NEURAL INFORMATION PROCESSING SYSTEMS

Mobilizing Personalized Federated Learning in Infrastructure-less and Heterogeneous Environments via Random Walk Stochastic ADMM


Ziba Parsons (University of Michigan-Dearborn)


Fei Dou (University of Georgia)


Houyi Du (University of Michigan-Dearborn)


Zheng Song (University of Michigan-Dearborn)


Jin Lu (University of Georgia)

UNIVERSITY OF MICHIGAN-DEARBORN

UNIVERSITY OF GEORGIA

Motivation

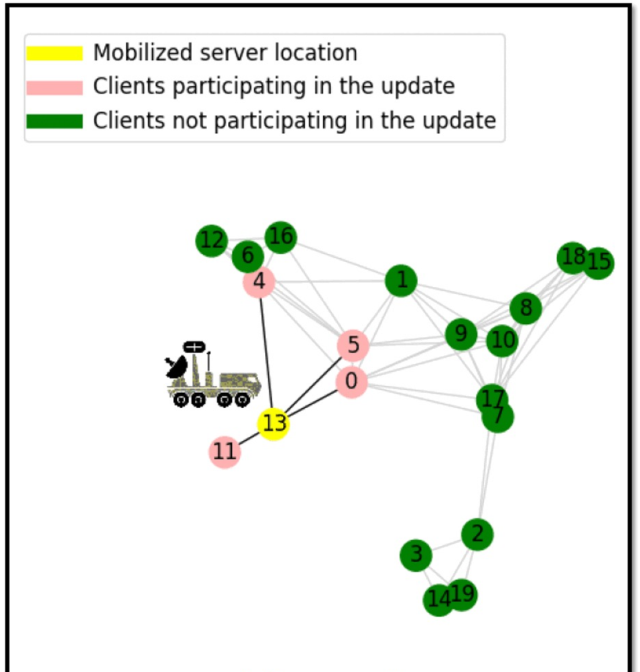
Federated Learning (FL) challenges in real-world applications:

- Limited applicability in environments lacking network infrastructures such as robotics and ad-hoc networks
  - Difficulty in maintaining consistent and reliable connections
  - Change in conditions in dynamic environments with rapidly evolving topologies and ongoing adaptations
  - Limited and constrained communication between central server and clients
- Difference in clients' data distribution and tasks
  - Clients' data distribution is non-IID (non-independent and identically distributed)
  - Clients perform different tasks
  - Lack of generalization of the global model => Model discrepancy

Contribution

❖ To address these FL challenges, we propose a novel and unique FL framework called Random Walk Stochastic Alternating Direction Method of Multipliers (RWSADMM):

- Server moves between clients based on a Random Walk (RW) algorithm
- Presence of data heterogeneity
- A dynamic reachability graph among distributed clients
- A movable vehicle as the central server

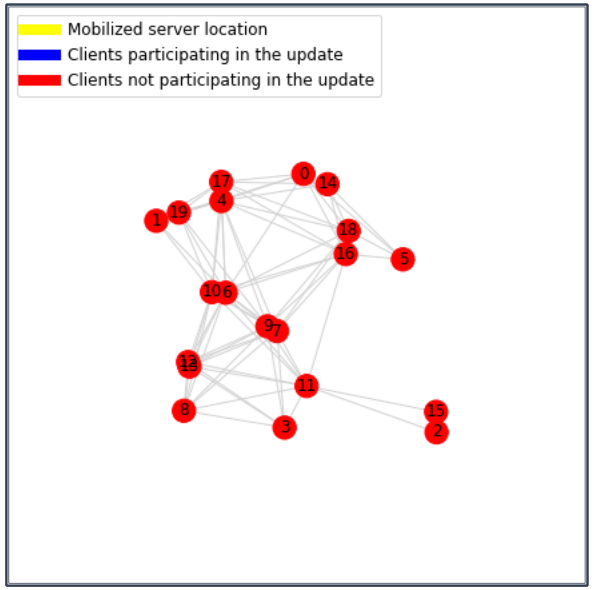


(a) step k

Framework Description

- Clients rely on short-range transmission devices to interact with the movable server
  - Communication is possible only within the communication range
  - Whenever the server arrives in the communication range of Client i, it and its neighbors participate in the computation round
- Server navigates using a non-homogeneous Markov Chain Random Walk method
- Probabilistic approach allows for a more effective server movement and navigation
- Transition matrix $P(k)$ at time k:

$$[P(k)]_{i,j} = \Pr\{i_{k+1} = i | i_k = j\} \in [0, 1]$$



UNIVERSITY OF MICHIGAN-DEARBORN
UNIVERSITY OF GEORGIA

Framework Formulation

- Objective: Minimizing the average loss while ensuring local proximity among clients' local models
- Graph: Dynamic connected graph $G = (V, E)$ with n clients and m edges.
- $V = \{v_1, v_2, \ldots, v_n\}$ is the set of n clients
- E is the set of m edges, which are created if within the communication range.

$$\min_{\mathbf{x}_{1:n} \in R^p} \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}_i)$$

$$\text{s.t.} \quad |\mathbf{x}_i - \mathbf{x}_j| \leq \mathbf{1} \otimes \boldsymbol{\epsilon}_i, \quad \forall i \in \{1, \ldots, n\}$$

- Parameters:
  - x_i: local model parameter stored in client i
  - $f_i(x_i)$: local loss function for client i, potentially non-convex
  - $\boldsymbol{\epsilon}_i$: Non-consensus relaxation between local neighboring clients, replacing model consensus requirement in typical FL frameworks

Framework Formulation

NEURAL INFORMATION PROCESSING SYSTEMS

- By introducing local proximity model y_i stored by the server, the problem is rewritten as:

$$\min_{\mathbf{x}_{1:n}\in R^p} \frac{1}{n} \sum_{n}^{i=1} f_i(x_i)$$

$$\text{s.t.} \quad \left|\mathbf{1}\otimes\mathbf{y}_i - \mathbf{X}_{N(i)}\right| \leq \mathbf{1}\otimes\boldsymbol{\epsilon}_i/2, \quad \forall i \in \{1, \dots, n\}$$

- Parameters:
  - $\boldsymbol{y}_i$: local proximity of $N(i)$
  - $\mathbf{X}_{N(i)}$: Concatenated matrix containing models of client set $N(i)$'s
  - $N(i)$: Vertex set containing client i and its neighbors

Framework Formulation

NEURAL INFORMATION PROCESSING SYSTEMS

- By introducing local proximity model y_i stored by the server, the problem is rewritten as:

$$\min_{\mathbf{x}_{1:n} \in R^p} \frac{1}{n} \sum_{n}^{i=1} f_i(x_i)$$

$$\text{s.t.} \quad \left|\mathbf{1} \otimes \mathbf{y}_i - \mathbf{X}_{N(i)}\right| \leq \mathbf{1} \otimes \boldsymbol{\epsilon}_i / 2, \quad \forall i \in \{1, \ldots, n\}$$

Constrained problem

Augmented Lagrangian Function L_β

$$L_\beta(\mathbf{y}_{1:n}, \mathbf{X}, \mathbf{Z}_{1:n}) = \frac{1}{n}\left[F(\mathbf{X}) + \sum_{i=1}^{n} \langle Z_i, \left|\mathbf{1} \otimes \mathbf{y}_i - \mathbf{X}_{N(i)}\right| - \boldsymbol{\varepsilon}_i \rangle + \frac{\beta}{2} \sum_{i=1}^{n} \left\| \left|\mathbf{1} \otimes \mathbf{y}_i - \mathbf{X}_{N(i)}\right| - \boldsymbol{\varepsilon}_i \right\|_F^2 \right]$$

- Parameters:
  - β: Barrier parameter
  - $\mathbf{Z}_i \in \boldsymbol{R}^{n_i \times p}$: dual variable
  - $\boldsymbol{\varepsilon}_i = \boldsymbol{\epsilon}_i / 2$

UNIVERSITY OF MICHIGAN-DEARBORN
UNIVERSITY OF GEORGIA

NEURAL INFORMATION PROCESSING SYSTEMS

Framework Formulation

- RWSADMM is derived by integrating RW and stochastic inexact approximation techniques into ADMM
  - At iteration k, server approaches client i_k using RW algorithm
  - The clients $N(i_k)$ participate in the federated update
  - The corresponding group of variables, $x_{i_k}, y_{i_k}, z_{i_k}$ are updated in a stochastic way by deriving the solver of each subproblem

$$\mathbf{x}_{i_k} = \arg\min_{\mathbf{x}_{i_k}} L_\beta\left(\mathbf{y}'_{i_k}, \mathbf{x}_{i_k}, \mathbf{z}'_{i_k}\right)$$

$$\mathbf{y}_{i_k} = \arg\min_{\boldsymbol{y}_{i_k}} L_\beta\left(\boldsymbol{y}_{i_k}, \boldsymbol{X}_{N(i_k)}, \mathbf{z}'_{N(i_k)}\right)$$

  - Then the Lagrangian multiplier is updated

$$\mathbf{z}_{i_k} = \mathbf{z}'_{i_k} + \beta\left(\left|\boldsymbol{y}_{i_k} - \mathbf{x}_{i_k}\right| - \boldsymbol{\varepsilon}_{i_k}\right)$$

  - $y'_{\boldsymbol{i}_k}, x_{\boldsymbol{i}_k}, z'_{\boldsymbol{i}_k}$: local parameters stored in client i_k at the $(k-1)th$ iteration

UNIVERSITY OF MICHIGAN-DEARBORN   UNIVERSITY OF GEORGIA

Framework Formulation

❖ X-update:

➢ Driving the solver updating X variable

$$\min_{\mathbf{x}_{i_k}} \left[f_{i_k}(\mathbf{x}_{i_k}) + \langle \mathbf{z}'_{i_k}, |\mathbf{y}'_{i_k} - \mathbf{x}_{i_k}| - \boldsymbol{\varepsilon}_{i_k} \rangle + \frac{\beta}{2} \left\| |\mathbf{y}'_{i_k} - \mathbf{x}_{i_k}| - \boldsymbol{\varepsilon}_{i_k} \right\|_F^2 \right]$$

Substituted by first order stochastic approximation

$$\min_{\mathbf{x}_{i_k}} \left[g_{i_k}(\mathbf{x}'_{i_k}, \xi_{i_k})(\mathbf{x}_{i_k} - \mathbf{x}'_{i_k}) + \langle \mathbf{z}'_{i_k}, |\mathbf{y}'_{i_k} - \mathbf{x}_{i_k}| - \boldsymbol{\varepsilon}_{i_k} \rangle + \frac{\beta}{2} \left\| |\mathbf{y}'_{i_k} - \mathbf{x}_{i_k}| - \boldsymbol{\varepsilon}_{i_k} \right\|_F^2 \right]$$

One of a few samples randomly selected by client i_k

$$\begin{aligned} \mathbf{x}_{i_k} &= \mathbf{y}'_{i_k} + \frac{1}{\beta} \mathbf{z}'_{i_k} \odot \operatorname{sgn}(\mathbf{t}') - \frac{1}{\beta} \operatorname{sgn}(\mathbf{t}') \odot \left(\boldsymbol{\varepsilon}_{i_k} + g_{i_k}(\mathbf{x}'_{i_k}, \xi_{i_k}) \right) \\ &= \mathbf{y}'_{i_k} + \frac{1}{\beta} \operatorname{sgn}(\mathbf{t}') \odot \left(\mathbf{z}'_{i_k} - \boldsymbol{\varepsilon}_{i_k} - g_{i_k}(\mathbf{x}'_{i_k}, \xi_{i_k}) \right) \end{aligned}$$

➢ Signum $\operatorname{sgn}(.)$ function extracts the sign of a vector and $\mathbf{t}'_{i_k} = \mathbf{y}'_{i_k} - \mathbf{x}'_{i_k}$

The stochastic approximation tremendously reduces memory consumption and computational costs in each computation round

Framework Formulation

- Y-update:
  - Driving the solver updating Y variable

$$\min_{\mathbf{y}_{i_k}} \left[\left\langle \mathbf{Z}'_{N(i_k)}, \left|\mathbf{1} \otimes \mathbf{y}_{i_k} - \mathbf{X}_{N(i_k)}\right| - \boldsymbol{\varepsilon}_{i_k} \right\rangle + \frac{\beta}{2} \left\| \left|\mathbf{1} \otimes \mathbf{y}_{i_k} - \mathbf{X}_{N(i_k)}\right| - \mathbf{1} \otimes \boldsymbol{\varepsilon}_{i_k} \right\|_F^2 \right]$$

$$\mathbf{y}_{i_k} = \frac{1}{n_{i_k}} \sum_{j \in N(i_k)} \left[\mathbf{x}_{i_k} - \left(\frac{\mathbf{z}_{i_k}}{\beta} + \boldsymbol{\varepsilon}_{i_k} \right) \odot \operatorname{sgn}(\mathbf{t}_{i_k}) \right]$$

Reducing the communication cost from $O(n)$ to $O(1)$

$$\mathbf{y}_{i_k} = \mathbf{y}'_{i_k} + \frac{1}{n_{i_k}} \left[\mathbf{x}_{i_k} - \left(\frac{\mathbf{z}_{i_k}}{\beta} + \boldsymbol{\varepsilon}_{i_k} \right) \odot \operatorname{sgn}(\mathbf{t}_{i_k}) \right] - \left[\mathbf{x}'_{i_k} - \left(\frac{\mathbf{z}'_{i_k}}{\beta} + \boldsymbol{\varepsilon}_{i_k} \right) \odot \operatorname{sgn}(\mathbf{t}_{i_k}) \right]$$

  - $\mathbf{t}_{i_k} = \mathbf{y}'_{i_k} - \mathbf{x}_{i_k}$

Substituting $\mathbf{y}_{i_k}$ through mathematical induction significantly reduces the communication costs in each computation round

Framework Formulation

❖ Z-update:

- Driving the solver updating Z variable

$$\mathbf{z}_{i_k} = \mathbf{z}'_{i_k} + \kappa\beta\left(\left|\mathbf{1}\otimes\mathbf{y}_{i_k} - \mathbf{X}_{N(i_k)}\right| - \boldsymbol{\varepsilon}_{i_k}\right)$$

- Strictly updated following standard ADMM scheme
- κ coeficient is decayed after each computation round for achieving better convergence

UNIVERSITY OF MICHIGAN-DEARBORN UNIVERSITY OF GEORGIA

NEURAL INFORMATION PROCESSING SYSTEMS

Algorithm

- Effectiveness
  - Convergent
  - Dynamic graph
  - Heterogeneous data distribution

- Efficiency
  - Save memory cost
  - Save communication cost

Algorithm 1 RWSADMM

Initialization:

Initialize Markov transition matrices $\{\mathbf{P}(0), \mathbf{P}(1), \ldots, \}$.

Initialize $\{\mathbf{x}_i^0\}_{i=1}^n = 0$, $\{\mathbf{z}_i^0\}_{i=1}^n = 0$, and

$$\mathbf{y}^1 = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i^0 - \frac{\mathbf{z}_i^0}{\beta}) = 0$$

RWSADMM($\beta, \mathbf{y}_1$)**:**

repeat

  for $k \in 0, 1, 2, \ldots$ **do**

    Client i_k receives $\mathbf{y}'_{i_k}$ and updates **X**, **Z**, and **y** using following equations:

$$\mathbf{x}_{i_k} = \arg\min_{\mathbf{x}_{i_k}} L_\beta(\mathbf{y}'_{i_k}, \mathbf{x}_{i_k}, \mathbf{z}'_{i_k}),$$

$$\mathbf{y}_{i_k} = \arg\min_{\mathbf{y}_{i_k}} L_\beta(\mathbf{y}_{i_k}, \mathbf{X}_{\mathcal{N}(i_k)}, \mathbf{Z}'_{\mathcal{N}(i_k)}),$$

$$\mathbf{z}_{i_k} = \mathbf{z}'_{i_k} + \beta(|\mathbf{y}_{i_k} - \mathbf{x}_{i_k}| - \boldsymbol{\varepsilon}_i),$$

  end for

  $\kappa = 0.99 \times \kappa$

until the termination condition is TRUE.

RETURN $\mathbf{X}^*, \mathbf{y}^*$

Theoretical Guarantees

- To prove the convergence, a Lyapunov function is defined: $L_\beta^k = L_\beta\,(\mathbf{y}^k, \mathbf{X}^k; \mathbf{Z}^k)$
- $(L_\beta^k)_{k\geq 0}$ is non-decreasing and is lower bounded by infimum of f $(inf(f))$

Convergence Theorem: Suppose the following two assumptions hold:

1. The objective function $f_i(x_i)$ is coercive and L-smooth
2. Random Walk forms an irreducible and aperiodic Markov Chain with mixing time $\tau(\delta)$. *(mixing time $\tau(\delta)$ (given δ>0) is the smallest integer s.t. $\left\|\left[P(k)^{\tau(\delta)}\right]_{ij} - \pi_j\right\| \leq \delta\pi^*$).*

For $\beta > 2L^2 + L + 2$, it holds that any limit point $(\mathbf{y}^*, \mathbf{X}^*, \mathbf{Z}^*)$ of the sequence $(\mathbf{y}^k, \mathbf{X}^k, \mathbf{Z}^k)$ generated by RWSADMM satisfies that $(\mathbf{y}^*, \mathbf{X}^*, \mathbf{Z}^*)$ is a stationary point with probability 1, that is,

$$Pr\left(0 \in \frac{1}{n}\sum_{i=1}^{n} \nabla f_i\right) = 1$$

Convergence Rate Theorem: (Sublinear convergence rate) With assumptions of convergence theorem and $\beta > 2L^2 + L + 2$, given local models initialized as $\nabla f_i(\boldsymbol{x}_i^0) = \beta \boldsymbol{x}_i^0 = \mathbf{z}_i^0$, $i \in \{1, \dots, n\}$, there exists a subgradient sequence $\{g^k\} \in \partial\, L_\beta^k$ *satisfying*

$$\min_{k\leq K} E\left\|g^k\right\|^2 \leq \frac{C}{K}\left(L_\beta^0 - inf(f)\right), \forall K \geq \tau(\delta) + 2$$

where C is a constant depending on β, L, n, and $\tau(\delta)$. Hence, a gradient sublinear convergence is proved.

Sublinear convergence rate is comparable with other FL frameworks' convergence rate; while they did not consider a dynamic environment.

In a convex problem, RWSADMM is provable to converge with linear convergence rate.

Communication Complexity: Using the convergence rate theorem, the communication complexity of RWSADMM for nonconvex nonsmooth problem is as follows. To achieve ergodic gradient deviation $E_t \coloneqq min_{k\leq K}\, E\left\|g^k\right\|^2 \leq \omega$, $\forall K \geq \tau(\delta) + 2$, it is sufficient to have

$$\frac{C}{K}\left(L_\beta^0 - inf(f)\right) \leq \omega \overset{(*)}{\rightarrow} O\left(\frac{1}{\omega} \cdot \frac{ln^2 n}{\left(1 - \lambda_2(\boldsymbol{P}(k))\right)^2}\right)$$

- (*) is achieved by taking L_β^0 and $inf(f)$ as constants and independent of n and network structure. $\lambda_2(\boldsymbol{P}(k)) = \max\{|\lambda_i(\boldsymbol{P}(k))| : \lambda_i(\boldsymbol{P}(k)) \neq 1\}$ (λ *as eigenvalue*).
- RWSADMM's communication $O(\omega^{-1})$ for K iterations. Per-FedAvg exhibits a higher communication complexity $O(\omega^{-3/2})$. APFL has the communication complexity of $O(\omega^{-3/4} n^{-3/4})$, n is total number of clients. When n is large, APFL's communication complexity is significantly higher than RWSADMM.

Experiments

- Benchmark Datasets: MNIST, Synthetic, and CIFAR10
- Training models: Strongly convex MLR, non-convex MLP, and non-convex CNN

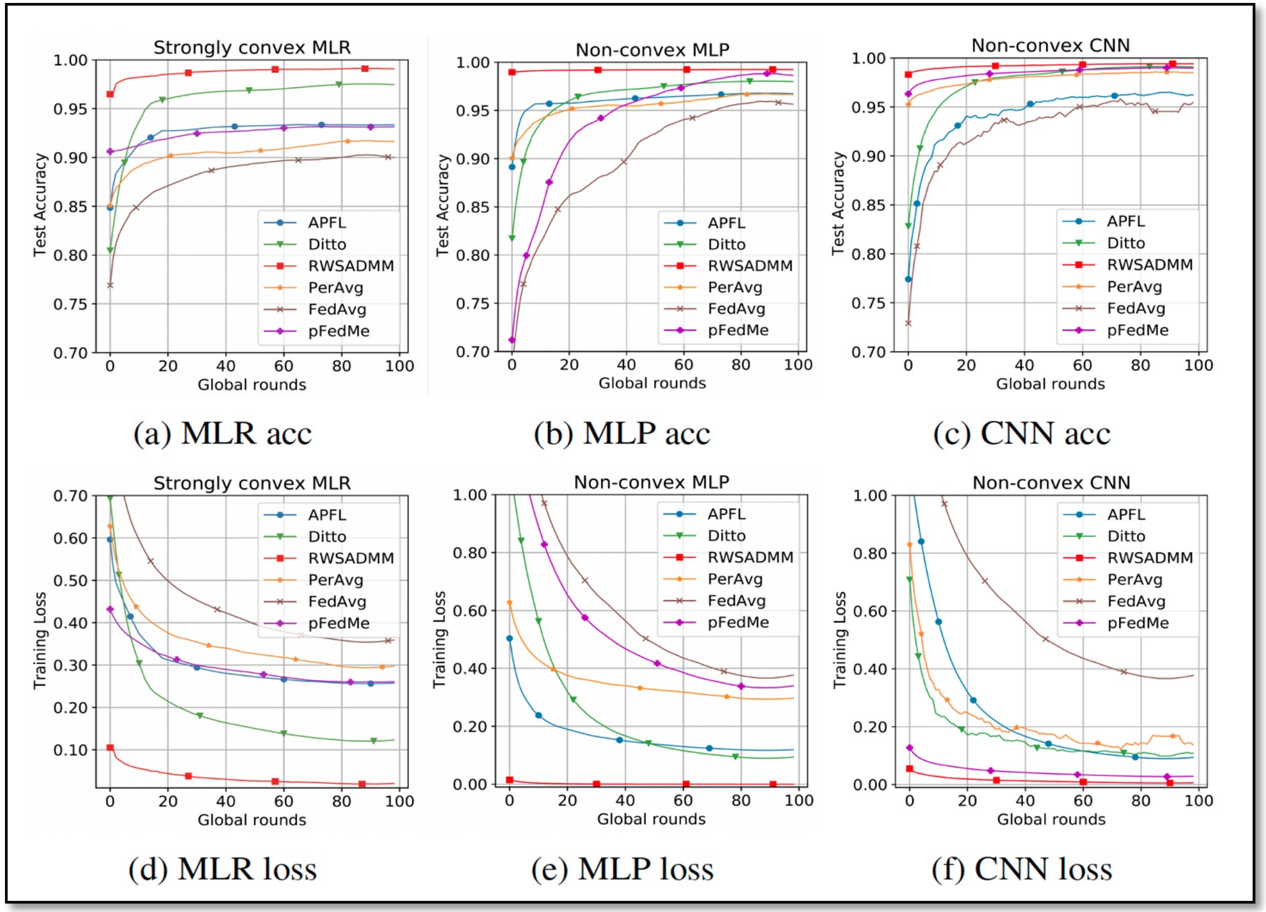
RWSADMM outperforming state-of-the-art FL frameworks with 20 clients for MNIST dataset

Frameworks	MNIST					
	MLR		MLP		CNN	
	acc(%)	t(s)	acc(%)	t(s)	acc(%)	t(s)
FedAvg	93.96 ± 0.02	384	98.79 ± 0.03	464	97.83 ± 0.15	7965
PerAvg	94.37 ± 0.04	472	98.90 ± 0.02	608	98.97 ± 0.08	7296
pFedMe	95.62 ± 0.04	1344	$\mathbf{99.46 \pm 0.01}$	2096	99.05 ± 0.06	16623
Ditto	97.37 ± 0.02	828	97.79 ± 0.03	1268	99.20 ± 0.11	9820
APFL	92.64 ± 0.03	913	97.74 ± 0.02	1598	98.58 ± 0.03	17800
RWSADMM (our method)	$\mathbf{98.63} \pm \mathbf{0.01}$	500	$\mathbf{99.29} \pm \mathbf{0.02}$	884	$\mathbf{99.52} \pm \mathbf{0.04}$	11570

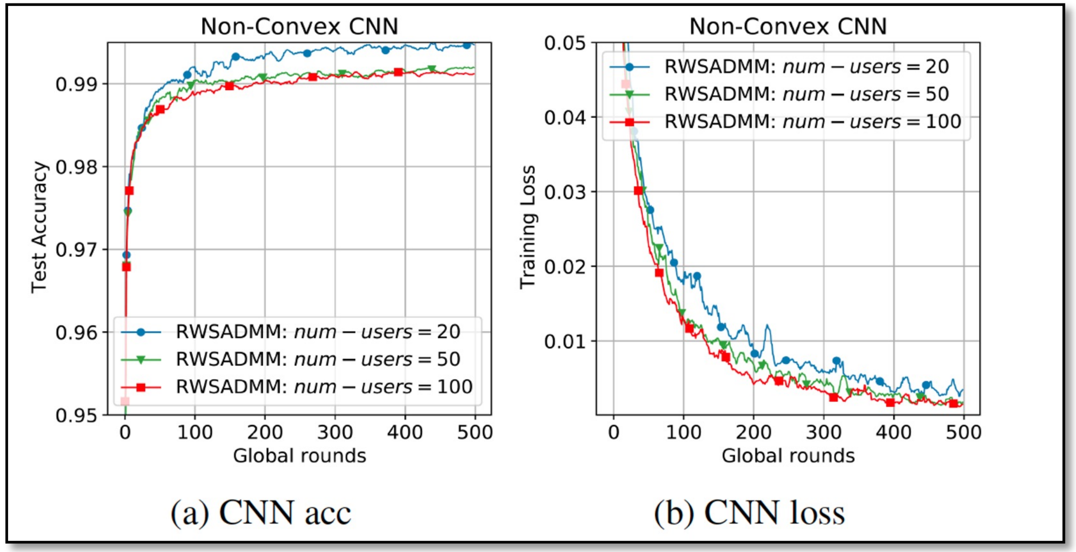
Experiments

NEURAL INFORMATION PROCESSING SYSTEMS

RWSADMM's (red curve) convergence performance with 20 clients for MNIST dataset



Scalability performance of RWSADMM for different number of clients



Conclusion

- Proposed a novel mobile server FL framework called RWSADMM:
  - Provably convergent with sublinear convergence rate for non-convex settings
  - Reduced memory and computation costs, due to stochasticity
  - Outperforming state-of-the-art FL frameworks relative to
    - Provably lower communication complexity
    - Higher accuracy

- In addition, successfully resolved the challenge of implementing FL in an unreliable network environment by:
  - Reliance on short-range communication of ad-hoc networks with a moving server
  - Implementing a dynamic environment and network topology