Distributed Learning in Wireless Networks: Recent Progress and Future Challenges

Anya, Michael, Tiancheng, Jinrui

M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Distributed learning in wireless networks: recent progress and future challenges," IEEE J. on Selected Areas in Communication, vol. 39, no. 12, Dec. 2021.

Table of Contents

Motivation of Distributed Learning - Introduction

Paradigm shift is driven by two trends in the evolution of computing:

- IoT networks provide platforms for executing large-scale tasks and generating large amounts of valuable data.

- The shift enables the deployment of ML algorithms in the proximity of edge devices to distill their collected data into intelligence.

Challenges of Deploying Distributed Learning - Introduction

- Find methods for distributed learning without raw-data sharing
- To perform training and inference of an ML model over wireless links
- Requires many rounds of exchanging between servers and devices
- Requires efficient ways to perform distributed computation
- Requires new distributed optimization frameworks to be efficient over wireless networks

Potential Techniques for Deploying Distributed Learning - Introduction

To accelerate the training of ML models using distributed data requires new algorithms and techniques for integrated communication and learning.

- Compression and sparsification
- Radio resource management
- Over-the-air computation
- Development of novel training methods

Table of Contents

Federated Learning (FL)

Using **local parameters** rather than **raw data** in learning process

- Decentralized/Collaboration
- Data privacy
- High learning accuracy
- Distributed Iterative Learning
- Shared model improvement
- ...

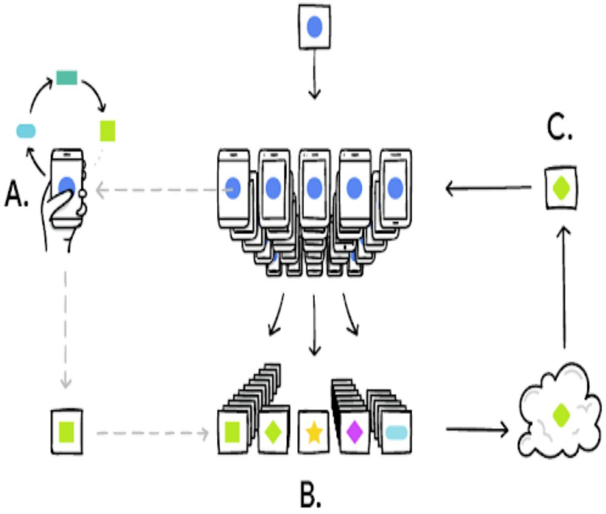


Figure 1: Federated Learning Illustration (image not from the paper) [1]

Preliminaries of the Federated Learning

Components of the Federated Learning

- Parameter Server (PS)
- Edge Devices (U_i)
- Datasets (K_i)
- Input Vector (x_i) and output vector (y_i)

IID Federated Learning Model

Common Federated Learning algorithm: (Ex. FedAvg)

1. PS broadcast parameters
2. Edge devices do local SGD
3. PS update global model
4. Repeat until iterations limit or model convergence

Non-IID Federated Learning Model (Personalized ML model)

Federated Multi-task Learning (FMTL):

- Use learning task relationships (**Ω**) to optimize separate tasks ($\mathbf{m}_i$).
- Optimize **Ω** with the updated $\mathbf{M}(m_1 \ldots m_n)$.

MAML-Based Federated Learning:

- Gradient descent in separate local devices ($\mathbf{U}_i$)
- Find common ML model for all devices
- Use $\mathbf{K}_i$ to update personalized ML models

Performance Metric of Federated Learning over Wireless Network

- Training Loss
- Convergence time
  - Parameter transmission delay (T_T)
  - Local Training time (T_C)
  - Learning Steps (N_T)
- Energy consumption
  - Parameter transmission consumption (E_T)
  - Local Training consumption (E_C)
  - Learning Steps (N_T)
- Reliability

Communication Factor	Training Loss	T_T and E_T	T_C and E_C	N_T	Reliability
Spectrum Resource	✔	✔		✔	✔
Computation capacity	✔		✔	✔	
Transmit Power	✔	✔		✔	✔
Wireless channel	✔	✔		✔	✔
# of devices in FL	✔	✔		✔	✔
Size of parameters trained	✔		✔	✔	✔
Size of parameters transmitted		✔			

Research Directions - Compression and Sparsification

- Training models are large, can’t transmit millions of parameters
    - Reduce number of elements by setting some to 0 via sparsification
    - Top-K sparsification can 2000x reduce load with minimal accuracy loss
    - Quantization adjusts weights so they’re less than 32 bits
    - Areas:
        - Sign-based quantization together with majority voting
        - Trade-off between the number of bits needed to encode compressed vectors and the compression error
        - FL algorithm to manage the trade-offs between power consumption, communication bit-rate and convergence rate

Research Directions - Wireless Resource Management

- Need to optimize resource allocation to efficiently complete the FL training process
    - Hard to quantify how each single model update affects the entire training process
    - Parameter server (PS) has no info on device datasets, only gradient vectors, so can’t use data stats to decide how resource allocation will affect FL convergence
    - Areas:
        - Trade-off between the local ML model updates and global ML model aggregation
        - Optimized device scheduling and resource allocation policies to maximize the model accuracy within a given total training time budget

Research Directions - FL Training Method Design

- Can adjust the learning parameters to enable efficient FL implementation
  - Wireless device energy and computation is limited, so size of ML model parameters that can be trained and transmitted by a device needs to be small and time duration to train is short
  - Areas:
    - Hierarchical FL with device clusters, local learning carried out by devices within each cluster with help of small base station (BS) or cluster head, while a global model is trained at the macro BS
    - Decentralized averaging methods to update the local ML model of each device - each device only needs to transmit its local ML parameters to its neighboring devices and then averages the global ML model

Open Problems of Deploying FL Over Wireless Networks

- Convergence Analysis - need to analyze how wireless factors affect the convergence of realistic FL with local ML models and loss function, existing models make unrealistic assumptions about FL loss function
- Wireless resource management - current research doesn’t account for mobility patterns of devices; adopting suitable frequency bands
- Compression and sparsification - Need heterogeneous compression schemes that consider link characteristics of each device will be different; need to design new schemes that consider data leakage
- FL Training Method Design - need to enable the devices to form an optimal network topology that maximizes many FL performance trade-offs; designing asynchronous training methods while considering the network topology optimization

Industry Interest

- Centralized based algorithms have high latency, can’t work for things like 5G which are near-real time while also satisfying privacy needs
    - Makes keeping data on edge devices (ie, smart phones) attractive
- In 2017, Google made a global model had been trained and deployed on Android devices to suggest search queries based on typing context from Android Gboard, training update occurred over WiFi
- Major potential interest in telecom industry - one paper citing significant reduction in network utilization, due to the sharp drop in the amount of data that needs to be shared

Table of Contents

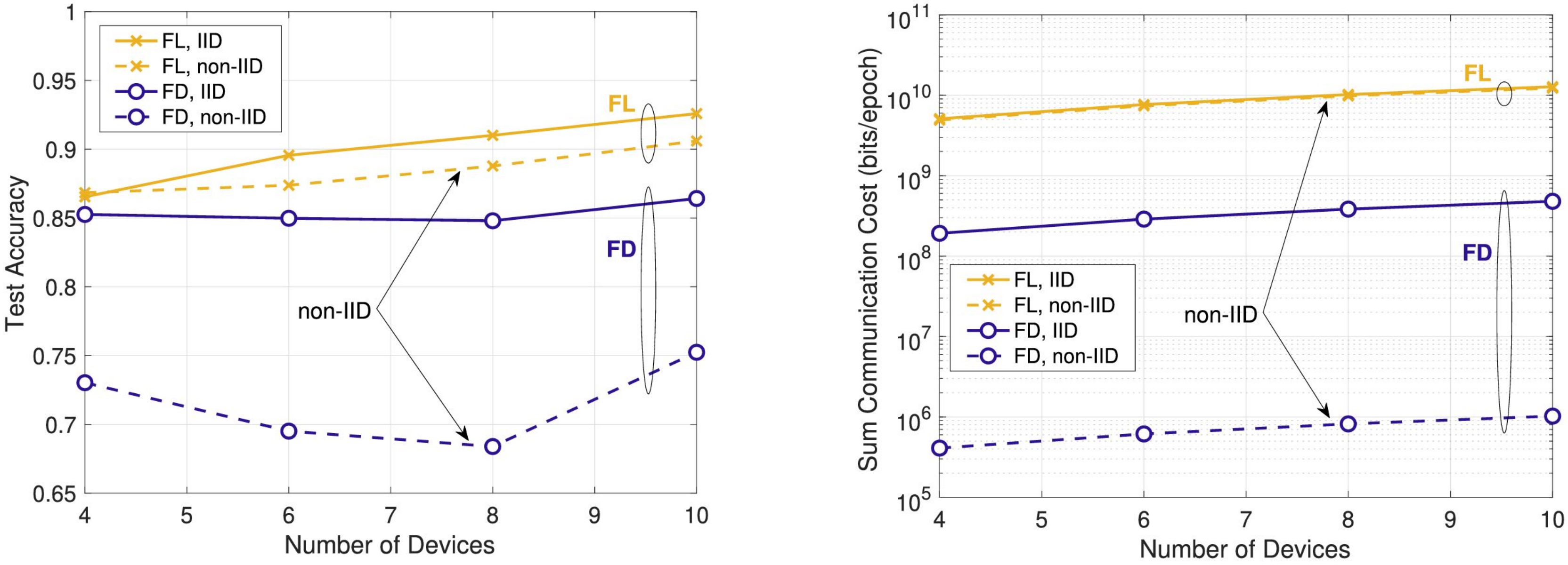
Federated Distillation - Preliminaries

- FD is introduced to avoid exchanging the entire DNN model parameters.

- Knowledge distillation (KD) is the process of transferring knowledge from a large model to a smaller one. While large models have higher knowledge capacity than small models.

Federated Distillation - Representative Result

- Comparison between FD and FL in terms of test accuracy and sum communication cost of all devices per epoch, under an IID or non-IID MNIST data.



(a) Test accuracy.

(b) Sum communication cost.

Table of Contents

Reinforcement Learning - Single Agent RL

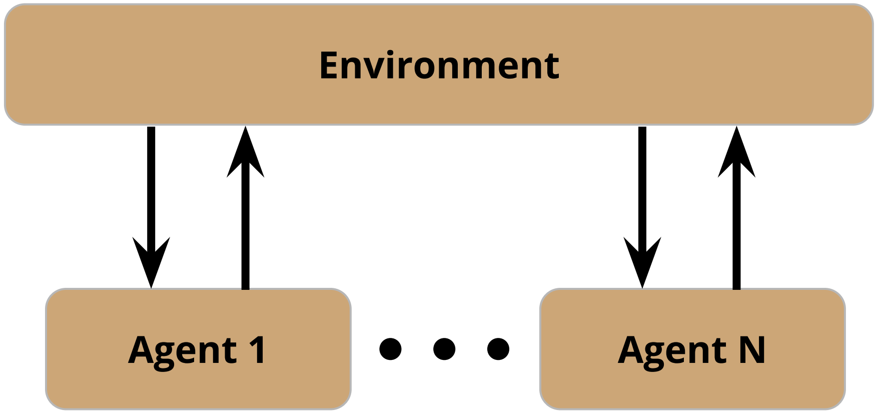
- Several implementations of reinforcement learning (RL)
- The most basic: Single Agent RL
- Teach through a Markov decision process
  - Main components: state, action, reward
- Goal is to maximize expected discounted reward

Reinforcement Learning - Single Agent RL

- Strengths of RL
  - non-convex problems
  - time dependent optimization problems
- Weaknesses of Single-Agent RL
  - As the # of devices increases, so does complexity
  - Complicated state space
  - Overhead communicating to every device

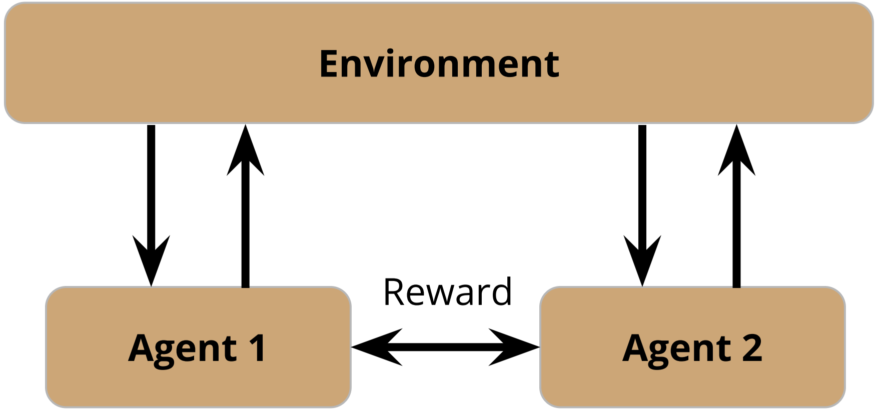
Reinforcement Learning - Independent Multi-Agent RL (MARL)

- A simple solution: make each device its own independent agent
- Each device maximizes without considering others
- Useful for base stations that can't communicate together
- Drawbacks:
  - Not guaranteed to converge
  - Cannot maximize the sum expected reward of all agents



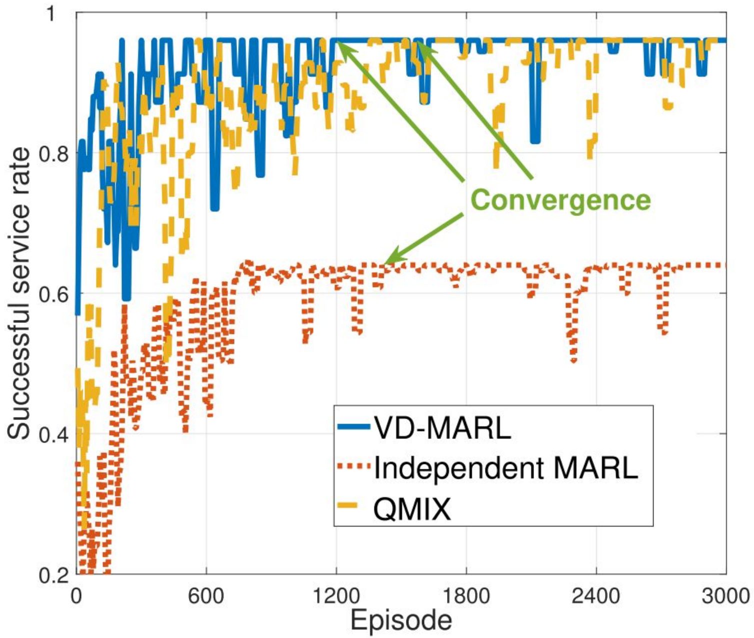
Reinforcement Learning - Collaborative Multi-Agent RL (MARL)

- Can overcome this with Collaborative MARL
- Agents share any combination of parameters
  - Reward, RL model parameters, action, state
  - Reward is the most important one
- Trade off between complexity and performance
  - Dependant on how much each agents shares



Reinforcement Learning - UAV Trajectory Design

- Clusters of users with unpredictable uplink access commands
- Can’t use branch and bound because of random variation
- Authors proposed VD-MARL, which shares rewards
  - Low overhead
  - 54% better than independent
- QMIX is significantly more complex
  - 31% slower convergence



Reinforcement Learning - Areas of Research

- Can predict convergence on single agent RL algorithms
  - Game theory
- Harder to predict more complex algorithms like QMIX
  - Will it converge?
  - How does # of agents affect convergence?
- Efficient wireless communication
  - optimization of resource block allocation
  - reliable and efficient transmission

Table of Contents

Conclusions

This paper:

- Introduced four distributed learning frameworks and the motivation behind: FL, FD, distributed inference, and MARL.
- Mainly focus on the FL framework for distributed learning.
- Provided detailed overview of federated averaging, federated multi-task learning, and model agnostic meta learning based FL and summarize their drawbacks and usage.
- Explored the possibility of performing joint learning and communications when FL is deployed in wireless networks.

Questions?

References

[1] https://ai.googleblog.com/2017/04/federated-learning-collaborative.html