>66.9</td>
<td>80.5</td>
</tr>
<tr>
<td></td>
<td>Arb.</td>
<td>87.0</td>
<td>91.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>24.5</td>
<td>25.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.1</td>
<td>27.9</td>
</tr>
<tr>
<td>PruneFL</td>
<td>L-Arb.</td>
<td>80.8</td>
<td>87.5</td>
</tr>
<tr>
<td></td>
<td>S-Arb.</td>
<td>91.5</td>
<td>94.8</td>
</tr>
<tr>
<td></td>
<td>MIX-Arb.</td>
<td>48.7</td>
<td>51.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>53.6</td>
<td>53.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60.0</td>
<td></td>
</tr>
<tr>
<td>OAP</td>
<td>L-Arb.</td>
<td>65.8</td>
<td>77.7</td>
</tr>
<tr>
<td></td>
<td>S-Arb.</td>
<td>84.9</td>
<td>92.2</td>
</tr>
<tr>
<td></td>
<td>MIX-Arb.</td>
<td>45.9</td>
<td>49.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50.5</td>
<td>50.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>56.0</td>
<td></td>
</tr>
<tr>
<td>RA-Fed(ours)</td>
<td>L-Arb.</td>
<td>85.2</td>
<td>89.0</td>
</tr>
<tr>
<td></td>
<td>S-Arb.</td>
<td>92.8</td>
<td>95.6</td>
</tr>
<tr>
<td></td>
<td>MIX-Arb.</td>
<td>49.2</td>
<td>54.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>57.8</td>
<td>57.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>65.1</td>
<td></td>
</tr>
<tr>
<td>RAM-Fed(ours)</td>
<td>L-Arb.</td>
<td>80.0</td>
<td>90.1</td>
</tr>
<tr>
<td></td>
<td>S-Arb.</td>
<td>95.4</td>
<td>96.6</td>
</tr>
<tr>
<td></td>
<td>MIX-Arb.</td>
<td>52.7</td>
<td>55.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>61.8</td>
<td>61.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>65.3</td>
<td></td>
</tr>
</tbody>
</table>

Corollary 2 indicates that when $Q$ is sufficiently large, the term $O(1/\sqrt{NTQ})$ will dominate the convergence rate and the convergence increases with $\Gamma^*$. 

Remark 6 Recall that $\Gamma^* = \min_{\theta, q} \Gamma^*_q \forall q$ measures the minimum number of submodels training the corresponding neuron region $i \in S_q$ in all rounds. Thus, it is obvious that $1 \leq \Gamma^* \leq N$. When $\Gamma^* = N$, all submodels can train neuron region $i \in S_q$, which achieves the same convergence rate $O(1/\sqrt{NTQ})$ as the vanilla FedAvg [16, 25]. When $\Gamma^* = 1$, we achieve the same convergence rate $O(1/\sqrt{TQ})$ as OAP [30].

The detailed theoretical proof of Theorem 2 and Corollary 2 are provided in Supplement.

6 EXPERIMENTS

In this section, we conduct extensive experiments to evaluate kinds of federated learning paradigms. The details are shown as follows.

6.1 Datasets and baselines

6.1.1 Task and dataset description. We perform all approaches on two different types of machine learning tasks: image classification on MNIST and CIFAR10, text classification on AGNews [29].

Image classification: We train CNN with 2 convolution layers and 3 hidden layers on CIFAR10. MLP with 2 hidden layers on MNIST. CIFAR10 contains 10 categories with 50k training images and 10k testing images. MNIST contains 10 categories with 60k training images and 10k testing images. Text classification: We train FastText [7] with 1 embedding layer and 2 hidden layers on AGNews. AGNews contains 4 categories with 120k training news articles and 7600 testing news articles.

6.1.2 Dataset partition. We use Dirichlet distribution Dir($\alpha$) [3] to set up different data heterogeneity levels. The smaller $\alpha$ represents the stronger heterogeneity levels.

6.1.3 Baselines and Metrics. We compare our learning paradigm with related state-of-the-art methods in resource-limited federated learning: Fedavg [16], SplitFL [20], IST [26], PruneFL [5] and OAP [30]. We evaluate all approaches on two important evaluation Metrics in resource-limited federated learning. The Mask level measures the average local submodel size and the arbitrariness level of submodels. The Accuracy measures the performance of different learning paradigms on various tasks.

6.2 Experimental setup

Submodel setup: The submodels are designed based on arbitrarily assigned neurons, which denotes that not all the neurons of the full network can be trained in each round. To achieve this goal, we randomly select partial neuron regions not to be trained periodically. We design three different mask levels to generate different numbers of submodel parameters. L or S denotes that submodels train 50% or 25% parameters of the full model respectively. Differently, MIX denotes that 50% submodels are trained with 50% parameters, while others are with 25% parameters. Specifically, in each round, the global model $\theta$ can be split into 4 neuron regions, $\theta = \{\theta^1, \theta^2, \theta^3, \theta^4\}$. Considering arbitrarily assigned neuron regions, when the mask level is set to L-Arb, each client can adaptively select 2 neuron regions (e.g. $\{\theta^1, \theta^2\}$) to train. Thus, at most 6 types of heterogeneous submodels can be generated. Under MIX-Arb setting, we randomly select 50% submodels with 2 adaptively chosen neuron regions, while others are with 1 adaptively chosen neuron region for training. In different rounds, the submodel structure within the same client could change due to the varying resources. It is worth noting that for FedAvg, full models need to be trained. The
Numerical results

About the results on image classification tasks in Table 2, we can observe that:

- On the whole, our algorithms outperform all baselines under different mask levels and data heterogeneity level settings. Except FedAvg with full model, in testing accuracy, RA-Fed nearly improves 2%-20% and RAM-Fed improves 4%-22%. Comparing with the baselines excepting FedAvg, RA-Fed and RAM-Fed improve accuracy by 8.5% and 10% on average respectively. Particularly, RAM-Fed with 25% submodel achieves comparable accuracy to PruneFL with 50% model.

- Compared with FedAvg, we observe that FedAvg is slightly higher than our algorithms in testing accuracy but RAM-Fed with L-Arb. mask level slightly outperforms FedAvg on MLP-MNIST, which demonstrates that our algorithm is robust in the resource-limited learning environment.

- Especially in our proposed RAM-Fed algorithm, it greatly improves the performance in non-uniform training and achieves the highest accuracy under high data heterogeneity levels, which further indicates that the new aggregation mechanism in RAM-Fed effectively mitigates the impact of non-uniform and data heterogeneity.

  - For the impact of mask level, all algorithms nearly perform worse due to the larger induced noises when the mask level varies from L-Arb. to S-Arb.. It indicates that the mask-induced error is a key factor impacting performance which is consistent with our theoretical analysis.

  - With the increment of data heterogeneity level $\alpha$, all methods’ accuracy generally becomes worse. But RAM-Fed decreases slightly which demonstrates that RAM-Fed is more robust in data heterogeneous scenarios.

  - In general, comparing with other baselines, PruneFL can achieve higher accuracy, which is due to the fact that important parameters are selected for training in every round.

About results on text classification task in Table 3, we conclude:

- Noticeably, in any mask level, RAM-Fed performs better than other algorithms, nearly achieving 7%-30% improvements. Comparing with the baselines, RA-Fed and RAM-Fed improve accuracy by 9% and 13% on average respectively.

- Surprisingly, RAM-Fed algorithm nearly performs better than FedAvg, which might be because partial stale gradients could be larger than the current gradients with the right direction. This further demonstrates the effectiveness of our proposed submodels joint novel training mechanism.

- RA-Fed and RAM-Fed algorithms all achieve higher accuracy than PruneFL, which indicates that adaptive strategy could perform better than high-wight parameter selection method.

- Even comparing with SplitFL and IST with uniform training, RA-Fed and RAM-Fed all achieve better performance with arbitrarily assigned neurons.

The convergence process of different learning paradigms on image classification ($L$-Arb. mask level, $\alpha = 0.15$ on MLP-MNIST) and text classification task ($L$-Arb. mask level, $\alpha = 0.2$ on FastText-AGNews) are depicted in Fig. 2.

- As shown in Fig. 2(a) and Fig. 2(b), RA-Fed and RAM-Fed have similar convergence trends with FedAvg on MLP-MNIST, this is because neuron regions can be trained sufficiently with training continues. Surprisingly, RAM-Fed achieves better performance significantly compared with other algorithms on FastText-AGNews.

- OAP diverges with obvious fluctuations during training, while IST converges slowly. SplitFL has the worst performance on MLP-MNIST, which is might be due to the over-fitting of the same submodel structures across all clients.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Mask Level</th>
<th>Accuracy (%)</th>
<th>$\alpha = 0.05$</th>
<th>$\alpha = 0.1$</th>
<th>$\alpha = 0.15$</th>
<th>$\alpha = 0.2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FedAvg</td>
<td>Full</td>
<td>73.7</td>
<td>71.8</td>
<td>82.0</td>
<td>82.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SameStr.</td>
<td>50.4</td>
<td>73.4</td>
<td>82.0</td>
<td>83.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arb.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>IST</td>
<td>U.A.</td>
<td>51.6</td>
<td>52.3</td>
<td>57.7</td>
<td>62.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arb.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>SplitFL</td>
<td>L-Arb.</td>
<td>67.3</td>
<td>72.0</td>
<td>80.3</td>
<td>80.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S-Arb.</td>
<td>57.8</td>
<td>59.6</td>
<td>71.2</td>
<td>72.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MIX-Arb.</td>
<td>66.2</td>
<td>70.3</td>
<td>79.5</td>
<td>80.3</td>
<td></td>
</tr>
<tr>
<td>PruneFL</td>
<td>L-Arb.</td>
<td>45.8</td>
<td>44.1</td>
<td>51.7</td>
<td>53.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S-Arb.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MIX-Arb.</td>
<td>37.3</td>
<td>39.7</td>
<td>46.2</td>
<td>49.4</td>
<td></td>
</tr>
<tr>
<td>OAP</td>
<td>L-Arb.</td>
<td>73.3</td>
<td>78.1</td>
<td>84.6</td>
<td>84.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S-Arb.</td>
<td>66.7</td>
<td>72.9</td>
<td>83.0</td>
<td>84.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MIX-Arb.</td>
<td>70.8</td>
<td>77.3</td>
<td>83.9</td>
<td>84.7</td>
<td></td>
</tr>
<tr>
<td>RA-Fed</td>
<td>L-Arb.</td>
<td>77.6</td>
<td>86.6</td>
<td>89.3</td>
<td>89.4</td>
<td></td>
</tr>
<tr>
<td>(ours)</td>
<td>S-Arb.</td>
<td>73.1</td>
<td>85.3</td>
<td>87.1</td>
<td>88.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MIX-Arb.</td>
<td>76.8</td>
<td>86.0</td>
<td>88.4</td>
<td>89.2</td>
<td></td>
</tr>
</tbody>
</table>

| RAM-Fed   | L-Arb.     | 73.3         | 78.1           | 84.6           | 84.9           |                |
| (ours)    | S-Arb.     | 66.7         | 72.9           | 83.0           | 84.1           |                |
|          | MIX-Arb.   | 70.8         | 77.3           | 83.9           | 84.7           |                |
6.4 Impact of key factors

6.4.1 Impact of minimum coverage rate \( \Gamma^* \) and minimum number of trained neuron regions \( S^* \). Based on the above analysis, the proposed adaptive learning paradigm is essential and effective. Combined with our theoretical analysis, we study two key factors impacting convergence and accuracy: \( \Gamma^* \) and \( S^* \). Fixing other impacting factors in RA-Fed, we set S-Arb. mask level, \( \alpha = 0.15 \) on MLP-MNIST task, and \( \alpha = 0.5 \) on FastText-AGNews task. Through varying \( \Gamma^* \) and \( S^* \), we set three combinations: (\( \Gamma^* = 1, S^* = 4 \)), (\( \Gamma^* = 2, S^* = 4 \)), (\( \Gamma^* = 10, S^* = 1 \)). As shown in Fig. 3, we have some observations:

- When fixing \( S^* = 4 \), we find that \( \Gamma^* = 2 \) performs slightly better than \( \Gamma^* = 1 \) in testing accuracy. As shown in testing loss, the trends clearly show that \( \Gamma^* = 2 \) can converge faster. It is worth noting that the testing loss increases rapidly at the initial rounds, and then decreases slowly. This is mainly because only partial neurons can be trained at first, but as the training continues, all neuron regions can be trained sufficiently. In addition, we observe that in testing loss, the \( \Gamma^* = 2 \) curve decreases earlier than \( \Gamma^* = 1 \), which is consistent with our theoretical analysis.

- The larger \( \Gamma^* \) does not mean the higher accuracy and faster convergence. Considering an extreme example with the largest \( \Gamma^* \) (e.g. \( \Gamma^* = 10, S^* = 1 \)), in this case, only one neuron region is trained by ten submodels in each round. However, the testing accuracy decreases significantly and testing loss converges very slowly which indicates that when a large number of neuron regions are not trained, the model performance becomes poor. Therefore, we can conclude that the performance and convergence rate are impacted by multiple factors comprehensively.

6.4.2 Impact of the maximum number of continuously non-trained rounds \( \tau_q \). We further consider two key factors impacting convergence and accuracy in RAM-Fed: \( \Gamma^* \) and \( \tau_q \). Fixing other impacting factors in RAM-Fed, we set S-Arb. as mask level, \( \alpha = 0.15 \) on MLP-MNIST task. Considering \( \Gamma^* \) and \( \tau_q \), we set four combinations: (\( \Gamma^* = 1, \tau_q = 200 \)), (\( \Gamma^* = 2, \tau_q = 200 \)), (\( \Gamma^* = 1, \tau_q = 4 \)), (\( \Gamma^* = 2, \tau_q = 4 \)). As shown in Fig. 4, we have some observations:

- When fixing \( \Gamma^* \), \( \tau_q = 4 \) performs better than \( \tau_q = 200 \) in testing accuracy and loss. Thus, when \( \tau_q \) is very large (e.g. \( \tau_q = 200 \)), some neurons can only be updated by stale gradients continuously, which causes a bias compared with the right descent direction.

- When fixing \( \tau_q \), it is obvious that as \( \Gamma^* \) increases, the performance becomes better, which further indicates that the larger \( \Gamma^* \), the more fully training the neuron regions.

- On the whole, \( \Gamma^* \) and \( \tau_q \) play important roles in convergence. For RAM-Fed, the optimal \( \Gamma^* \) and \( \tau_q \) can significantly improve performance.

7 CONCLUSION

In traditional cross-device federated learning, massive devices are usually equipped with limited resources for computation and communication which would be unaffordable to run the full model for coordination. To this end, we designed an adaptive learning paradigm, in which heterogeneous local submodels with arbitrarily assigned neurons can be jointly trained to produce a single global model. In order to address the arising submodels heterogeneity, non-uniform training and data heterogeneity challenges, we proposed general RA-Fed algorithm and RAM-Fed with a new server aggregation mechanism. We theoretically proved the proposed RA-Fed and RAM-Fed can both converge with asymptotically optimal rate \( O(1/\sqrt{\tau q T}) \) under given assumptions. We investigated several key factors impacting convergence, such as minimum coverage rate, data heterogeneity level, submodel induced noises. Extensive experiments were conducted on two types of tasks with three widely used datasets. Compared with the state-of-the-art baselines, our algorithms improved the accuracy up to 10% on average. Particularly, RAM-Fed with 50% model achieved comparable accuracy compared with FedAvg with full model, even outperforming FedAvg.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China under Grant No. 2022YFF0712100, in part by the National Natural Science Foundation of China under Grant 62202273, in part by National Science Fund for Excellent Young Scholars of China under Grant 62122042, in part by Major Basic Research Program of Shandong Provincial Natural Science Foundation under Grant ZR2022ZD02, in part by Shandong Provincial Natural Science Foundation of China under Grant ZR2021QE044, in part by the Fundamental Research Funds for the Central Universities.
A SUPPLEMENT

A.1 Part One

Let us start the proof of RA-Fed from L-Lipschitzian Condition:

\[
E[F(\theta_{q+1})] - E[F(\theta_1)] \leq \sum_{q=1}^{Q} E[\nabla F(\theta_q), \theta_{q+1} - \theta_q >]
\]

Therefore, we have:

\[
\frac{Ty}{8} \sum_{q=1}^{Q} \sum_{i \in S_q} E[\nabla^2 F^i(\theta_q)] \leq E[F(\theta_1)] - E[F(\theta_{Q+1})]
\]

\[
+ (8w^3yT^2N^2L^2 + 12L^3y^2T^2N^3w^2) \sum_{q=1}^{Q} E[\|\theta_q\|]^2
\]

\[
+ TyQ \sum_{N} \left(32y^2T^2L^2 + 1 + 96y^3T^3 + 3LyT\right)\delta^2
\]

\[
y^2TQ \sum_{N} \left(4yTL + \frac{3}{2} + 12L^2y^2T^2\right)\sigma^2
\]

dividing both sides by \( Q \) and \( \frac{Ty}{8} \):

\[
\frac{1}{Q} \sum_{q=1}^{Q} \sum_{i \in S_q} E[\nabla F^i(\theta_q)] \leq \frac{E[F(\theta_1)]}{TyQ}
\]

\[
+ (64w^3yT^2N^2L^2 + 96y^3T^3 + 3LyT)\delta^2
\]

\[
y^2TQ \sum_{N} \left(4yTL + \frac{3}{2} + 12L^2y^2T^2\right)\sigma^2
\]

Supposing that the step size \( \gamma = O(\frac{1}{\sqrt{TQ}}) \) and that \( \delta = O(\frac{1}{\sqrt{TQ}}) \), when the constant \( C > 0 \) exists, the convergence rate can be expressed as follows:

\[
\frac{1}{Q} \sum_{q=1}^{Q} \sum_{i \in S_q} E[\nabla F^i(\theta_q)] \leq C(\frac{1}{\sqrt{TQ}} + \frac{1}{\sqrt{TQ}} + \frac{1}{\sqrt{TQ}} + \frac{1}{\sqrt{TQ}})
\]


A.2 Part Two

Let us start the proof of RAM-Fed from L-Lipschitzian Condition:

\[
\sum_{i \in S_q} E[\nabla F^i(\theta_q), \theta_{q+1} - \theta_q > := \sum_{i \in S_q} E[\nabla F^i(\theta_q), -y\nabla^2 F^i(\theta_q)]
\]

\[
\leq -\frac{Ty}{8} \sum_{i \in S_q} E[\nabla F^i(\theta_q)]^2
\]

\[
- TyQ \sum_{i \in S_q} \left(32y^2T^2L^2 + 1 + 96y^3T^3 + 3LyT\right)\delta^2
\]

\[
y^2TQ \sum_{N} \left(4yTL + \frac{3}{2} + 12L^2y^2T^2\right)\sigma^2
\]

where \( a \) follows because:

\[
32y^2T^2N^2L^2 \leq \frac{1}{8} \Rightarrow y \leq -\frac{\sqrt{C}}{16TLN}
\]

\[
96y^3T^3N^3 \leq \frac{1}{8} \Rightarrow y \leq -\frac{(F_1)^{1/2}}{768^{1/2}LTN^{1/2}}
\]

\[
\frac{3}{2}LyT \leq \frac{1}{8} \Rightarrow y \leq -\frac{1}{12TL}
\]
bound $T_1$:
\[
\sum_{i \in S_q} E \|\nabla F_i(\theta_q) - \frac{1}{N} \sum_{n=1}^{N} \frac{1}{T} \sum_{t=1}^{T} \nabla F_n^i(\theta_q)\|^2 + \frac{1}{T} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
\leq (32L_2^2 + 256 + 1152N^2L^2 + 48N^2T^2 + 72L^2N^2T^2 + 108N^2F^2T^2)G^2 \\
+ (32y^2T + 12N^2F^2 + 4 + 144N^2L^2y^2T)\sigma^2 \\
+ 128y^2T^2\delta^2 (2 + \frac{N}{F})^2 \\
+ 8w^2(4 + 9N^2L^2)\|\theta_q\|^2 + 72N^2L^2w^2\|\theta_q\|^2
\]

For another term in $L$-Lipschitz condition, we have:
\[
\frac{L}{2} \sum_{i \in S_q} E \|\theta_{q+1}^i - \theta_q^i\|^2 \\
= \frac{1}{2} \sum_{i \in S_q} E \|\nabla F_i(\theta_q)\|^2 - \frac{1}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
+ \frac{1}{T} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
\leq \frac{L}{2} \sum_{i \in S_q} E \|\nabla F_i(\theta_q)\|^2 + \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
+ \frac{1}{T} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
\leq -\frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 + \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2
\]

For another term in $L$-Lipschitz condition, we have:
\[
\frac{L}{2} \sum_{i \in K-S_q} E \|\theta_{q+1}^i - \theta_q^i\|^2 \\
= \frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 - \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
+ \frac{1}{T} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
\leq \frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 + \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
+ \frac{1}{T} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
\leq -\frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 + \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2
\]

For another term in $L$-Lipschitz condition, we have:
\[
\frac{L}{2} \sum_{i \in K-S_q} E \|\theta_{q+1}^i - \theta_q^i\|^2 \\
= \frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 - \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
+ \frac{1}{T} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
\leq \frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 + \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
+ \frac{1}{T} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
\leq -\frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 + \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2
\]

For another term in $L$-Lipschitz condition, we have:
\[
\frac{L}{2} \sum_{i \in K-S_q} E \|\theta_{q+1}^i - \theta_q^i\|^2 \\
= \frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 - \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
+ \frac{1}{T} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
\leq \frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 + \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
+ \frac{1}{T} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
\leq -\frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 + \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2
\]

For another term in $L$-Lipschitz condition, we have:
\[
\frac{L}{2} \sum_{i \in K-S_q} E \|\theta_{q+1}^i - \theta_q^i\|^2 \\
= \frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 - \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
+ \frac{1}{T} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
\leq \frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 + \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
+ \frac{1}{T} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2 \\
\leq -\frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 + \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2
\]

Combining $i \in S_q$ and $i \in K-S_q$:
\[
\sum_{i \in S_q} E \|\nabla F_i(\theta_q)\|^2 + \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 \\
\leq \sum_{i \in S_q} E \|\nabla F_i(\theta_q)\|^2 + \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 \\
\leq -\frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 + \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2
\]

Then, we can obtain:
\[
\sum_{i \in S_q} E \|\nabla F_i(\theta_q)\|^2 + \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 \\
\leq -\frac{L}{2} \sum_{i \in K-S_q} E \|\nabla F_i(\theta_q)\|^2 + \frac{L}{2} \sum_{i \in nS_q} E \|\nabla F_i(\theta_q)\|^2
\]

Letting $\sum_{i \in S_q} \|\theta_q\|^2 = \tau$ and dividing both sides by $\frac{T}{2}$ we get:
\[
\frac{1}{2} \sum_{i \in S_q} E \|\nabla F_i(\theta_q)\|^2 \leq \tau
\]

All proof details are available on Github. \(^3\)

\(^3\)https://github.com/wyy-123-xyy/RA-Fed