Incentivize Contribution and Learn Parameters Too: Federated Learning with Strategic Data Owners

Drashthi Doshi
IIT Bombay
Mumbai, India
drashthi@iitb.ac.in

Aditya Vema Reddy Kesari
IIT Bombay
Mumbai, India
22B3985@iitb.ac.in

Swaprava Nath
IIT Bombay
Mumbai, India
swaprava@iitb.ac.in

Avishek Ghosh
IIT Bombay
Mumbai, India
avishek@cse.iitb.ac.in

Suhas S Kowshik [*]
Microsoft
Bangalore, India
suhaskowshik1992@gmail.com

Abstract

Classical federated learning (FL) assumes that the clients have a limited amount of noisy data with which they voluntarily participate and contribute towards learning a global, more accurate model in a principled manner. The learning happens in a distributed fashion without sharing the data with the center. However, these methods do not consider the *incentive* of an agent for participating and contributing to the process, given that data collection and running a distributed algorithm is costly for the clients. The question of *rationality of contribution* has been asked recently in the literature and some results exist that consider this problem. This paper addresses the question of simultaneous *parameter learning* and *incentivizing contribution*, which distinguishes it from the extant literature. Our first mechanism incentivizes each client to contribute to the FL process at a Nash equilibrium and simultaneously learn the model parameters. However, this equilibrium outcome can be away from the *optimal*, where clients contribute with their *full* data and the algorithm learns the *optimal* parameters. We propose a second mechanism with monetary transfers that is *budget balanced* and enables the full data contribution along with optimal parameter learning. Large scale experiments with real (federated) datasets (CIFAR-10, FeMNIST, and Twitter) show that these algorithms converge quite fast in practice, yield good welfare guarantees, and better model performance for all agents.

1 Introduction

A high quality machine learning model is built when the model is trained on a large amount of data. However, in various practical situations, e.g., for languages, images, disease modeling, such data is split across multiple entities. Moreover, in many applications, data lies in edge devices and a model is learnt when these edge devices interact with a parameter server. Federated Learning (FL) [17, 22, 12, 38] is a recently developed distributed learning paradigm where the edge devices are users' personal devices (like mobile phones and laptops) and FL aims to leverage on-device intelligence. In FL, there is a center (also known as the parameter server) and several agents (edge devices). To learn the model parameters, the edge devices can only communicate through the center. However, a major challenge of federated learning is *data-scarcity*, i.e., owing to its limited storage capacity, the edge devices possesses only a few amounts of data, which may not be sufficient for the

[*] Work done while the author was at Amazon India.

Preprint. Under review.

learning task. Hence, the end user participates in a distributed learning process, where she exploits the data of similar users present in the system.

In the classical federated learning problem, it is typically assumed that the agents participate with all of their data-points *voluntarily*. However, in the presence of rational users, such an assumption may not always hold since sampling data is costly for agents. It may happen that some agents try to contribute very few data-points and try to *exploit* the system by learning based on others' data-points. This phenomenon is called *free-riding* [13] and disincentivizes honest participants to contribute their data to the FL process. Hence, naturally, a few recent works have concentrated on designing incentive mechanisms in federated learning [24, 5, 1].

One of the most successful use-cases of federated learning is in hospital management systems. For instance, in a given geographical region every hospital can sample medical records from its patients for disease modeling. To learn the model reliably, each hospital needs to collect a large amount of data, which is generally very expensive. Instead, they might participate in a *federated learning* (FL) process where different hospitals can train the model locally on their individual datasets but update a consolidated global model parameter. Given that different hospitals have different capabilities in collecting data, it may be best for certain hospitals not to sample any data and therefore not incur any cost, yet get the learning parameters from the FL process if there is a sufficient population in the FL who are sharing their learned parameters. This occurrence of *free-riding* is not ideal, and an important question to ask is:

"Can a mechanism be designed that incentivizes each user to participate and contribute their maximum data in FL and simultaneously learn the optimal model parameters?"

In this paper, we address this question in two stages. First, we consider a utility model of the agents where the accuracy of learning a parameter and the contribution levels of all the agents give every agent some benefit, while their individual contribution levels lead to their personal costs. We propose an algorithm, namely Updated Parameter Best Response Dynamics (`UPBReD`) in the federated learning setting that achieves simultaneous contribution and learning of the model parameters by the agents. However, this method suffers from a sub-optimal contribution by the agents and therefore the global learning is also sub-optimal. In the second stage, we improve this dynamics by allowing monetary transfer where *contributors* and *consumers* are treated differently using the monetary transfer and incentivizes all agents to contribute their maximum amount of data to the learning process. This mechanism, namely Two Phase Updated Parameter Best Response Dynamics (`2P-UPBReD`), learns the optimal parameters for all agents and asks the *data-consumers* to *pay* and *pays* the *data-contributors*, without keeping any *surplus*. `2P-UPBReD` is distinguished in the fact that it simultaneously learns the optimal model parameters, incentivizes full contribution from the agents, yet is quite simple to use in practice.

1.1 Related work

Federated Learning [17] has gained significant attention in the last decade or so. The success story of FL is primarily attributed to the celebrated FedAvg algorithm [22], where one of the major challenges of FL, namely communication cost is reduced by *local steps*. In subsequent works, several other challenges of FL, such as data heterogeneity [15, 9], byzantine robustness [36, 14, 11], communication overhead [28, 16, 10] and privacy [33, 32] were addressed.

Later works focused on incentivizing client participation with advanced aggregation methods. Approaches included dynamic weighting of client updates to ensure that the global model outperforms local models [4], Shapley value to assess contributions [31], contract theory to maximize fairness in data contribution [13], and fairness by eliminating malicious clients and rewarding those who enhance performance of the model [7].

Monetary incentives in FL have been explored through various mechanisms. Yu et al. [37] proposed dynamic budget allocation to maximize utility and reduce inequality. Blockchain-based methods [26] incentivize high-quality data contributions under budget constraints, and client rewards have been shown to improve final model utility [35]. Georgoulaki and Kollias [8] analyzed utility-sharing in FL games, showing a price of anarchy of two and price of stability of one under budget-balanced payments. Auction-based FL systems use reinforcement learning to let agents adjust bids for profit while preserving accuracy [29], and centers can optimize budgets to enhance utility and reduce delays [30].

Coalitional approaches have also been studied. Donahue and Kleinberg [5] modeled FL as a coalitional game allowing joint training, while Ray Chaudhury et al. [27] introduced CoreFed to ensure core stability. Their work was generalized to ordinal utility settings [3], and Shapley value-based approaches ensure reciprocal fairness [23]. However, these do not consider non-cooperative agents choosing data contributions strategically. Repeated-game formulations [20] show that subgame perfect equilibria can be inefficient, prompting budget-balanced mechanisms that ensure social efficiency and individual rationality. Incentive-aware learning frameworks by Blum et al. [1] and their extension with best response dynamics and budget-balance by Murhekar et al. [24] achieve Nash equilibria and maximum welfare under constraints. Yet, these works overlook simultaneous model training, a gap our work addresses by incorporating strategic data contribution and model parameter learning together.

1.2 Our contributions

We consider a strategic federated learning problem where the clients (agents) contribute by sampling data from their data distribution. The quantum of sampling is chosen such that it maximizes their individual *utilities* that consist of two opposing forces: (i) individual benefit and (ii) costs (of data sampling). Moreover, similar to the classical federated learning setting, the center aims to learn an overall model to maximize *total* accuracy (to be defined shortly) simultaneously. Our contributions can be summarized as follows.

- We propose a mechanism called Updated Parameter Best Response Dynamics, `UPBReD` (Algorithm 1) that allows simultaneous learning and contribution by the agents in FL. This mechanism does *not* use any monetary transfers.
- We show that `UPBReD` converges to a *pure strategy Nash equilibrium* (Theorem 1).
- However, the Nash equilibrium can be different from the *socially optimal* outcome where agents contribute with their full dataset $s^{\max}$ and the center learns the optimal model parameters $w^{\texttt{OPT}}$ (Example 1).
- We then propose an updated and cleaner two-phase mechanism namely Two Phase Updated Parameter Best Response Dynamics, `2P-UPBReD` (Algorithm 2) that allows monetary transfers *only* among the agents, i.e., budget balanced. The *data contributors* (those who contribute above average quantities of data) get paid and the *data consumers* (those who contribute below average quantities of data) make the payments. However, both types of agents learn the optimal model parameters.
- We show that the Nash equilibrium for `2P-UPBReD` now shifts to where all agents make their maximum contribution of data $s^{\max}$ and learn the optimal model parameters $w^{\texttt{OPT}}$ (Theorem 2).
- Experiments on real datasets (CIFAR-10, FeMNIST, and Twitter) demonstrate that while `UPBReD` leads to suboptimal social welfare, `2P-UPBReD` achieves performance comparable to FedAvg for all datasets (see Section 5). Notably, FedAvg performs poorly in strategic settings.

2 Preliminaries

Consider a federated learning setup where a set of data contributors, given by $N = \{1, 2, \dots, n\}$, is interacting with a center. Each data contributor (agent) i has access to a private labeled dataset D_i, with the size of the dataset given by $s_i^{\max} = |D_i|$. The agents are interested in learning a parameter vector $w \in \mathbb{R}^m$ from these data so that it helps them predict some unlabeled data accurately (e.g., to perform a classification task). However, each individual agent has a limited amount of data, and learning w only from that data may not be accurate enough. So, they learn this parameter via a *federated learner* such that the model is trained on the consolidated data of all the n users. Assume that the datasets are drawn from the same distribution, e.g., all the agents sample human disease data from a certain geographical location. However, sampling such data is costly and agent i incurs a cost $c_i(s_i)$ when she trains the model locally on her dataset of size $s_i \in S_i := [0, s_i^{\max}]$. Under this setup, each agent i gets a utility based on how much data s_i she decides to sample and train on. Hence, the utility is given by

$$u_i(w, s_i, s_{-i}) = a_i(w, s_i, s_{-i}) - z_i(s_i, s_{-i}), \tag{1}$$

where s_{-i} is the data chosen by the agents other than i in this federated learning process. The function $a_i(w, s_i, s_{-i})$ is the *accuracy function*, which denotes the *benefit* to agent i if the parameter learned by the center is w and the agents contribute by running the federated learning algorithm on their dataset sizes given by the vector $s = (s_i, s_{-i})$.

Remark (Arguments of the u_i function). Note that we have explicitly assumed that the utility function u_i depends on (w, s_i, s_{-i}). We now motivate this dependence. In FL problems, we typically run iterative algorithms to optimize the accuracy (or loss) function. Note that, with this, the weight w^t at time t depends on all the $s_i^\ell, \ell < t, \forall i \in N$. However, the utility u_i at time t depends on (s_i^t, s_{-i}^t) as well, which is not captured through w^t. Hence, we require the said explicit dependencies.

The function $z_i(s_i, s_{-i})$ is the *effective cost* to agent i which may depend on the data contributions of all agents. In the first part of this paper, we consider the case where the cost is only personal to agent i, i.e., z_i is only a function of agent i's chosen data size s_i. In the second part, we assume that this cost may be additionally *taxed* or *subsidized* by the center based on their contributed data-sizes.

Since the agents are strategic, every agent i's aim is to maximize her utility by appropriately choosing her strategy s_i given the strategies s_{-i} of the other players and the parameter w chosen by the center. The center, on the other hand, is interested in learning the parameter w that maximizes the sum of the accuracies as follows, when all agents contribute their maximum data-sizes, i.e., $s_i^{\max}, i \in N$.

$$w^{\texttt{OPT}} \in \operatorname*{argmax}_{w} \sum_{i \in N} a_i(w, s_i^{\max}, s_{-i}^{\max}). \tag{2}$$

Note that the goal of the center does not consider the effective costs of the agents since those are incurred by the agents. We will refer to the term $\sum_{i \in N} a_i(w, s_i, s_{-i})$ as the *social welfare* in this context. We are interested in the question that whether we can design a federated learning algorithm that can make $(s_i^{\max}, s_{-i}^{\max})$ a Nash equilibrium of the underlying game and the center can learn $w^{\texttt{OPT}}$. In this context, the Nash equilibrium is defined as follows.

Definition 1 (Nash equilibrium)**.** *A* pure strategy Nash equilibrium *(PSNE) for a given parameter w is a strategy profile (s_i^*, s_{-i}^*) of the agents such that $u_i(w, s_i^*, s_{-i}^*) \geqslant u_i(w, s_i, s_{-i}^*)$, $\forall s_i \in S_i, \forall i \in N$.*

This definition is a modification of the standard definition of Nash equilibrium since the equilibrium profile (s_i^*, s_{-i}^*) depends on the parameter w (we do not write it explicitly for notational cleanliness). Existence of such a PSNE in this context is obvious due to Nash's theorem [25]. To see this, consider another game Γ' where the pure strategies are to pick 0 or $s_i^{\max}$ for every $i \in N$. Every mixed strategy of Γ' is a pure strategy of the original data contribution game we consider above. Since by Nash's theorem, a mixed strategy Nash equilibrium exists for any finite game, PSNE exists in our game. We will refer to the term PSNE as Nash equilibrium (NE) in the rest of the paper.

We also aim for the following property of a mechanism that involve monetary transfers.

Definition 2 (Budget balance)**.** *A mechanism that uses monetary transfers $p_i(s_i, s_{-i})$ for every $s_i \in S_i, i \in N$ is called* budget balanced *(BB) if $\sum_{i \in N} p_i(s_i, s_{-i}) = 0$.*

This property ensures that the net monetary in or out-flow is zero and the mechanism only allows monetary redistribution among the agents. In the following section, we consider the general case where the effective cost is arbitrary.

3 Learning with arbitrary effective costs

In this section, we consider the effective cost of agent i, i.e., $z_i(s_i, s_{-i})$, to be an arbitrary function of $s = (s_i, s_{-i})$. We know that an alternative interpretation of NE (Definition 1) is a strategy profile (s_i^*, s_{-i}^*) where every agent's *best response* to the strategies of the other players is its own strategy in that profile (see [21, e.g.]), i.e., $s_i^* \in \operatorname{argmax}_{s_i \in S_i} u_i(w, s_i, s_{-i}^*)$, $\forall i \in N$. Hence, an algorithm that simultaneously updates all agents' strategies with the best responses to the current strategies of the other players is called a *best response dynamics* of a strategic form game (see [6, e.g.] for a detailed description). In our problem, this approach cannot be directly employed since the center also needs to learn and update the model parameters w as the agents choose their data contributions. We, therefore, propose a mechanism for federated learning that simultaneously updates both the agents' strategies and the center's choice of w.

In this mechanism, each agent i starts with some initial choices of s_i^0 and shares that with the center.[2] The center also starts with a w^0 and broadcasts that and the entire initial data-contribution vector s^0 with all the agents. In every subsequent iteration t, each agent i locally computes two quantities: (i) her updated contribution s_i^{t+1} by taking one gradient ascent step w.r.t. her own contribution s_i, and (ii) the local gradient of agent i's accuracy component of the social welfare w.r.t. w. Both are evaluated at the current values of w^t and s^t and sent back to the center. The center calculates an updated w^{t+1} that averages (in spirit of the FedAvg algorithm [22]) all the local gradients sent by the agents. The center then shares w^{t+1} and s^{t+1} with all the agents. Formally, the updates are given as follows.

$$s^{t+1} = s^t + \gamma g(w^t, s^t, \mu^t), \tag{3}$$

$$w^{t+1} = w^t + \eta \tilde{g}(w^t, s^t), \tag{4}$$

where the function $\tilde{g}(w^t, s^t) = \frac{1}{n} \sum_{i \in N} \nabla_w a_i(w^t, s^t)$ and g is defined as

$$[g(w^t, s^t, \mu^t)]_i = \frac{\partial}{\partial s_i} u_i(w^t, s^t) + \mu_i^t,$$

$$\text{where } \mu_i^t = \begin{cases} -\frac{\partial}{\partial s_i} u_i(w^t, s^t), & \text{when either } s^t = 0, \frac{\partial}{\partial s_i} u_i < 0 \\ & \text{or } s^t = s_i^{\max}, \frac{\partial}{\partial s_i} u_i > 0 \\ 0. & \text{otherwise} \end{cases}$$

The mechanism is detailed out in Algorithm 1. Our main result of this section is that under certain bounded derivative conditions, Algorithm 1 always converges to a Nash equilibrium. We need a few matrices, defined as follows for every w and s. For $i, j \in N$ and $k, \ell \in \{1, \ldots, m\}$,

$$\begin{aligned} G(w,s)_{ij} &= \frac{\partial^2}{\partial s_j \partial s_i} u_i(w,s); \\ \tilde{G}(w,s)_{k\ell} &= \frac{1}{n} \sum_{i \in N} \frac{\partial^2}{\partial w_\ell \partial w_k} a_i(w,s); \\ H(w,s)_{ik} &= \frac{\partial^2}{\partial w_k \partial s_i} u_i(w,s); \\ \tilde{H}(w,s)_{kj} &= \sum_{i \in N} \frac{\partial^2}{\partial s_j \partial w_k} a_i(w,s). \end{aligned} \tag{5}$$

Assumption 1. *Consider the utility functions given by Equation* (1) *where functions $a_i, z_i, i \in N$ are such that the following properties hold for every $w \in \mathbb{R}^m$ and $s \in \prod_{i \in N} S_i$ (the matrices below are as defined in Equation* (5)*).*

1. *The matrices $G(w,s) + \lambda \mathbb{I}$ and $\tilde{G}(w,s) + \tilde{\lambda} \mathbb{I}$ are negative semi-definite.*
2. *We assume the following bounds: (i) $\forall i, j \in N, |G(w,s)_{ij}| \leqslant L$, (ii) $\forall k, \ell \in \{1, \ldots, m\}, |\tilde{G}(w,s)_{k\ell}| \leqslant \tilde{L}$, (iii) $\|H(w,s)\|_{op} \leqslant P$, where $\|A\|_{op} := \inf\{c \geqslant 0 : \|Av\| \leqslant c\|v\|, \forall v\}$, (iv) $\|\tilde{H}(w,s)\|_{op} \leqslant \tilde{P}$.*

Note that the definition of G and $\tilde{G}$ do not imply the strong concavity of u_i or a_i. Moreover, the above assumptions are standard and have appeared in the literature before (see [24]). We define the following expressions for a cleaner presentation.

$$\begin{aligned} W_1 &= \sqrt{1 + \gamma^2 n^2 L^2 - 2\gamma\lambda} + \sqrt{\tilde{P}^2 \gamma^2}, \\ W_2 &= \sqrt{1 + \eta^2 m^2 \tilde{L}^2 - 2\eta\tilde{\lambda}} + \sqrt{P^2 \eta^2}, \\ E &= \|g(w^0, s^0, \mu^0)\|_2 + \|\tilde{g}(w^0, s^0)\|_2, \quad \text{where } w^0, s^0 \text{ are arbitrary, and} \\ T_0(w^0, s^0) &= \left(\ln \frac{E}{\epsilon}\right) \Big/ \left(\ln \frac{1}{W}\right), \quad \text{where } W = \max\{W_1, W_2\}. \end{aligned} \tag{6}$$

[2] Note that, only the number of data points, s_i is shared with the center and not the data, which is consistent with the principle of federated learning.

We are now ready to present the main result of this section. Owing to the space constraints, all proofs of the technical results are presented in the Appendix.

Theorem 1 (`UPBReD` convergence to NE)**.** *Under Assumption 1,* `UPBReD` *(Algorithm 1) converges to a Nash equilibrium. Formally, for a given $\epsilon > 0$, for every initial value (w^0, s^0) of Algorithm 1, the gradients $\|g(w^T, s^T, \mu^T)\| < \epsilon$ and $\|\tilde{g}(w^T, s^T)\| < \epsilon$, for all $T \geqslant T_0(w^0, s^0)$, when the step sizes are chosen as follows:*

$$\gamma < \min\left\{1, \frac{1}{\tilde{P}}, \frac{2\lambda}{n^2L^2}, \frac{\lambda - \tilde{P}}{n^2L^2 - \tilde{P}^2}\right\}, \text{ given } \lambda > \tilde{P},$$

$$\eta < \min\left\{1, \frac{1}{P}, \frac{2\tilde{\lambda}}{m^2\tilde{L}^2}, \frac{\tilde{\lambda} - P}{m^2\tilde{L}^2 - P^2}\right\}, \text{ given } \tilde{\lambda} > P.$$

Algorithm 1: Updated Parameter Best Response Dynamics (`UPBReD`)

```
Input: Step size $\gamma, \eta$, initialization $w^0$, $s_i^0$ for $i \in N$, number of iterations $T$
Output: $w^T$
for $t = 0$ to $T - 1$ do
    Center broadcasts $w^t, s^t$
    foreach agent $i \in N$ in parallel do
        $s^{t+1} = s^t + \gamma g(w^t, s^t, \mu^t)$
        Compute local gradient: $d_i^{t+1} = \nabla_w \, a_i(w^t, s_i^t, s_{-i}^t)$
        Send $s_i^{t+1}, d_i^{t+1}$ to the center
    Center updates: $w^{t+1} = w^t + \frac{\eta}{n} \sum_{i \in N} d_i^{t+1} = w^t + \eta\tilde{g}(w^t, s^t)$
return $w^T$
```

The theorem provides a guarantee of convergence of Algorithm 1 for any arbitrary initial condition (w^0, s^0). It needs to run for a minimum number of iterations $T_0(w^0, s^0)$, with appropriate parameters of γ and η that govern the gradient ascent rates of the agents' and center's objectives respectively. Note that for a better convergence, i.e., a smaller ϵ, the algorithm needs to run longer. The parameter W determines the contraction factor of the recurrence of $\|g(w^t, s^t)\|_2 + \|\tilde{g}(w^t, s^t)\|_2$ and is chosen to be smaller than unity by choosing γ and η appropriately. Appendix A.2 shows an example of such a feasible region.

Remark. Note that the choice of step sizes γ and ν scale inversely with m and n respectively. Moreover, we impose certain restrictions on λ and $\tilde{\lambda}$, which may be restrictive. In Section 4, we remove such issues and propose a learning algorithm with choices of γ and ν that do not scale with m or n.

Remark. We emphasize that in experiments, sufficiently small and constant values of γ and ν work. Hence, the implicit choice obtained in the above theorem is an artifact of our theoretical analysis, and is not a reflection of practical scenarios.

In Appendix A.3, we provide an example of an application of this setting where the NE is different from the social optima. Therefore, it is imperative to ask for a method where the social optima can be obtained in a NE. We address this in the following section.

4 Welfare maximization with monetary transfers

In this section, we consider the scenario where the center can make a payment $p_i(s_i, s_{-i})$ *to* agent i to alter her utility function. Therefore, the effective cost of agent i becomes $z_i(s_i, s_{-i}) = c_i(s_i) - p_i(s_i, s_{-i})$. Note that, the payment can be either positive or negative, which determines whether the agent is *subsidized* or *taxed* respectively. It is reasonable to expect that the *data contributors* of the federated learning process are subsidized for their data contribution while the *data consumers* are charged payment for obtaining the learned parameter. We assume that the derivatives of the accuracy a_i and cost function c_i are bounded. We choose the following payment function:

$$p_i(s) = \beta(s_i - \frac{1}{n-1}\sum_{j \neq i} s_j), \tag{7}$$

where β is a parameter of choice. Note that this payment mechanism is budget balanced by design, since $\sum_{i \in N} p_i(s_i, s_{-i}) = 0, \forall s_i \in S_i, s_{-i} \in S_{-i}$. With payment, the utility of agent i becomes:

$$u_i(w, s) = a_i(w, s) - c_i(s_i) + \beta(s_i - \frac{1}{n-1} \sum_{j \neq i} s_j). \tag{8}$$

In this section, we are also interested in the *quality* of the NE in terms of social welfare and want to reach the desired NE where $s_i^* = s_i^{\max}, \forall i \in N$ and $w^* = w^{\texttt{OPT}}$.

4.1 Convergence to the NE $(w^{\text{OPT}}, s^{\max})$

In this setup, we first prove that the utility of every agent i can be made strictly increasing in her own contribution s_i in the following manner.

Assumption 2. *The derivative of $a_i(w, s)$ with respect to s_i is bounded away from $-\infty$, i.e., $\frac{\partial}{\partial s_i} a_i(w, s) \geqslant -\tau$ for some $0 \leqslant \tau < \infty$, for all $i \in N, w$, and s. Moreover, the cost functions $c_i(.)$ has bounded derivatives, i.e., $c_i'(s_i) \leqslant \zeta$, for all s_i and i.*

Note that the above assumptions are pretty mild. Previous works assume even stronger assumptions (like convexity) on a_i and c_i (see [24]). We show that the above bounds on the derivatives of a_i and c_i are sufficient to claim our result.

Lemma 1 (Increasing utility)**.** *Suppose Assumption 2 holds and we use the payment function given by Equation* (7)*. Then, the utility of every agent is increasing in her own data contribution, i.e., $\frac{\partial}{\partial s_i} u_i(w, s_i, s_{-i}) > 0, \forall s_i \in (0, s_i^{\max}), i \in N,$ and w provided $\beta > \zeta + \tau$.*

Remark (Knowledge of β). The knowledge of β is required for theoretical tractability only. In the experiments, we do not assume any knowledge of β, rather we tune this hyperparameter. We observe that uniformly across all datasets, if β is chosen larger than a threshold, we obtain the best performance of our proposed algorithms which also validates the above theoretical requirement.

This lemma implies that even in Algorithm 1 if we apply the above payment, s will converge to the maximum value $s^{\max}$. However, unlike Algorithm 1, in this section we provide an algorithm which gives the step sizes of the gradient ascents in a more concrete manner and is, therefore, a superior one. In order to keep the convergence rate same as Theorem 1, we assume that the negative[3] social welfare function at $s^{\max}$, given by $f(w, s^{\max}) = -\frac{1}{n} \sum_{i \in N} a_i(w, s^{\max})$, is M-smooth and ν-strictly convex in w. These properties are formally defined below.

Definition 3 (M-smoothness)**.** *A function $f : \mathbb{R}^m \to \mathbb{R}$ is M-smooth if for all $x, x' \in \mathbb{R}^m$, we have $f(x') \leqslant f(x) + \langle \nabla_x f(x), x' - x \rangle + \frac{M}{2} \|x - x'\|_2^2$.*

Definition 4 (ν-strictly convex)**.** *A function $f : \mathbb{R}^m \to \mathbb{R}$ is ν-strictly convex if for all $x, x' \in \mathbb{R}^m$, we have $f(x') \geqslant f(x) + \langle \nabla_x f(x), x' - x \rangle + \frac{\nu}{2} \|x - x'\|_2^2$.*

[3] We negate the welfare so that we can consider functions as convex to apply the results of convex analysis easily.

Algorithm 2: Two Phase Updated Parameter Best Response Dynamics (`2P-UPBReD`)

```
Input: Step size $\gamma, \eta$, initialization $w^0$, $s_i^0$ for $i \in N$, number of iterations $T$
Output: $w^T$
Center broadcasts $w^0, s^0$, set $t = 0$                               /* Begin phase 1 */
while $s \neq s^{\max}$do
    Center broadcasts $s^t$
    for agent $i \in N$ in paralleldo
        $s_i^{t+1} = s_i^t + \gamma[g(w^0, s^t, \mu^t)]_i$
        Send $s_i^{t+1}$ to the center
    $t = t + 1$                                                          /* End phase 1 */
for $t = 0$ to $T$do                                                     /* Begin phase 2 */

    Center broadcasts $w^t$
    for agent $i \in N$ in paralleldo
        Compute local gradient: $d_i^{t+1} = \nabla_w\, a_i(w^t, s_i^{\max}, s_{-i}^{\max})$
        Send $d_i^{t+1}$ to the center
    Center updates: $w^{t+1} = w^t + \frac{\eta}{n} \sum_{i \in N} d_i^{t+1} = w^t + \eta \tilde{g}(w^t, s^{\max})$
return $w^T$                                                             /* End phase 2 */
```

We propose a two-phase algorithm given by Algorithm 2. The algorithm, in the first phase, incentivizes the agents to contribute $s^{\max}$, and in the second phase, converges to $w^{\texttt{OPT}}$. Note that in the first phase of Algorithm 2, every agent runs a gradient ascent step. Thanks to the increasing utility property (Lemma 1), we have an increasing sequence of s_i^t for all $i \in N$. Moreover, from the definition of $[g(w^0, s^t, \mu^t)]_i$ it is ensured that $s_i^{\max}$ is the fixed point of this gradient ascent update. Hence after a finite number of iterations, every agent reaches $s_i^{\max}$. Note that in this phase the model parameter w^0 remains unchanged.

In the second phase of the algorithm, we update the model parameter w^t. Note that since all the agents contribute the maximum amount of data they own, this phase is simply *pure* federated learning (without incentive design). As such, each agent now computes the local gradient d_i^{t+1} and sends it to the center. The center then aggregates the gradients and takes a gradient ascent step with learning rate η. The center then broadcasts the updated parameter to the agents and the process continues.

We now provide the convergence guarantees of Algorithm 2. Before that, let us discuss the assumptions as discussed in Appendix A.3.

Assumption 3. *The negative social welfare function at $s^{\max}$, $f(w, s^{\max})$, is M-smooth and ν-strictly convex in w, with $M > \nu$.*

The smoothness and strong convexity assumptions have featured in several previous papers on FL [15, 36]. The main result of this section is as follows.

Theorem 2 (`2P-UPBReD` convergence to the optimal NE)**.** *Suppose Assumptions 2 and 3 hold and we consider the utility function with payment scheme given in Equation* (8) *with $\beta > \zeta + \tau$. Then, for every $w \in \mathbb{R}^m$ and $s \in \prod_{i \in N} S_i$,* `2P-UPBReD` *(Algorithm 2) converges to the Nash equilibrium $(w^{\texttt{OPT}}, s^{\max})$ and is budget balanced. In particular, for any given $\epsilon > 0$, we have*

$$\frac{1}{n} \sum_{i \in N} a_i(w^{\texttt{OPT}}, s^{\max}) - \frac{1}{n} \sum_{i \in N} a_i(w^T, s^{\max}) < \epsilon$$

and $s^T = s^{\max}$ in $T = \kappa + T_0$ iterations provided we choose $\gamma = c$ (an universal constant) and $\eta = 1/M$ with

$$\kappa \geqslant \max_i \left\{ \frac{s_i^{\max} - s_i^0}{c\,\Delta_i} \right\}, \qquad T_0 > \left(\ln \frac{f(w^0, s^{\max}) - f(w^{\texttt{OPT}}, s^{\max})}{\epsilon} \right) \Big/ \left(\ln \frac{1}{1 - \frac{\nu}{M}} \right) \tag{9}$$

where, $f(w, s) := -\frac{1}{n} \sum_{i \in N} a_i(w, s)$ and $\Delta_i = \beta - \tau - \zeta$.

Remark (Strong convexity and smoothness). The convergence rate is $\mathcal{O}(\ln {}^1\!/\!_\epsilon)$ which is the same as that of Algorithm 1. However, Algorithm 2 is cleaner in terms of the phases of convergence, the choices of the step sizes γ, η. The structural assumptions like strong convexity and smoothness may be relaxed at the expense of weaker rates. For example, we observe convergence rates of $\mathcal{O}(\frac{1}{\sqrt{\epsilon}})$ for non-convex smooth functions and $\mathcal{O}(\frac{1}{\epsilon})$ for convex smooth functions.

Remark (γ and rationality). Phase 1 of Algorithm 2 achieves the optimal contribution $s^{\max}$ without updating the learning parameter w. The speed of convergence of this phase depends on the parameter $\gamma = c$, which can be interpreted as the minimum *rationality* level of the society. For instance, if all agents are sufficiently rational, i.e., c is large, they may converge to $s^{\max}$ in a single iteration. However, the algorithm converges even if agents are boundedly rational. In Phase 2 of the algorithm, the agents stop updating their contributions and only updates the gradients d_i and the center accumulates them to update w^t.

Thanks to the smoothness and strong convexity properties of $f(\cdot, s^{\max})$, we can show that running Algorithm 2 guarantees that the model iterate w^T converges to $w^{\texttt{OPT}}$ at an exponential speed as well.

Corollary 1 (Iterate Convergence)**.** *With the same setup as above and $\gamma = c$ (universal constant), $\eta = \frac{2}{M+\nu}$, we obtain $\|w^T - w^{\texttt{OPT}}\|_2 < \epsilon$, and $s^T = s^{\max}$, where $T = \kappa + \tilde{T}_0$, with the same κ as in Equation* (9) *and $\tilde{T}_0 > \left(\ln \frac{\|w^0 - w^{\texttt{OPT}}\|_2}{\epsilon}\right) / \left(\ln \frac{1+\frac{\nu}{M}}{1-\frac{\nu}{M}}\right)$.*

5 Experiments

Our results in Sections 3 and 4 guarantee convergence of Algorithms 1 and 2 to a pure Nash equilibrium under the given assumptions. However, it is important to ask how these FL algorithms behave in practice with real datasets under more general setups. We capture results in these realistic settings in our experiments. We train models on the CIFAR-10 [18], FeMNIST, and Twitter datasets [2] and compare the social welfare of `UPBReD`, `2P-UPBReD`, and `FedAvg`. We also consider another algorithm `FedAvgStrategic`. In this algorithm, the agents choose to contribute their data strategically as in `UPBReD`, and stick to this contribution in the entire training phase of `FedAvg` (details in Appendix C.1). The algorithm proposed in [24], which is closest to this paper, do not consider the simultaneous learning of parameters and strategy updates, and is therefore not meaningful to compare against.

The cost function used is linear $c_i(s_i) = c_i s_i$, where c_i is sampled from $U[0, 1]$ for each agent. We choose our accuracy function to be $a_i(w^k, s^k) = r_i - \mathcal{L}_i(w^k, s^k)$ where the $\mathcal{L}_i(w^k, s^k)$ is the loss evaluated on agents i's test dataset at the k-th round of training. The social welfare is given by $\sum_{i \in N} a_i(w^k, s^k)$. We perform strategy updates using the numerical derivative as follows:

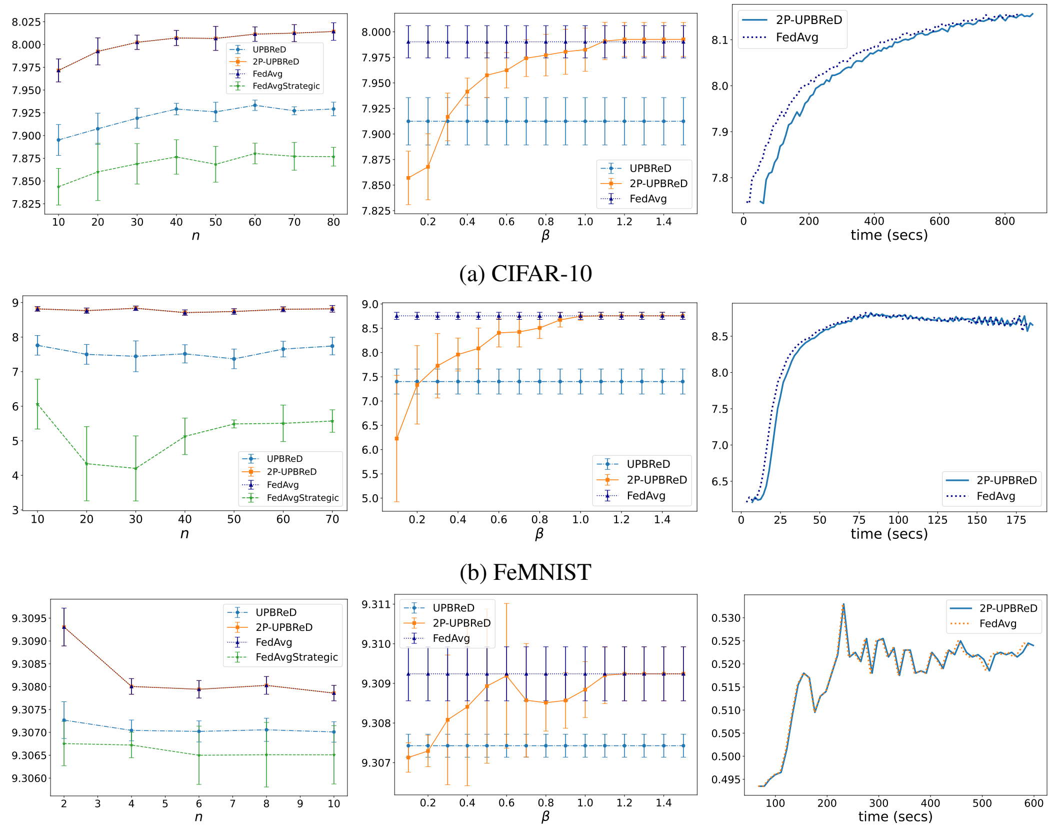
$$s_i^{k+1} = s_i^k - \frac{L_i(w_i^k) - L_i(w^{k-1})}{s_i^k - s_i^{k-1}} - c_i + \beta. \tag{10}$$

The model parameters obtained by agent i after performing one training step locally on the model initialized with w^{k-1} is denoted by w_i^k. We use cross-entropy loss function for $\mathcal{L}$. All the experiments are performed on a *NVIDIA RTX A5000* with a 24GB core and a *NVIDIA RTX A6000* with a 48GB core. The experiments required to obtain the results we report take several days to run. For comparing social welfare against β and number of agents, we randomize over 10 runs, sampling an integer s_i^0 from $U[s_i^{max}/3, 2s_i^{max}/3]$ and cost from $U[0, 1]$. We plot the average value of time taken across 10 runs.

Dataset specifics. For CIFAR-10, we distribute the data equally among 100 agents by sampling the data uniformly at random. We train a CNN with 1,250,858 parameters for 100 rounds with agents updating their training datasets using the best response strategy in each round for `UPBReD`. We choose $\gamma = 0.5$ and $\beta = 2$. We use the Adam optimizer with learning rate 0.001.

For FeMNIST, we use 20% of the dataset that generates users with data that is *not identically distributed* as per the codebase provided by [2]. We train a CNN with $6,603,710$ parameters for 100 rounds. We choose $\gamma = 0.1$ and $\beta = 2$. We use the Adam optimizer with learning rate 0.001.

For the Twitter dataset, we use 15% of the dataset that generates users with data that is *identically distributed* as per the codebase provided by [2]. We further group 1000 agents together to create agents with larger data shares. We use an LSTM network to train for 50 rounds. We choose $\gamma = 0.5$ and $\beta = 2$. We use the Adam optimizer with learning rate 0.00001.



(a) CIFAR-10

(b) FeMNIST

(c) Twitter

Figure 1: The three rows in this figure correspond to the experiments with the datasets CIFAR-10, FeMNIST, and Twitter respectively. The `y-axis` in every plot is the social welfare. The `x-axis` corresponds to the number of agents in the first plot, the choice of β in the second, and the time taken (in seconds) in the third plot respectively in every row.

All results are summarized in Figure 1. We provide more details of the model architecture for all models in Appendix C.2. We observe that `2P-UPBReD` and `FedAvg` have the same social welfare performance w.r.t. the number of agents (first column) and running times are nearly the same for all the datasets (third column). `2P-UPBReD` outperforms `UPBReD` in nearly all the runs with respect to social welfare. We also observe that `2P-UPBReD` converges to the maximum contribution of the agents for values of β that are lower than the theoretically predicted ones. We also note that the data contributors get paid and data consumers pay in `2P-UPBReD` and their payments are monotone non-decreasing with data contribution (Figure 2). The utilities vary because the dataset on which the model is tested to obtain the accuracy varies for the agents.

6 Conclusions and future work

In this paper, we proposed `UPBReD`, which ensures convergence to a Nash equilibrium along with the model parameters w. However, w may deviate from the optimal $w^{\texttt{OPT}}$, so we introduced a two-phase mechanism, `2P-UPBReD`, that guarantees full agent contribution and convergence to $w^{\texttt{OPT}}$. This comes at the cost of requiring monetary transfers, though these can be internal among agents.

An immediate direction for future work is to incorporate budget constraints for agents, limiting the maximum payment each can make while preserving approximate guarantees. Another important extension is to handle heterogeneous data distributions, allowing agents to form clusters and learn optimal parameters within each cluster.

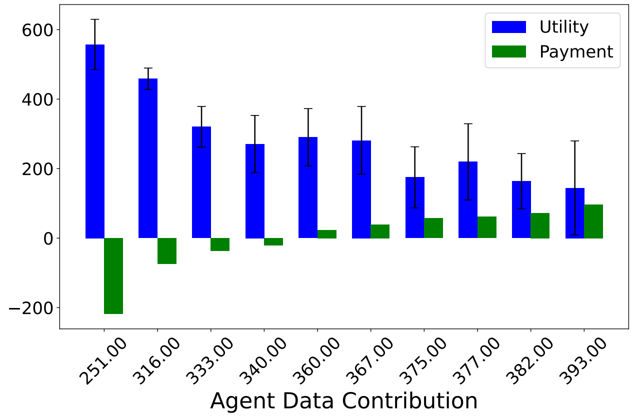


Figure 2: Utilities and payments received by agents for 2P-UPBReD based on their data contributions for FeMNIST (averaged over 10 runs).

Acknowledgments

This work has been primarily supported by an Amazon grant (RD/0123-AMAZO09-005) and AG likes to acknowledge that. Moreover, SN likes to also acknowledge the supports of a Core Research Grant (CRG/2023/001442) from SERB, Govt. of India, a TCS Research grant (RD/0123-TCSIL18-001), and an IIT Bombay grant (2022058).

References

[1] Avrim Blum, Nika Haghtalab, Richard Lanas Phillips, and Han Shao. 2021. One for One, or All for All: Equilibria and Optimality of Collaboration in Federated Learning. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 1005–1014. https://proceedings.mlr.press/v139/blum21a.html

[2] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2019. LEAF: A Benchmark for Federated Settings. arXiv:1812.01097 [cs.LG] https://arxiv.org/abs/1812.01097

[3] Bhaskar Ray Chaudhury, Aniket Murhekar, Zhuowen Yuan, Bo Li, Ruta Mehta, and Ariel D. Procaccia. 2024. Fair Federated Learning via the Proportional Veto Core. In *Forty-first International Conference on Machine Learning*. https://openreview.net/forum?id=6Zgjrowepn

[4] Yae Jee Cho, Divyansh Jhunjhunwala, Tian Li, Virginia Smith, and Gauri Joshi. 2022. To Federate or Not To Federate: Incentivizing Client Participation in Federated Learning. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*. https://openreview.net/forum?id=pG08eM0CQba

[5] Kate Donahue and Jon Kleinberg. 2021. Model-sharing Games: Analyzing Federated Learning Under Voluntary Participation. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 6 (May 2021), 5303–5311. https://doi.org/10.1609/aaai.v35i6.16669

[6] Drew Fudenberg. 1991. *Game theory*. MIT press.

[7] Liang Gao, Li Li, Yingwen Chen, Wenli Zheng, ChengZhong Xu, and Ming Xu. 2021. FIFL: A Fair Incentive Mechanism for Federated Learning. In *Proceedings of the 50th International Conference on Parallel Processing* (Lemont, IL, USA) *(ICPP '21)*. Association for Computing Machinery, New York, NY, USA, Article 82, 10 pages. https://doi.org/10.1145/3472456.3472469

[8] Eirini Georgoulaki and Kostas Kollias. 2023. Arbitrary Profit Sharing in Federated Learning Utility Games. In *Algorithmic Game Theory*, Argyrios Deligkas and Aris Filos-Ratsikas (Eds.). Springer Nature Switzerland, Cham, 58–70.

[9] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems* 33 (2020), 19586–19597.

[10] Avishek Ghosh, Raj Kumar Maity, Swanand Kadhe, Arya Mazumdar, and Kannan Ramachandran. 2020. Communication efficient and byzantine tolerant distributed learning. In *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2545–2550.

[11] Avishek Ghosh, Raj Kumar Maity, Swanand Kadhe, Arya Mazumdar, and Kannan Ramchandran. 2021. Communication-efficient and byzantine-robust distributed learning with error feedback. *IEEE Journal on Selected Areas in Information Theory* 2, 3 (2021), 942–953.

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

[13] Sai Praneeth Karimireddy, Wenshuo Guo, and Michael I Jordan. 2022. Mechanisms that incentivize data sharing in federated learning. *arXiv preprint arXiv:2207.04557* (2022).

[14] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. 2020. Byzantine-robust learning on heterogeneous datasets via bucketing. *arXiv preprint arXiv:2006.09365* (2020).

[15] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*. PMLR, 5132–5143.

[16] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. 2019. Error feedback fixes signsgd and other gradient compression schemes. In *International Conference on Machine Learning*. PMLR, 3252–3261.

[17] Jakub Konečnỳ. 2016. Federated Learning: Strategies for Improving Communication Efficiency. *arXiv preprint arXiv:1610.05492* (2016).

[18] Alex Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images. `https://api.semanticscholar.org/CorpusID:18268744`

[19] Xiao-Bai Li and Srinivasan Raghunathan. 2014. Pricing and disseminating customer data with privacy awareness. *Decision Support Systems* 59 (2014), 63–73. `https://doi.org/10.1016/j.dss.2013.10.006`

[20] Wuxing Mao, Qian Ma, Guocheng Liao, and Xu Chen. 2024. Game Analysis and Incentive Mechanism Design for Differentially Private Cross-Silo Federated Learning. *IEEE Transactions on Mobile Computing* 23, 10 (2024), 9337–9351. `https://doi.org/10.1109/TMC.2024.3364372`

[21] Michael Maschler, Shmuel Zamir, and Eilon Solan. 2020. *Game theory*. Cambridge University Press.

[22] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[23] Aniket Murhekar, Bhaskar Ray Chaudhary, Jiaxin Song, and Ruta Mehta. 2024. You get what you give: Reciprocal fair federated learning. *Under submission* (2024).

[24] Aniket Murhekar, Zhuowen Yuan, Bhaskar Ray Chaudhury, Bo Li, and Ruta Mehta. 2023. Incentives in Federated Learning: Equilibria, Dynamics, and Mechanisms for Welfare Maximization. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[25] John F Nash Jr. 1950. Equilibrium points in n-person games. *Proceedings of the national academy of sciences* 36, 1 (1950), 48–49.

[26] Shashi Raj Pandey, Lam Nguyen, and Petar Popovski. 2022. FedToken: Tokenized Incentives for Data Contribution in Federated Learning. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*. `https://openreview.net/forum?id=u8eB_1q_LRX`

[27] Bhaskar Ray Chaudhury, Linyi Li, Mintong Kang, Bo Li, and Ruta Mehta. 2022. Fairness in Federated Learning via Core-Stability. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 5738–5750. `https://proceedings.neurips.cc/paper_files/paper/2022/file/25e92e33ac8c35fd49f394c37f21b6da-Paper-Conference.pdf`

[28] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. 2018. Sparsified SGD with memory. *Advances in neural information processing systems* 31 (2018).

[29] Xiaoli Tang and Han Yu. 2023. Competitive-Cooperative Multi-Agent Reinforcement Learning for Auction-based Federated Learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, Edith Elkind (Ed.). International Joint Conferences on Artificial Intelligence Organization, 4262–4270. `https://doi.org/10.24963/ijcai.2023/474` Main Track.

[30] Xiaoli Tang and Han Yu. 2024. Multi-Session Multi-Objective Budget Optimization for Auction-based Federated Learning. In *2024 International Joint Conference on Neural Networks (IJCNN)*. 1–8. `https://doi.org/10.1109/IJCNN60899.2024.10650637`

[31] Nurbek Tastan, Samar Fares, Toluwani Aremu, Samuel Horváth, and Karthik Nandakumar. 2024. Redefining Contributions: Shapley-Driven Federated Learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 5009–5017. `https://doi.org/10.24963/ijcai.2024/554` Main Track.

[32] Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. 2020. LDP-Fed: Federated learning with local differential privacy. In *Proceedings of the third ACM international workshop on edge systems, analytics and networking*. 61–66.

[33] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE transactions on information forensics and security* 15 (2020), 3454–3469.

[34] Stephen J. Wright and Benjamin Recht. 2022. *Optimization for Data Analysis*. Cambridge University Press.

[35] Yunchao Yang, Yipeng Zhou, Miao Hu, Di Wu, and Quan Z. Sheng. 2023. BARA: Efficient Incentive Mechanism with Online Reward Budget Allocation in Cross-Silo Federated Learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, Edith Elkind (Ed.). International Joint Conferences on Artificial Intelligence Organization, 4478–4485. `https://doi.org/10.24963/ijcai.2023/498` Main Track.

[36] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*. Pmlr, 5650–5659.

[37] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. 2020. A Fairness-aware Incentive Scheme for Federated Learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (New York, NY, USA) *(AIES '20)*. Association for Computing Machinery, New York, NY, USA, 393–399. `https://doi.org/10.1145/3375627.3375840`

[38] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. 2021. A survey on federated learning. *Knowledge-Based Systems* 216 (2021), 106775.

A Appendix to Section 3

A.1 Proof of Theorem 1

Consider the first-order Taylor expansions of g and $\tilde{g}$ given as follows.

$$\begin{aligned} g(w^{t+1}, s^{t+1}, \mu^{t+1}) &= g(w^t, s^t, \mu^t) + G(w^t, s', \mu^t) \cdot (s^{t+1} - s^t) \\ &\quad + H(w', s^t, \mu^t)(w^{t+1} - w^t), \\ \tilde{g}(w^{t+1}, s^{t+1}) &= \tilde{g}(w^t, s^t) + \tilde{G}(w', s^t) \cdot (w^{t+1} - w^t) \\ &\quad + \tilde{H}(w^t, s')(s^{t+1} - s^t), \end{aligned}$$

where $s' = \theta_s s^t + (1-\theta_s)s^{t+1}, w' = \theta_w w^t + (1-\theta_w)w^{t+1}$, for some $\theta_s, \theta_w \in [0,1]$.

Using the update rules of Algorithm 1 given by Equation (3) and Equation (4), we get

$$\begin{aligned} g(w^{t+1}, s^{t+1}, \mu^{t+1}) &= g(w^t, s^t, \mu^t) + G(w^t, s', \mu^t) \cdot \gamma g(w^t, s^t, \mu^t) \\ &\quad + H(w', s^t, \mu^t)\eta \tilde{g}(w^t, s^t) \\ \tilde{g}(w^{t+1}, s^{t+1}) &= \tilde{g}(w^t, s^t) + \tilde{G}(w', s^t) \cdot \eta \tilde{g}(w^t, s^t) \\ &\quad + \tilde{H}(w^t, s')\gamma g(w^t, s^t, \mu^t) \end{aligned}$$

Using triangle inequality on each of these identities, we get

$$\begin{aligned} \|g(w^{t+1}, s^{t+1}, \mu^{t+1})\|_2 &\leqslant \|(I_{n\times n} + \gamma G(w^t, s', \mu^t))g(w^t, s^t, \mu^t)\|_2 \\ &\quad + \eta \|H(w', s^t, \mu^t)\tilde{g}(w^t, s^t)\|_2 \\ \|\tilde{g}(w^{t+1}, s^{t+1})\|_2 &\leqslant \|(I_{m\times m} + \eta \tilde{G}(w', s^t))\tilde{g}(w^t, s^t)\|_2 \\ &\quad + \gamma \|\tilde{H}(w^t, s')g(w^t, s^t, \mu^t)\|_2 \end{aligned} \tag{11}$$

From condition 1 of Assumption 1, we get that $v^\top(G + \lambda I_{n\times n})v \leqslant 0, \forall v \in \mathbb{R}^n$ and $v'^\top(\tilde{G} + \lambda I_{m\times m})v' \leqslant 0, \forall v' \in \mathbb{R}^m$. In particular, for $v = g(w^t, s^t, \mu^t)$ and $v' = \tilde{g}(w^t, s^t)$, we get

$$\begin{aligned} g(w^t, s^t, \mu^t)^\top G(w^t, s', \mu^t) g(w^t, s^t, \mu^t) &\leqslant -\lambda \|g(w^t, s^t, \mu^t)\|_2^2 \\ \tilde{g}(w^t, s^t)^\top \tilde{G}(w', s^t)\tilde{g}(w^t, s^t) &\leqslant -\tilde{\lambda}\|\tilde{g}(w^t, s^t)\|_2^2 \end{aligned} \tag{12}$$

Consider the square of the first term of the RHS of the inequality of g in Equation (11)

$$\begin{aligned} &\|(I_{n\times n} + \gamma . G(w^t, s', \mu^t))g(w^t, s^t, \mu^t)\|^2 \\ &= \|g(w^t, s^t, \mu^t)\|_2^2 + \gamma^2 . \|G(w^t, s^t)g(w^t, s^t, \mu^t)\|_2^2 \\ &\quad + 2\gamma g(w^t, s^t, \mu^t)^\top G(w^t, s', \mu^t) g(w^t, s^t, \mu^t) \\ &\leqslant \|g(w^t, s^t, \mu^t)\|_2^2 + \gamma^2 . n^2 L^2 \|g(w^t, s^t, \mu^t)\|_2^2 - 2\gamma\lambda \|g(w^t, s^t, \mu^t)\|_2^2 \\ &= (1 + \gamma^2 . n^2 L^2 - 2\gamma\lambda)\|g(w^t, s^t, \mu^t)\|_2^2 \end{aligned} \tag{13}$$

where the equality comes by expanding the squared norm and the inequality comes from the facts that (i) $\|G(w^t, s^t)g(w^t, s^t, \mu^t)\|_2^2 \leqslant \|G(w^t, s^t)\|_F^2 \|g(w^t, s^t, \mu^t)\|_2^2$, where $\|A\|_F := \sqrt{\sum_i \sum_j |A_{ij}|^2}$ is the Frobenius norm of a matrix A and (ii) using Equation (12). By condition 2 of Assumption 1, $\|G(w^t, s^t)\|_F^2 \leqslant n^2 L^2$. Hence, we get

$$\begin{aligned} &\|(I_{n\times n} + \gamma . G(w^t, s^t))g(w^t, s^t, \mu^t)\| \\ &\leqslant \sqrt{1 + \gamma^2 . n^2 L^2 - 2\gamma\lambda} \cdot \|g(w^t, s^t, \mu^t)\|_2 \end{aligned}$$

given the term inside the square root is positive.

Consider the second term of the RHS of the inequality of g in Equation (11), where

$$\begin{aligned} \|H(w', s^t, \mu^t)\tilde{g}(w^t, s^t)\|_2^2 &\leqslant \|H(w', s^t, \mu^t)\|_{op}^2 \|\tilde{g}(w^t, s^t)\|_2^2 \\ &\leqslant P^2 \|\tilde{g}(w^t, s^t)\|_2^2 \end{aligned}$$

by condition 2 of Assumption 1.

Defining $\alpha := 1 + \gamma^2.n^2L^2 - 2\gamma\lambda, \beta := P^2\eta^2, \tilde{\alpha} := 1 + \eta^2.m^2\tilde{L}^2 - 2\eta\tilde{\lambda}, \tilde{\beta} := \tilde{P}^2\gamma^2$, and carrying out a similar analysis for $\tilde{g}$ in Equation (11), we get

$$\begin{aligned} \|g(w^{t+1}, s^{t+1}, \mu^{t+1})\|_2 &\leqslant \sqrt{\alpha}\|g(w^t, s^t, \mu^t)\|_2 + \sqrt{\beta}\|\tilde{g}(w^t, s^t)\|_2 \\ \|\tilde{g}(w^{t+1}, s^{t+1})\|_2 &\leqslant \sqrt{\tilde{\alpha}}\|\tilde{g}(w^t, s^t)\|_2 + \sqrt{\tilde{\beta}}\|g(w^t, s^t, \mu^t)\|_2 \end{aligned} \tag{14}$$

Adding the inequalities of Equation (14)

$$\begin{aligned} &\|g(w^{t+1}, s^{t+1}, \mu^{t+1})\|_2 + \|\tilde{g}(w^{t+1}, s^{t+1})\|_2 \\ &\leqslant \max\{\sqrt{\alpha} + \sqrt{\tilde{\beta}}, \sqrt{\tilde{\alpha}} + \sqrt{\beta}\} \left(\|g(w^t, s^t, \mu^t)\|_2 + \|\tilde{g}(w^t, s^t)\|_2\right) \end{aligned} \tag{15}$$

To ensure that the above inequality is a contraction, we need to ensure $\sqrt{\alpha} + \sqrt{\tilde{\beta}}, \sqrt{\tilde{\alpha}} + \sqrt{\beta} \in (0, 1)$. These imply (i) $0 < \alpha < 1, 0 < \tilde{\alpha} < 1$, (ii) $\sqrt{\tilde{\beta}} < 1, \sqrt{\beta} < 1$, and (iii) $\sqrt{\alpha} + \sqrt{\tilde{\beta}} < 1, \sqrt{\tilde{\alpha}} + \sqrt{\beta} < 1$. We can solve for γ and η from these inequalities and obtain the sufficient conditions when $\lambda > \tilde{P}$ and $\tilde{\lambda} > P$:

$$\begin{aligned} &\gamma < \frac{1}{\tilde{P}}, \gamma < \frac{2\lambda}{n^2L^2}, \text{ and } \gamma < \frac{\lambda - \tilde{P}}{n^2L^2 - \tilde{P}^2}, \\ &\eta < \frac{1}{P}, \eta < \frac{2\tilde{\lambda}}{m^2\tilde{L}^2}, \text{ and } \eta < \frac{\tilde{\lambda} - P}{m^2\tilde{L}^2 - P^2}. \end{aligned}$$

Notice that from conditions 1 and 2 of Assumption 1, we have that $\lambda < L$ and $\tilde{\lambda} < \tilde{L}$. Thus $n^2L^2 - \tilde{P}^2 > 0$ and $m^2\tilde{L}^2 - P^2 > 0$. For the γ, η chosen as above, we find $W_1 = \sqrt{\alpha} + \sqrt{\tilde{\beta}} < 1$ and $W_2 = \sqrt{\tilde{\alpha}} + \sqrt{\beta} < 1$, and hence $W = \max\{W_1, W_2\} < 1$. Therefore,

$$\begin{aligned} &\|g(w^{t+1}, s^{t+1}, \mu^{t+1})\|_2 + \|\tilde{g}(w^{t+1}, s^{t+1})\|_2 \\ &\leqslant W\left(\|g(w^t, s^t, \mu^t)\|_2 + \|\tilde{g}(w^t, s^t)\|_2\right). \end{aligned}$$

Recursively iterating over this inequality, we get

$$\begin{aligned} &\|g(w^T, s^T, \mu^T)\|_2 + \|\tilde{g}(w^T, s^T)\|_2 \\ &\leqslant W^T\left(\|g(w^0, s^0, \mu^0)\|_2 + \|\tilde{g}(w^0, s^0)\|_2\right). \end{aligned}$$

Defining $E = \|g(w^0, s^0, \mu^0)\|_2 + \|\tilde{g}(w^0, s^0)\|_2$ and

$$T_0(w^0, s^0) = \left(\ln \frac{E}{\epsilon}\right) / \left(\ln \frac{1}{W}\right),$$

we get that for all $T \geqslant T_0(w^0, s^0)$,

$$\begin{aligned} &\|g(w^T, s^T, \mu^T)\|_2 + \|\tilde{g}(w^T, s^T)\|_2 < \epsilon \\ \implies &\|g(w^T, s^T, \mu^T)\|_2 < \epsilon, \text{ and } \|\tilde{g}(w^T, s^T)\|_2 < \epsilon. \end{aligned}$$

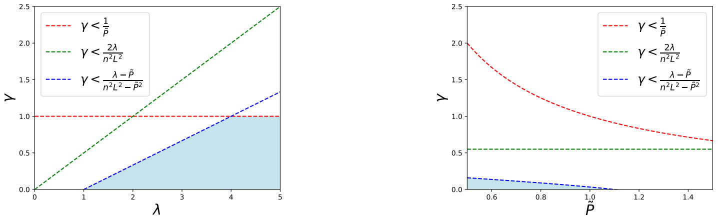
This completes the proof.

A.2 Feasible region of γ for Theorem 1 to hold

Figure 3 shows feasible regions of γ for certain $n, L, \lambda, \tilde{P}$.

A.3 Application of Theorem 1: effective costs are only personal

One special but practical case of the above setup is when the effective cost is only borne by the agent. Mathematically, it is represented as $z_i(s_i, s_{-i}) = c_i(s_i), \forall s_i \in S_i, s_{-i} \in S_{-i}, \forall i \in N$. We assume convex cost functions $c_i, \forall i \in N$, which is a commonly used assumption [19] in the literature. This setup is consistent with the assumptions of Assumption 1, and according to Theorem 1, Algorithm 1 converges to a Nash equilibrium, which we denote as (w^*, s^*). However, it is possible that none of the following things happen: (i) $s^* \neq s^{\max}$, i.e., the agents do not contribute their entire data for the federated learning process, leading to a suboptimal learning, or (ii) $w^* \neq w^{\texttt{OPT}}$ (see Equation (2) for the definition), which is neither the objective of the center nor the agents. The following example shows such an instance.



(a) $\tilde{P} = 1$

(b) $\lambda = 1.1$

Figure 3: Shaded regions show the feasible choices of γ for $n = 2, L = 1$ and $\tilde{P}$ and λ as shown. The dashed lines show the boundary of the regions in the legends. A similar set of choices is true for η.

Example 1 (NE different from socially optimal)**.** *Consider the federated learning setup with two agents that have linear costs and are trying to learn a model with two parameters. The accuracy function is identical and is given by* $a_i(w, s) = 1 - \frac{(1-w_1)^2 + (2-w_2)^2}{s_1 + s_2}, i = 1, 2$. *The cost functions for agents* 1 *and* 2 *are given by* $c_1(s_1) = 0.04 \cdot s_1$ *and* $c_2(s_2) = 0.02 \cdot s_2$ *respectively. The agents' strategy sets are* $S_i = [0, 5], i = 1, 2$. *Notice that, at* $s = s^{\max} = (5, 5)$, *the value of* $w^{\texttt{OPT}}$ *(see Equation* (2)*) is* $(1, 2)$. *It yields an optimal social welfare of* 2. *However, this is not an NE, since the derivatives* $\partial u_i / \partial s_i|_{s_i^{\max}} < 0$ *for both* $i = 1, 2$. *Hence, the best response of each agent is to reduce* s_i. *However, from this point, with the choice of* $\gamma = 0.25$ *and* $\eta = 0.25$, *Algorithm 1 converges to an NE profile of* $w^* = (0.5, 1.5), s^* = (0, 5)$ *that yields a social welfare of* 1.8.

So, our objective in this paper is to allow monetary transfer among the agents so that we get the best of both worlds: (a) We achieve convergence to a Nash equilibrium of $(w^{\texttt{OPT}}, s^{\max})$, where all agents to contribute their entire data and center learns the optimal parameter, and (b) the transfers are *budget balanced*, i.e., the center does not accumulate any money – it is used just to realign the agents' utilities to reach the desired Nash equilibrium. We discuss this mechanism in the following section.

B Appendix to Section 4

B.1 Proof of Lemma 1

Proof. From Equation (8), we get the derivative of the utility function for $s_i \in (0, s_i^{\max})$ (note that this holds only for the interior of S_i, the derivative at the boundaries are zero by definition of function g in Section 3) to be $[g(w, s, \mu)]_i = \frac{\partial}{\partial s_i} u_i(w, s) = \frac{\partial}{\partial s_i} a_i(w, s) - c_i'(s_i) + \beta$, where $c_i'(s_i) = \frac{d}{ds_i} c_i(s_i)$. Using Assumption 2 and choosing in Equation (7) the parameter $\beta > \zeta + \tau$, we have

$$[g(w, s, \mu)]_i = \frac{\partial}{\partial s_i} a_i(w, s) - c_i'(s_i) + \beta \geqslant -\tau - \zeta + \beta > 0.$$

The last inequality holds since $\beta > \zeta + \tau, \forall i \in N$. □

B.2 Proof of Theorem 2

Phase 1. Using 1

$$[g(w^0, s^t, \mu^t)]_i \geqslant -\tau - \zeta + \beta > 0 \tag{16}$$

For the choice of $\beta > \zeta + \tau$, we get $\Delta_i := \beta - \zeta - \tau > 0$. Applying the update rule for s given by Algorithm 2 in phase 1, we get

$$s^{t+1} = s^t + \gamma g(w^0, s^t, \mu^t) \geqslant s^t + \gamma \Delta_i,$$

and applying the inequality repeatedly for t iterations yields

$$s_i^t \geqslant s_i^0 + \gamma(t\Delta_i).$$

Let $l_i = s_i^{\max} - s_i^0$. We have,

$$s_i^t \geqslant s_i^{\max} - l_i + \gamma(t\Delta_i)$$

Substituting $t = \kappa$ where $\kappa \geqslant \dfrac{l_i}{\Delta_i\gamma}$, we obtain

$$\begin{aligned} s_i^\kappa &\geqslant s_i^{\max} - l_i + \gamma\Delta_i(\frac{l_i}{\Delta_i\gamma}) \\ &\geqslant s_i^{\max} - l_i + l_i \\ &\geqslant s_i^{\max} \\ s_i^\kappa &\geqslant s_i^{\max} \end{aligned} \tag{17}$$

Since, $s_i^t \leqslant s_i^{\max}$ for all t and $i \in N$, we conclude that $s_i^\kappa = s_i^{\max}$ for all $i \in N$. Hence, $s^{\max}$ is a fixed point of the update $s^{t+1} = s^t + \gamma g(w^0, s^t, \mu^t)$ at the end of phase 1.

Phase 2 For $t > \kappa$, we can focus on the second phase of the algorithm. From the fixed point property of $s^{\max}$, we can now focus entirely on the function $f(w, s^{\max}) = -\frac{1}{n}\sum_{i\in N} a_i(w, s^{\max})$. From the algorithmic description, the second phase is just a simple gradient descent, run by the center, on $f(w, s^{\max})$. This is easy to see since at every iteration the center gets d_i^{t+1} from all the agents $i \in N$, aggregates them and construct $\tilde{g}(w^t, s^{\max})$, which is the gradient of $f(w, s^{\max})$ computed at w^t. We denote $f(w, s^{\max})$ with $f(w)$ in this part for notational cleanliness.

Note that the gradient descent is run with initialization w^κ (which is same as w^0 as given by first phase of the algorithm). We exploit the strong convexity and smoothness of $f(w, s^{\max})$. Running the second phase for T_0 iterations, using [34], we obtain

$$f(w^{T_0}) - f(w^{\texttt{OPT}}) \leq (1 - \frac{\nu}{M})^{T_0}[f(w^0) - f(w^{\texttt{OPT}})].$$

Hence, for $f(w^{T_0}) - f(w^{\texttt{OPT}}) < \epsilon$, we require

$$T_0 > \left(\ln\frac{f(w^0) - f(w^{\texttt{OPT}})}{\epsilon}\right) \Big/ \left(\ln\frac{1}{\left(1 - \frac{\nu}{M}\right)}\right),$$

which proves the theorem.

B.3 Proof of Corollary 1

Proof. The proof of this follows in the same lines as Theorem 2. Using the choice of β, we first ensure that with κ steps, we obtain $s_i^\kappa = s_i^{\max}$ for all $i \in N$. Now, the framework is same as minimization of a strongly convex and smooth function $f(w)$ with initialization $w^\kappa = w^0$. Using [34], with $\eta = \frac{2}{M+\nu}$, we obtain the iterate convergence, namely $\|w^{\tilde{T}_0} - w^{\texttt{OPT}}\|_2 \leqslant \left(\frac{1-\frac{\nu}{M}}{1+\frac{\nu}{M}}\right)^{\tilde{T}_0}\|w^0 - w^{\texttt{OPT}}\|_2$. Taking log both sides implies the result. □

C Appendix to Section 5

C.1 Algorithm for `FedAvgStrategic`

In the `FedAvgStrategic` algorithm, the agents choose to contribute their data strategically as in `UPBReD`, and stick to this contribution in the entire training phase of `FedAvg`. `FedAvgStrategic` is formally given by Algorithm 3.

C.2 Model Architecture

Algorithm 3: `FedAvgStrategic`

```
Input: Step size $\gamma, \eta$, initialization $w^0$, $s_i^0$ for $i \in N$, number of iterations $T$
Output: $w^T$
Center broadcasts $w^0, s^0$, set $t = 0$                    /* Begin phase 1 */
while $\exists i \in N$ such that $[g(w^0, s^t, \mu^t)]_i > 0$do
    Center broadcasts $s^t$
    for agent $i \in N$ in paralleldo
        Perform local training step and compute $[g(w^0, s^t, \mu^t)]_i$
        $s_i^{t+1} = s_i^t + \gamma[g(w^0, s^t, \mu^t)]_i$
        Send $s_i^{t+1}$ to the center
    $t = t + 1$                                                /* End phase 1 */
$s^* = s^t$
for $t = 0$ to $T$do                                           /* Begin phase 2 */

    Center broadcasts $w^t$
    for agent $i \in N$ in paralleldo
        Compute local gradient: $d_i^{t+1} = \nabla_w \, a_i(w^t, s_i^*, s_{-i}^*)$
        Send $d_i^{t+1}$ to the center
    Center updates: $w^{t+1} = w^t + \frac{\eta}{n} \sum_{i \in N} d_i^{t+1} = w^t + \eta \tilde{g}(w^t, s^*)$
return $w^T$                                                   /* End phase 2 */
```

Table 1: Layer-wise Summary of CNNModel used for FeMNIST

Layer (type:depth-idx)	**Output Shape**	**Param #**
CNNModel	[1, 10]	–
Conv2d: 1-1	[1, 32, 28, 28]	832
MaxPool2d: 1-2	[1, 32, 14, 14]	–
Conv2d: 1-3	[1, 64, 14, 14]	51,264
MaxPool2d: 1-4	[1, 64, 7, 7]	–
Linear: 1-5	[1, 2048]	6,424,576
Linear: 1-6	[1, 10]	20,490
Total Params		**6,497,162**
Trainable Params		**6,497,162**
Non-trainable Params		**0**
Total Mult-Adds (MB)		**17.15**

Table 2: Layer-wise Summary of CNNModel for CIFAR

Layer (type:depth-idx)	**Output Shape**	**Param #**
CNNModel	[1, 10]	–
Conv2d: 1-1	[1, 32, 32, 32]	896
Conv2d: 1-2	[1, 32, 30, 30]	9,248
MaxPool2d: 1-3	[1, 32, 15, 15]	–
Conv2d: 1-4	[1, 64, 15, 15]	18,496
Conv2d: 1-5	[1, 64, 13, 13]	36,928
MaxPool2d: 1-6	[1, 64, 6, 6]	–
Linear: 1-7	[1, 512]	1,180,160
Linear: 1-8	[1, 10]	5,130
Total Params		**1,250,858**
Trainable Params		**1,250,858**
Non-trainable Params		**0**
Total Mult-Adds (MB)		**20.83**
Input Size (MB)		**0.01**
Forward/Backward Pass Size (MB)		**0.70**
Params Size (MB)		**5.00**
Estimated Total Size (MB)		**5.71**

Table 3: Layer-wise Summary of Twitter Model

Layer (type:depth-idx)	**Output Shape**	**Param #**
Twitter	[1, 2]	–
Embedding: 1-1	[1, 25, 300]	120,000,300
LSTM: 1-2	[1, 25, 100]	241,600
Linear: 1-3	[1, 128]	12,928
Linear: 1-4	[1, 2]	258
Total Parameters		**120,255,086**
Trainable Parameters		**120,255,086**
Non-trainable Params		**0**
Total Mult-Adds (MB)		**126.05**
Input Size (MB)		**0.00**
Forward/Backward Pass Size (MB)		**0.08**
Params Size (MB)		**481.02**
Estimated Total Size (MB)		**481.10**