

Hierarchical Federated Learning with Multi-Timescale Gradient Correction

Our experiments are based on the implementation of [Federated Learning Based on Dynamic Regularization](#).

Requirements

Please install the required packages. The code is compiled with Python 3.7 dependencies in a virtual environment via:

```
pip install -r requirements.txt
```

Instructions

How to Run

For different configurations, use the following commands:

1. **MTGC:**

```
python run_MTGC.py \  
  --rule 'noniid' \  
  --rule_arg 0.1 \  
  --com_amount 100 \  
  --epoch 2 \  
  --E 30
```

2. **Group Correction:**

```
python run_MTGC_Y.py \  
  --rule 'noniid' \  
  --rule_arg 0.1 \  
  --com_amount 100 \  
  --epoch 2 \  
  --E 30
```

3. **Local Correction:**

```
python run_MTGC_Z.py \  
  --rule 'noniid' \  
  --rule_arg 0.1 \  
  --com_amount 100 \  
  --epoch 2 \  
  --E 30
```

4. FedDyn:

```
python run_FedDyn.py \  
  --rule 'noniid' \  
  --rule_arg 0.1 \  
  --com_amount 100 \  
  --epoch 2 \  
  --E 30
```

5. FedProx:

```
python run_FedProx.py \  
  --rule 'noniid' \  
  --rule_arg 0.1 \  
  --com_amount 100 \  
  --epoch 2 \  
  --E 30
```

6. HFedAvg:

```
python run_HFL.py \  
  --rule 'noniid' \  
  --rule_arg 0.1 \  
  --com_amount 100 \  
  --epoch 2 \  
  --E 30
```

Training Log

The training logs are recorded in the `training_log` directory.

Rule and Rule Arguments

- **Rule:**
 - 'noniid': Both Group and Client Non-IID
 - 'Dirichlet': Group IID and Client Non-IID
 - 'Mix2': Group Non-IID and Client IID
- **Rule Argument:**
 - Dirichlet parameter as shown in the manuscript.
- **com_amount:** Number of global communication rounds.
- **E:** Group aggregation period.
- **Relationship between H and the # epoch:** ($H = \frac{\text{number of samples at local dataset}}{\text{batch size}} \times \text{epoch}$)
- $\frac{1}{2}$
- Please refer to `utils_options.py` for more parameters.