Optimizing Federated Learning on Non-IID Data with Reinforcement Learning

Letícia Mayumi A. Tateishi (RA 201454)

Institute of Computing - Unicamp

Agenda

- Motivation
- Challenges
- Goals
- Reinforcement Learning
- The FAVOR framework
  - Deep Reinforcement Learning for Client Selection
  - Deep-Q network
  - Algorithm
- Experiments and results
- Conclusions

Motivation

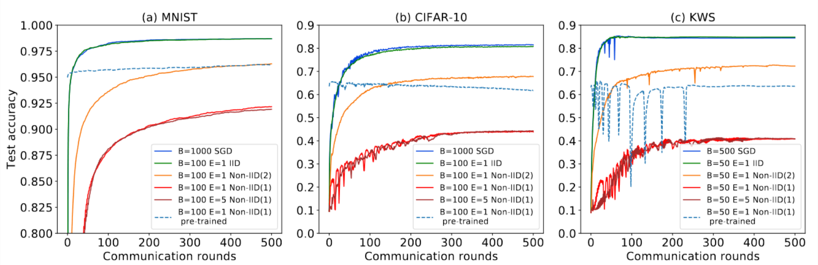


Figure 1: Test accuracy vs communication rounds for FedAvg and SGD with IID and non-IID data. Used datasets are MNIST (b) CIFAR-10 and (c) KWS

Challenges

- Limited connectivity of wireless networks;
- Unstable availability of mobile devices;
- Non-IID distributions of local datasets;
- Counterbalancing the bias introduced by non-IID data.

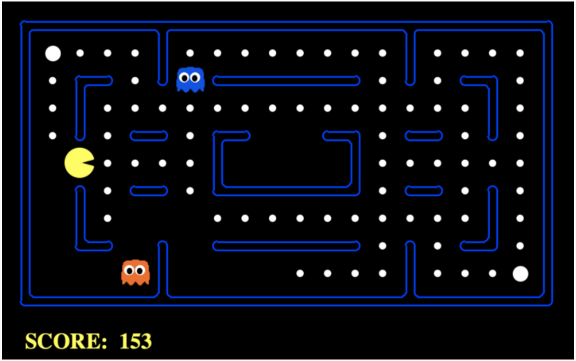
Goals

- Minimize the number of communication rounds needed for convergence;
- Guarantee user privacy;
- Counterbalance the bias introduced by non-IID data;
- Improve FL performance by intelligently choosing the client devices to participate in each round.

Reinforcement Learning

Definition

Reinforcement Learning is the learning process of an agent that acts in corresponding to the environment to maximize its rewards.

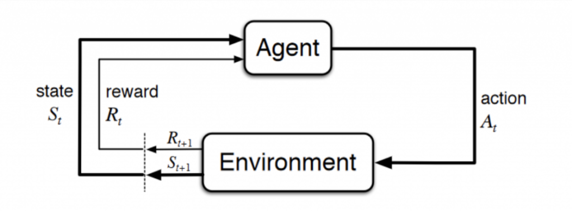


Reinforcement Learning

At each time step t:

1. Agent observes state s_t and performs action a_t
2. The environment transits to state s_{t+1}
3. The agent receives reward r_t

The goal is to maximize the expectation of the cumulative return.

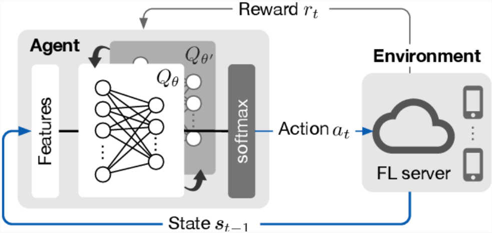


Reinforcement Learning

At each time step t:

1. Agent observes state s_t and performs action a_t
2. The environment transits to state s_{t+1}
3. The agent receives reward r_t

The goal is to maximize the expectation of the cumulative return.



The cumulative return we want to maximize is also called Q-value (Quality Value):

$$Q^*(s_t, a) = r_t(s_t, a) + \gamma \max_a Q(s_{t+1}, a)$$

- r_t: reward at step t
- γ: discount factor for future rewards ($0 \leq \gamma < 1$)

Reinforcement Learning

$$Q^*(s_t, a) = r_t(s_t, a) + \gamma \max_a Q(s_{t+1}, a)$$

Typically, a deep neural network is used to represent the function approximator. The RL learning problem becomes minimizing the MSE loss between the target and the approximator:

$$l_t(\theta_t) = (Q^*(s_t, a) - Q(s_t, a, \theta_t))^2$$

$$l_t(\theta_t) = (r_t(s_t, a) + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a, \theta_t))^2$$

FAVOR: DRL for Client Selection

FL as a Markov Decision Process

The per-round FL process can be modeled as a Markov Decision Process, where the state s_t represents the global model weights and the model weights of each client device in each round.

The goal is to use the local model weights to select a subset of devices to perform local training.

FAVOR: DRL for Client Selection

At each time step t:

1. The DRL agent takes an action a_t to select a subset of devices;
2. The selected devices perform local training and update the global model (state s_{t+1});
3. A reward r_t is observed, which is a function of the test accuracy.

The objective is to train the DRL agent to converge to the target accuracy for federated learning as quickly as possible.

Deep-Q Network: the state

At each time step t, the state s_t is defined as:

$$s_t = (w_t, w_t^{(1)}, ..., w_t^{(N)})$$

where w_t represents the gloabl model weights and $w_t^{(1)}, ..., w_t^{(N)}$ are the weights from the N devices.

The agent must keep a list of weights $\{w^{(k)} | k \in [N]\}$.

Deep-Q Network: the state

The agent must keep a list of weights $\{w^{(k)} | k \in [N]\}$.

Problem: Keeping the weights of many neural networks leads to a large state space

Solution: Apply principal component analysis (PCA) to model weights and use the compressed model weights to represent states

Deep-Q Network: the state

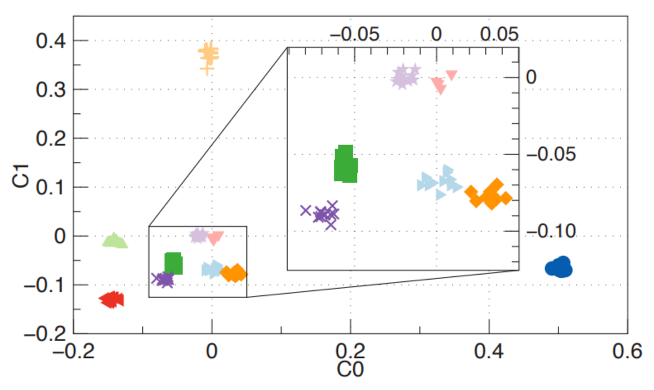


Figure 2: Model weight vectors for imbalanced data samples (80% vs. 20%). Different shapes (or colors) indicate local models trained on devices with different dominant labels.

Deep-Q Network: the reward

At round t, the reward is:

$$r_t = \xi^{(\omega_t - \Omega)} - 1$$

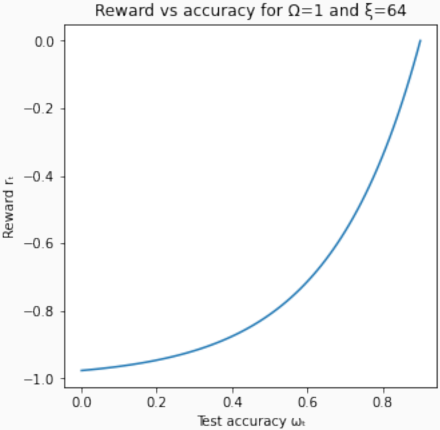
$\omega_t :=$ test accuracy

$\Omega :=$ target accuracy

$\xi :=$ positive constant

Reward

$$R = \sum_{t=1}^{T} \gamma^{t-1} r_t$$



FAVOR

Algorithm 1 Server

$s \leftarrow [w_{init}, w_{init}, ..., w_{init}]$
for $k = 1$ to N **do**
    $s[k] \leftarrow train(k, s[0])$
end for
for $t = 1$ to T **do**
    $Q \leftarrow Q\_Value(s)$
    for k in TopK(Q) **do**
        $s[k] \leftarrow train(k, s[0])$
    end for
    $s[0] \leftarrow FedAvg(s)$
end for

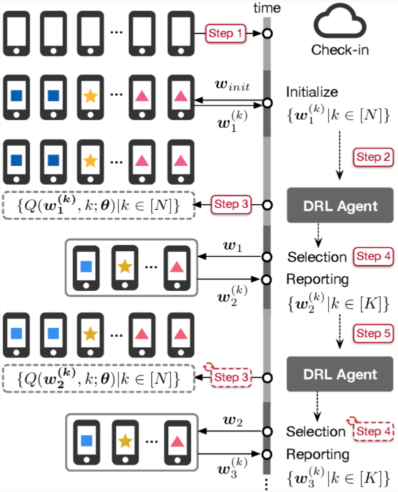


Figure 3: Federated Learning workflow with FAVOR.

Evaluation

- Official implementation in Pytorch
- Open-source project
- Trained CNN models on 3 benchmark datasets: MNIST, FashionMNIST and CIFAR-10
- Classification models consist of two 5x5 convolution layers, changing the output size according to the dataset
- The DDQN model in the DRL agent consists of two two-layer MLP networks, with 512 hidden states.

Results

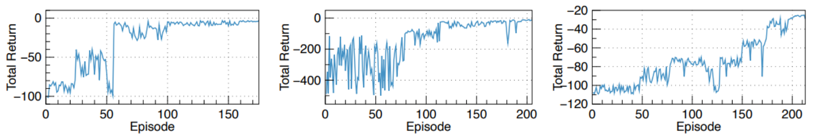


Figure 4: Total return per episode on three different datasets: MNIST, FashionMNIST and CIFAR-10, respectively.

Results

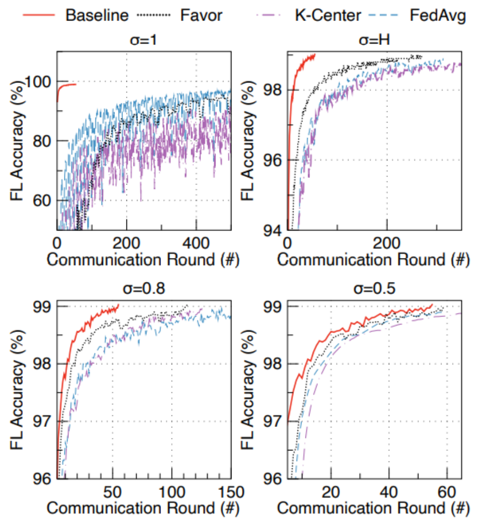


Figure 5: Accuracy v.s. communication rounds on different levels of non-IID MNIST data.

Results

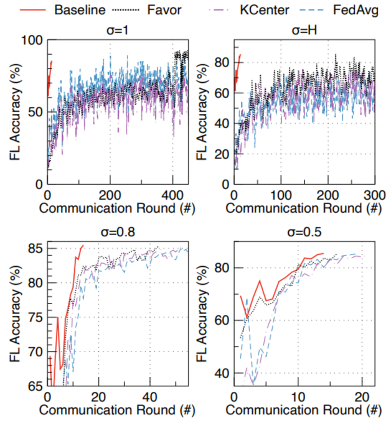


Figure 6: Accuracy vs communication rounds on non-IID FashionMNIST data.

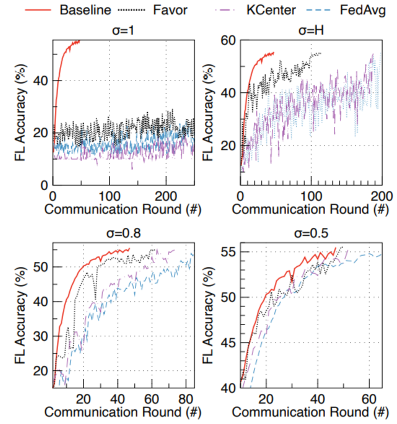


Figure 7: Accuracy vs communication rounds on non-IID CIFAR-10 data.

Results

	σ	MNIST	FashionMNIST	CIFAR-10
FedAvg	0 (IID)	55	14	47
FedAvg	1.0	1517	1811	1714
K-Center	1.0	1684	2132	1871
FAVOR	1.0	**1232**	**1497**	**1383**
FedAvg	H	313	1340	198
K-Center	H	421	1593	188
FAVOR	H	**272**	**1134**	**114**
FedAvg	0.8	221	52	87
K-Center	0.8	126	62	74
FAVOR	0.8	**113**	**43**	**61**
FedAvg	0.5	**59**	19	67
K-Center	0.5	67	21	52
FAVOR	0.5	**59**	**16**	**50**

Conclusions

- Non-IID data exacerbates the divergence of model weights on devices, and increases the number of communication rounds
- Proposed to actively select a specific subset of devices to participate in training at each round
- Designed a DRL agent that applies DDQN to select the best subset of devices at each round
- Reduced the number of communication rounds by up to 49% on the MNIST dataset, up to 23% on FashionMNIST, and up to 42% on CIFAR-10

References

[1] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li.
Optimizing federated learning on non-iid data with reinforcement learning.
In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 1698–1707, 2020.

[2] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra.
Federated learning with non-iid data.
CoRR, abs/1806.00582, 2018.

[3] R. J. Williams.
Simple statistical gradient-following algorithms for connectionist reinforcement learning.
Machine Learning, 8:229–256, 1992.

[4] Hado van Hasselt, Arthur Guez, and David Silver.
Deep reinforcement learning with double q-learning.
CoRR, abs/1509.06461, 2015.