*Type 1: Decentralized learning without RL*

1.DisPFL [origin code, decentralized, personalized by sparse masks, published on ICML 22] <https://github.com/rongdai/DisPFL/blob/master/fedml_api/standalone/dpsgd/client.py>

The main idea of this paper is implementing PFL by personalized sparse masks on decentralized but staic toplogy without RL.

文本

描述已自动生成

Same as DspodFL after adding parameter agent\_id to Class Agent()

文本

描述已自动生成

This repository also contains other PFL baselines such as Ditto (centralized personalized FL), FOMO (centralized personalized FL).

2. p2pfl [decentralized FL, no personalization, no RL, **too long!** ] and its fork fedstellar <https://github.com/pguijas/p2pfl/tree/main>

I think this code is too long (2000 lines) for just realizing decentralized FL across physical and virtualized devices, not recommended.

3. DsPodFL [decentralized learning, dynamic graph, no RL]

the decentralized consensus learning over changing topology code we are using

<https://github.com/ShahryarBQ/DSpodFL/blob/main/src/DSpodFL.py>

RGG radius

It contains aggregation weight wij and aggregation probability vij, convenient for our extension

Runs on python 3.8.6 python 3.11 fails

4. Basegraph [decentralized consensus learning, dynamic graph] <https://github.com/yukiTakezawa/BaseGraph>

Code for Beyond Exponential Graph: Communication-Efficient Topologies for Decentralized Learning via Finite-time Convergence (NeurIPS 2023)

The code is short, recommended.

5. FLAGS-FL [decentralized learning] nothing in particular

<https://github.com/ahnaflodhi/FLAGS-FL/tree/main>

6. SFL-Structural-Federated-Learning [decentralized PFL, contains many benchmarks such as scaffold and FedPro for comparison] <https://github.com/dawenzi098/SFL-Structural-Federated-Learning>

Personalized Federated Learning with Graph, IJCAI 2022

*Type 2: Centralized Learning without RL*

7. Microsft PersonalizedFL Baselines [centralized, personalized] <https://github.com/microsoft/PersonalizedFL/tree/main>

Contains FedAvg , FedProx, FedBN, FedAP, MetaFed, FedCLIP.

8. pFedMe [centralized, personalized by regularization] <https://github.com/CharlieDinh/pFedMe/tree/master>

Origin code for Personalized Federated Learning with Moreau Envelopes (NeurIPS 2020)

9. FedALA [centralized, personalized, learns optimal local aggregation weight, simple code] <https://github.com/TsingZ0/FedALA/blob/main/system/flcore/servers/serverALA.py>

AAAI 2023 accepted paper, FedALA: Adaptive Local Aggregation for Personalized Federated Learning

10. Ditto [centralized, personalized by regularization] <https://github.com/litian96/ditto/blob/master/flearn/models/client.py>

11. Mocha(in matlab) [Federated Multi-Task Learning, compared with COCOA]

<https://github.com/gingsmith/fmtl/tree/master>

*Type 3: Learning with RL*

*(a). RL-assisted FL*

12. flsim\_dqn [centralized FL (server as RL agent), handwritten DQN, no gym, no personalization] <https://github.com/tian1327/flsim_dqn>

Reproduction of paper: Optimizing FL on Non-IID Data with RL-assisted client selection, Infocom 2020.

With detailed reproduction report. Forked from iQua/flsim (origin code without RL).

13. MFPO-INFOCOM24 [centralized FL, RL server, gym, handwritten RL]

<https://github.com/HansenHua/MFPO-INFOCOM24/tree/main>

A Framework for Federated Reinforcement Learning with Interaction and Communication Efficiency, Infocom 2024 accpted.

14. FLDRL-in-Wireless-Communication [RL server, no gym, handwritten DQN] <https://github.com/Mauriyin/FLDRL-in-Wireless-Communication/tree/master>

Apply Deep Reinforcement Learning aided by Federated Learning with Multiple Access Channel

15. FLASH-RL [handwritten RL server, no gym] <https://github.com/Sofianebouaziz1/FLASH-RL/tree/main>

RL assisted client selection in FL. Compared with FedProx and Favor.

*(b.) Federated training of multiple RL agents*

16. Federated-DRL [handwritten multiple RL agents, use gym]

<https://github.com/TroddenSpade/Federated-DRL/tree/main>

Federated training of multiple RL agents. Useful to us.

17. federated-model-averaging-for-DQN [handwritten DQN agents, use gym]

Federated training of multiple RL agents. Useful to us.

18. federated\_learning\_rl [multiple RL agents, use Tianshou, use gym] <https://github.com/minhrongcon2000/federated_learning_rl>

Apply federated averaging to RL agents. Useful to us.

Conclusion: None of these codes tackles decentralized RL-assisted learning.