INFOCOM’20

Optimizing Federated Learning on Non-IID Data with Reinforcement Learning

Hao Wang*, Zakhary Kaplan*, Di Niu^, Baochun Li*

*University of Toronto, ^University of Alberta
Machine Learning
Federated Learning
Sure. Umami burger?

Yeah. Know the address?

738 E. 3rd St.
Federated Averaging Algorithm (FedAvg)
Random selection

Local model

Local data
Random selection

Local model

Local data
Thank you for the feedback

Local model

Local data
ML algorithms assume the training data is independent and identically distributed (IID)
Federated Learning reuses the existing ML algorithms but on **non-IID data**
Non-IID data introduces bias into the training and leads to a slow convergence and training failures
MNIST

http://yann.lecun.com/exdb/mnist/
<table>
<thead>
<tr>
<th>Communication Round (#)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FedAvg-IID</td>
</tr>
<tr>
<td>1</td>
<td>91</td>
</tr>
<tr>
<td>10</td>
<td>93</td>
</tr>
<tr>
<td>19</td>
<td>95</td>
</tr>
<tr>
<td>28</td>
<td>97</td>
</tr>
<tr>
<td>37</td>
<td>97</td>
</tr>
<tr>
<td>46</td>
<td>97</td>
</tr>
<tr>
<td>55</td>
<td>97</td>
</tr>
<tr>
<td>64</td>
<td>97</td>
</tr>
<tr>
<td>73</td>
<td>97</td>
</tr>
<tr>
<td>82</td>
<td>97</td>
</tr>
<tr>
<td>91</td>
<td>97</td>
</tr>
<tr>
<td>100</td>
<td>97</td>
</tr>
<tr>
<td>109</td>
<td>97</td>
</tr>
<tr>
<td>118</td>
<td>97</td>
</tr>
<tr>
<td>127</td>
<td>97</td>
</tr>
<tr>
<td>136</td>
<td>97</td>
</tr>
<tr>
<td>145</td>
<td>97</td>
</tr>
<tr>
<td>154</td>
<td>97</td>
</tr>
</tbody>
</table>

The graph shows the comparison between FedAvg-IID and FedAvg-non-IID in terms of accuracy over communication rounds.
Build IID training data?

No, we don’t have any access to the data on your phone.
4 Proposed Solution

In this section, we propose a data-sharing strategy to improve FedAvg with non-IID data by creating a small subset of data which is globally shared between all the edge devices. Experiments show that test accuracy can be increased by ~30% on CIFAR-10 dataset with only 5% globally shared data.

4.1 Motivation

As shown in Figure 5, the test accuracy falls sharply with respect to EMD beyond a certain threshold. Thus, for highly skewed non-IID data, we can significantly increase the test accuracy by slightly reducing EMD. As we have no control on the clients' data, we can distribute a small subset of global data containing a uniform distribution over classes from the cloud to the clients. This fits in with the initialization stage of a typical federated learning setting. In addition, instead of distributing a model with random weights, a warm-up model can be trained on the globally shared data and distributed to the clients. Because the globally shared data can reduce EMD for the clients, the test accuracy is expected to improve.

Optimizing Federated Learning on Non-IID Data with Reinforcement Learning
Build IID training data? **No**

**Peeking** into the data distribution on each device without violating data privacy

**Probing** the bias of non-IID data
Carefully select devices to balance the bias introduced by non-IID data.
Probing the data distribution
Initial model

A two-layer CNN model with 431,080 parameters

Local model

100 devices, each has 600 samples

Non-IID data

80% data has the same label, e.g., “6”
We apply Principle Component Analysis (PCA) to reduce dimensionality

431,080-dimension model weight  →  2-dimension space
An implicit connection between model weights and data distribution
Probing the data distribution

Selecting devices for federated learning
K-Center Clustering
Random Selection from Groups
Accuracy (%) vs Communication Round (#)

- **FedAvg-IID**
- **FedAvg-non-IID**
- **K-Center-non-IID**
Probing the data distribution

Selecting devices for federated learning

How to select devices to speed up training?
It is difficult to select the appropriate subset of devices

- Model weights —> device selection choice
- A dynamic and undeterministic problem

Reinforcement Learning (RL)
Learn to maximize $\sum(\text{reward})$
States

Global weights

Local model weights

100-dimension vector
Select K devices from a pool of N devices — a huge action space

Selecting 10 devices from a pool of 100 devices leads to

1.7310309e+13 possible actions
Modify the RL training algorithm
Only one device is selected during the RL training

Now the action space is \( \{1, 2, \ldots, N\} \), instead of selecting K devices from N devices.
Evaluating Each Device

Select the top K
Rewards

\[ r_t = \Xi^{(\omega_t - \Omega)} - 1 \]

\[ 0 \leq \omega_t \leq \Omega \leq 1 \]

\[ r_t \in (-1,0] \]

| \[ \Xi \] | Positive constant |
| \[ \omega_t \] | Training Accuracy |
| \[ \Omega \] | Target accuracy |
| \[ t \] | Communication round # |

Accuracy increase: \( \omega_t \uparrow \longrightarrow r_t \uparrow \)

More communication rounds: \( t \uparrow \longrightarrow \text{sum}(r_t) \downarrow \)
Training the DRL Agent

Look for a function that points out the actions leading to the maximum cumulative return under a particular state.

Max \[ R = \sum_{t=1}^{T} \gamma^{t-1} r_t = \sum_{t=1}^{T} \gamma^{t-1}(\varepsilon(\omega_t - \Omega) - 1) \]

discount factor \( \gamma \in (0,1) \)
DDQN

Environment

Reward $r_t$

Agent

State $s_{t-1}$

Action $a_t$

Features

softmax

FL server
Training the DRL agent
Evaluating Our Solution

**Benchmark:** MNIST, FashionMNIST, CIFAR-10

**Non-IID level:** 1, half-and-half, 80%, 50%

**Half-and-half**

```
3 3 3 3 3 3 7 7 7 7
```

**80%**

```
6 6 6 6 6 6 6 6 3 7
```
Non-IID level

1

Communication Rounds

MNIST

FashionMNIST

CIFAR-10

FedAvg

K-Center

Favor
Non-IID level
half & half

Communication Rounds

- FedAvg
- K-Center
- Favor

Datasets:
- MNIST
- FashionMNIST
- CIFAR-10
Communication Rounds

Non-IID level
80%

MNIST
FashionMNIST
CIFAR-10

FedAvg
K-Center
Favor
Non-IID level
50%

Communication Rounds

FedAvg  K-Center  Favor

MNIST  FashionMNIST  CIFAR-10

49
Indirect data distribution probing
DRL-based device selection
Communication rounds can be reduced by up to

- 49% on the MNIST
- 23% on FashionMNIST
- 42% on CIFAR-10