

1. ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ ФРЕЙМВОРКА

1.1 Федеративные режимы

Фреймворк имеет несколько платформ федеративного обучения:

- **FedML Octopus** предназначена для кросс-сило
- **FedML Parrot** предназначена для симуляции федеративного обучения и проведении быстрых экспериментов
- **FedML Beehive** предназначена для кросс-девайс
- **FedML Spider** предназначена для обучения на приватных данных через браузер
- **FedML Cheetah** предназначена для обучения в облаке

1.2 Типы данных

Фреймворк работает с изображениями, текстом на естественном языке, табличными данными.

В FedML есть уже загруженные датасеты, по многим из них на github есть примеры обучения. На github перечислены следующие встроенные датасеты:

Компьютерное зрение (изображения)

- MNIST
- cifar10
- cifar100
- fed_cifar100
- fed_emnist
- cinic10
- ImageNet
- Landmarks

Обработка естественного языка

- shakespeare
- fed_shakespeare
- stackoverflow

Автономное вождение (распознавание объектов)

- KITTI
- CityScapes
- Waymo
- Lyft Level 5
- UCB BDD100K
- ArgoVerse
- nuScenes

Финансовые данные (табличная форма)

- lending_club_loan

Другие

- NUS_WIDE (изображения)
- UCI (изображения)
- Synthetic
- edge_case_examples

1.3 Модели анализа данных

Из классических моделей машинного обучения во фреймворке есть только логистическая регрессия.

Что касается нейронных сетей, на гитхабе есть простые примеры с использованием CNN (сверточная нейронная сеть), RNN(рекуррентная), MobileNet, Resnet56, Resnet20, Generating adversarial networks, DART.

Так же на гитхабе есть примеры приложений , разработанных на платформе FedML.

Для компьютерного зрения есть модели:

1. Классификация изображений:
 - CNN
 - DenseNet
 - MobileNetv3
 - EfficientNet
2. Сегментация изображений
 - UNet
 - DeeplabV3

- TransUnet
3. Распознавание объектов
- YOLOv5

Для обработки естественного языка:

- Классификация текстов
- Распознавание команд (seq tagging, seq2seq)
- Извлечение фрагментов текста (span extraction)

Федеративное обучение для графовых нейронных сетей:

1. На уровне графов (Graph-Level)
 - i. Предсказание свойств молекул (MoleculeNet Property Prediction)
 - Классификация графов
 - Регрессия графов
 - ii. Социальные сети (Social Networks)
 - Классификация графов
2. На уровне подграфов (Subgraph-Level)
 - i. Рекомендательные системы (Recommendation Systems)
 - Предсказание связей (Link Prediction)
3. На уровне узлов (Node-Level)
 - i. Сети Ego(Citation & Coauthor Networks)
 - Предсказание связей
 - Классификация узлов

1.4 Функции агрегирования

Согласно константам на гитхабе, доступны следующие алгоритмы федеративного обучения: FedAvg, FedOpt, FedProx, classical_vertical, split nn, decentralized FL, FedGan, FedAvg robust, FedAvg seq, FedOpt seq, FedGKT, FedNAS, FedSeg, turbo_aggregate, FedNova, FedDyn, SCAFFOLD, Mime, Hierarchical FL, FedSGD, Fed Local SGD, Async FedAvg.

Больше всего примеров есть с FedAvg, остальные в основном использовались только в режиме симуляции.

1.5 Защита конфиденциальности

Есть дифференциальная приватность (механизм Гаусса и Лапласа). Так же есть информация о проведении тестов со следующими типами атак и механизмами защиты:

Attack

1. ByzantineAttack: (1) zero mode (2) random mode (3) flip mode
2. (NeurIPS 2019) DLGAttack: "Deep leakage from gradients" <https://proceedings.neurips.cc/paper/2019/file/60a6c4002cc7b29142def8871531281a-Paper.pdf>
3. (NeurIPS 2020) InvertAttack: "Inverting gradients-how easy is it to break privacy in federated learning?" <https://github.com/JonasGeiping/invertinggradients/>
4. LabelFlippingAttack: "Data Poisoning Attacks Against Federated Learning Systems" <https://arxiv.org/pdf/2007.08432>
5. (NeurIPS 2021) RevealingLabelsFromGradientsAttack: "Revealing and Protecting Labels in Distributed Training" <https://proceedings.neurips.cc/paper/2021/file/0d924f0e6b3fd0d91074c22727a53966-Paper.pdf>
6. (NeurIPS 2019) BackdoorAttack: "A Little Is Enough: Circumventing Defenses For Distributed Learning" <https://proceedings.neurips.cc/paper/2019/file/ec1c59141046cd1866bbcbdfb6ae31d4-Paper.pdf>
7. (NeurIPS 2020) EdgeCaseBackdoorAttack: "Attack of the Tails: Yes, You Really Can Backdoor Federated Learning" <https://proceedings.neurips.cc/paper/2020/file/b8ffa41d4e492f0fad2f13e29e1762eb-Paper.pdf>
8. (PMLR'20) ModelReplacementBackdoorAttack: "How To Backdoor Federated Learning" <http://proceedings.mlr.press/v108/bagdasaryan20a/bagdasaryan20a.pdf>

Рисунок 1 – Типы атак

Defense

1. (PMLR 2018) BulyanDefense: "The Hidden Vulnerability of Distributed Learning in Byzantium." <http://proceedings.mlr.press/v80/mhamdi18a/mhamdi18a.pdf>
2. CClipDefense: "Byzantine-Robust Learning on Heterogeneous Datasets via Bucketing" <https://arxiv.org/pdf/2006.09365.pdf>
3. GeometricMedianDefense: "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent." <https://dl.acm.org/doi/pdf/10.1145/3154503>
4. (NeurIPS 2017) KrumDefense: "Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent" <https://papers.nips.cc/paper/2017/file/f4b9ec30ad9f68f89b29639786cb62ef-Paper.pdf>
5. (ICLR 2021) MultiKrumDefense: "Distributed momentum for byzantine-resilient stochastic gradient descent" <https://infoscience.epfl.ch/record/287261>
6. NormDiffClippingDefense: "Can You Really Backdoor Federated Learning?" <https://arxiv.org/pdf/1911.07963.pdf>
7. (AAAI 2021) RobustLearningRateDefense: "Defending against backdoors in federated learning with robust learning rate." <https://github.com/TinfoilHat0/Defending-Against-Backdoors-with-Robust-Learning-Rate>
8. SoteriaDefense: "Provable defense against privacy leakage in federated learning from representation perspective." <https://arxiv.org/pdf/2012.06043>
9. SLSGDDefense: "SLSGD: Secure and efficient distributed on-device machine learning" <https://arxiv.org/pdf/1903.06996.pdf>
10. RFA_defense: "Robust Aggregation for Federated Learning" <https://arxiv.org/pdf/1912.13445>
11. (USENIX2020) FoolsGoldDefense: "The Limitations of Federated Learning in Sybil Settings" <https://www.usenix.org/system/files/raid20-fung.pdf>
12. (ICML 2018) CoordinateWiseMedianDefense: "Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates" <http://proceedings.mlr.press/v80/yin18a/yin18a.pdf>
13. (NeurIPS 2021) WbcDefense: "Enhancing Robustness against Model Poisoning Attacks in Federated Learning from a Client Perspective" <https://arxiv.org/abs/2110.13864>
14. (ICML 2021) CRFLDefense: "CRFL: Certifiably Robust Federated Learning against Backdoor Attacks" <http://proceedings.mlr.press/v139/xie21a/xie21a.pdf>

Рисунок 2 – Реализованные механизмы защиты

1.6 Распределения данных

Согласно константам на гитхабе, для кросс-сило поддерживается только горизонтальное и иерархическое распределение.

Для режима симуляции поддерживается так же централизованное, децентрализованное, вертикальное и сплит распределение.

FedML Parrot supports representative algorithms in different communication topologies (as the figure shown below), including Fedvg, FedOpt (ICLR 2021), FedNova (NeurIPS 2020), FedGKT (NeurIPS 2020), Decentralized FL, Vertical FL, Hierarchical FL, FedNAS, and Split Learning.

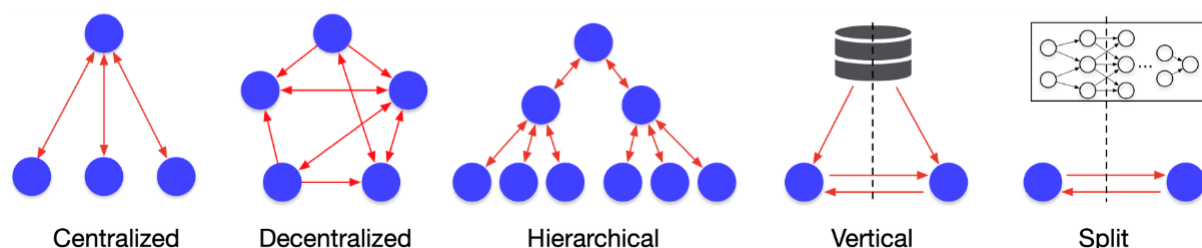


Рисунок 3 – Распределения для симуляционного режима

1.7 Метрики

Для простых примеров на встроенных датасетах рассчитывается только accuracy и loss, расчет метрик находится в файле [my_server_aggregator_prediction.py](#). Судя по объявлению словаря с метриками, планировалось так же реализовать расчет precision и recall, но пока их нет.

Для некоторых продвинутых приложений, примеры которых находятся в папке app, прописывается расчет других метрик. Например, precision, recall, confusion matrix. ([Пример расчета метрик для модели компьютерного зрения](#))

1.8 Простота настройки

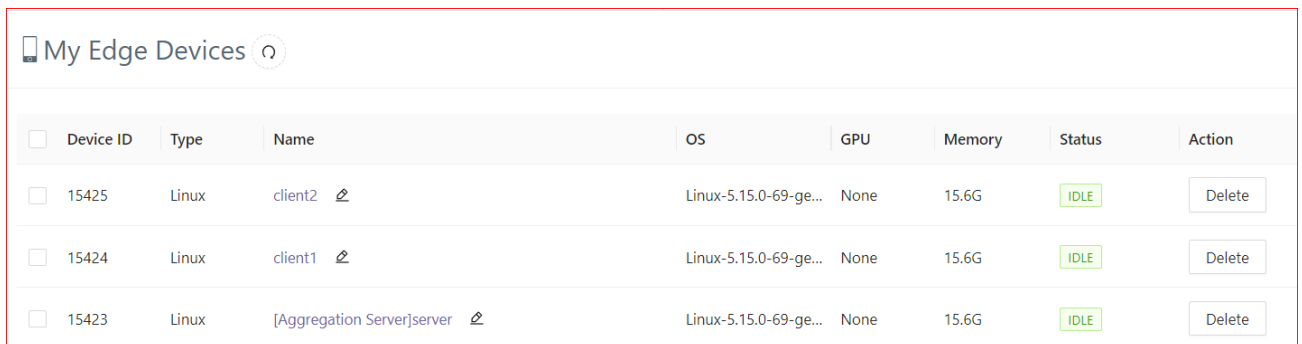
Документация скудная. Нет описаний функций, только общие описания архитектуры, способов применения. Информацию о данных, моделях, безопасности, метриках приходится искать по разным папкам на гитхабе.

Настройка обучения сама по себе удобная и несложная, но найти информацию о настройке очень сложно. По возможным проблемам и ошибкам в процессе настройки так же нет никакой информации. Коммьюнити не активное.

2. ЭКСПЕРИМЕНТЫ

Была проведена настройка платформы FedML для федеративного обучения и запущен пример, согласно следующим шагам:

1. На виртуальные машины необходимо установить FedML:
`$ pip install fedml`
2. Нужно зарегистрироваться на сайте FedML для доступа к MLOps
<https://open.fedml.ai/octopus/profile>
3. Прикрепить машины к MLOps с помощью команды (Account ID указан в профиле):
`$ fedml login < Account ID >`
4. Чтобы подключить машину в качестве сервера:
`$ fedml login < Account ID > -s`
5. Проверить подключение машин можно во вкладке Edge device. Тут же можно переименовать машины для удобства



<input type="checkbox"/>	Device ID	Type	Name	OS	GPU	Memory	Status	Action
<input type="checkbox"/>	15425	Linux	client2 🔗	Linux-5.15.0-69-ge...	None	15.6G	IDLE	Delete
<input type="checkbox"/>	15424	Linux	client1 🔗	Linux-5.15.0-69-ge...	None	15.6G	IDLE	Delete
<input type="checkbox"/>	15423	Linux	[Aggregation Server]server 🔗	Linux-5.15.0-69-ge...	None	15.6G	IDLE	Delete

Рисунок 4 – Подключенные устройства

6. На любой из машин создать следующую структуру проекта ([пример](#)):

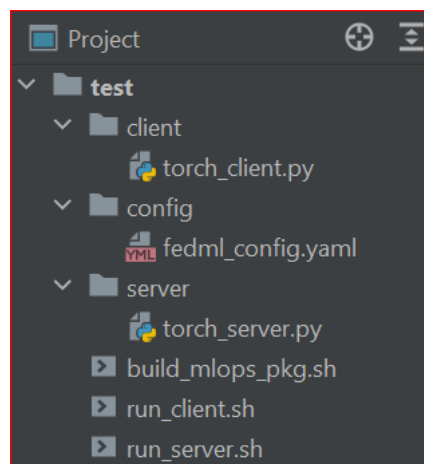


Рисунок 5 – Структура проекта

7. В файле указать необходимую конфигурацию: количество клиентов, датасет, распределение и т.д.
8. Запустить генерацию zip-файлов для MLOps:

```
$ bash build_mlops_pkg.sh
```

9. Скачать сгенерированные файлы на локальный компьютер. Скачать файлы с виртуальной машины на Windows можно следующей командой (виртуальная машина должна быть запущена):

```
pscp [имя пользователя]@[имя сервера/ip-адрес]:[путь к файлу в Linux]  
[путь к файлу в Windows]
```

```
C:\Users\User>pscp etu@158.160.44.128:/home/etu/test/mlops/dist-packages/*.zip .  
server-package.zip      | 2 kB | 2.3 kB/s | ETA: 00:00:00 | 100%  
client-package.zip      | 2 kB | 2.3 kB/s | ETA: 00:00:00 | 100%
```

Рисунок 6 – Скачивание файлов на Windows

10. Создать новый application в MLOps, прикрепив сгенерированные zip-файлы

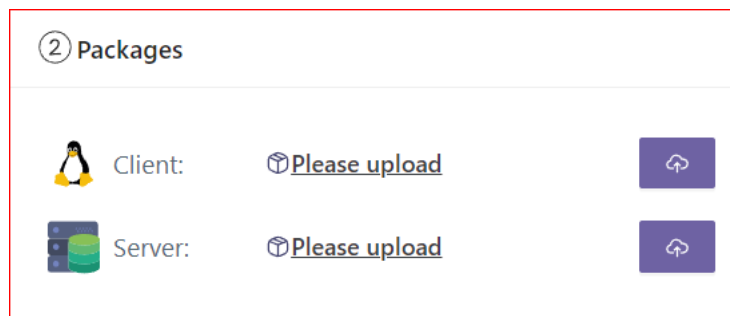


Рисунок 7 – Прикрепление zip-файлов

11. Создать новый проект
12. Зайти в проект и создать новый запуск, выбрав необходимые устройства и указав созданный ранее application. Так же здесь, если нужно, можно будет изменить конфигурацию
13. После этого начнется обучение, когда завершатся все раунды и устройства получат статус Finished, можно будет посмотреть результаты обучения в соответствующих вкладках.
14. Если возникает проблема, что статусы клиентов отображаются как Queued и обучение не начинается, можно попробовать удалить

устройства на вкладке Edge device, присоединить их заново, а потом начать новый запуск

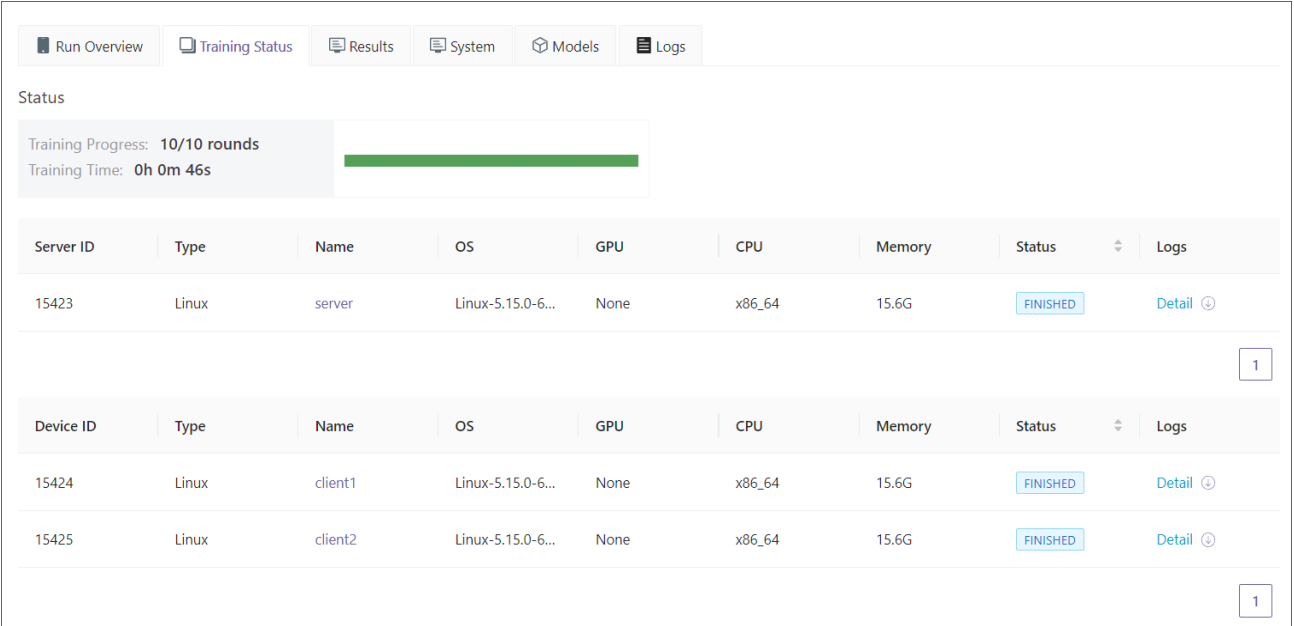


Рисунок 8 – Результаты обучения

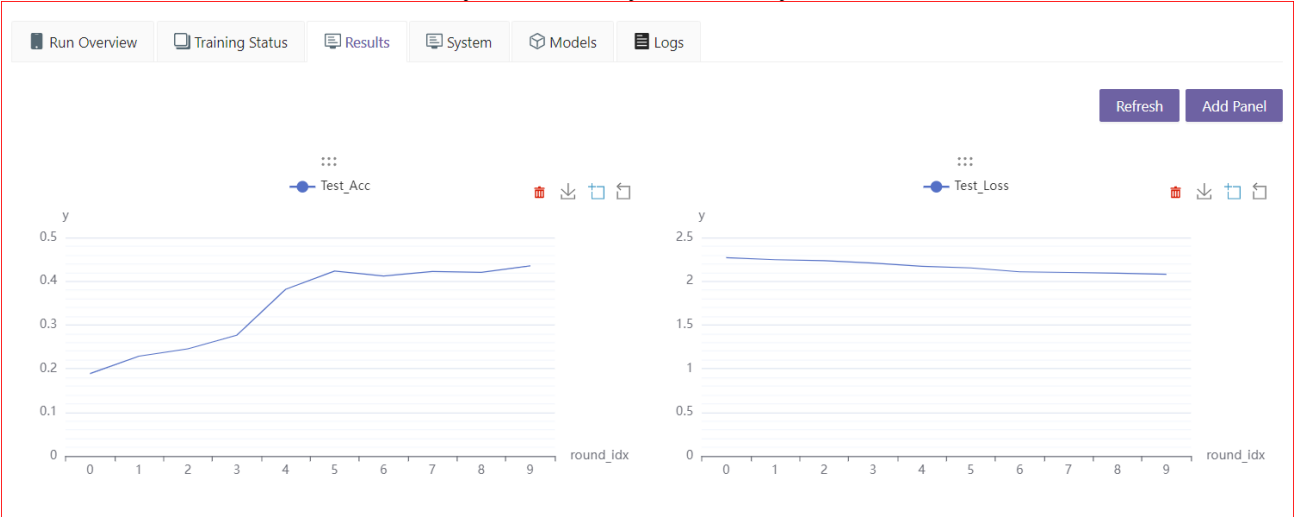


Рисунок 9 – Результаты обучения. Метрики

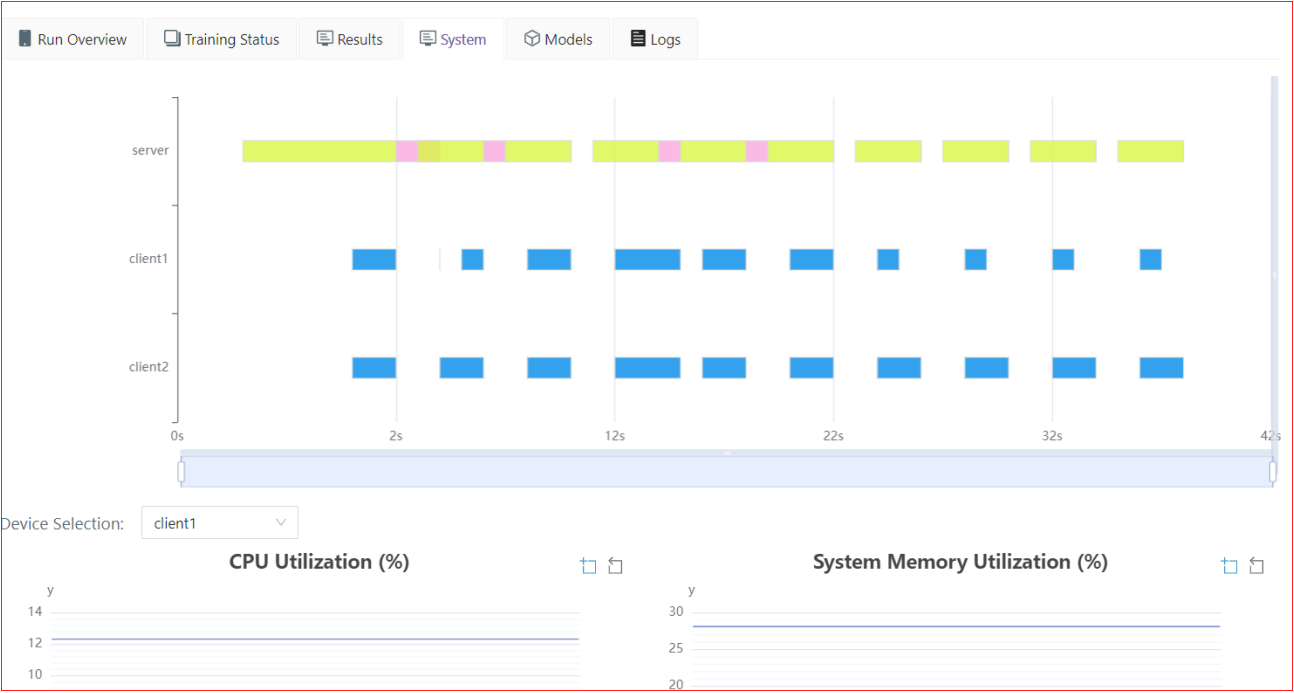


Рисунок 10 – Результаты обучения. Система

Run Overview Training Status Results System Models Logs

Aggregated Model Client Model

RunId	Round_idx	CreateTime	Action
8567	10	05-03-2023	Download Create Model Card Update Model
8567	9	05-03-2023	Download Create Model Card Update Model
8567	8	05-03-2023	Download Create Model Card Update Model
8567	7	05-03-2023	Download Create Model Card Update Model
8567	6	05-03-2023	Download Create Model Card Update Model
8567	5	05-03-2023	Download Create Model Card Update Model
8567	4	05-03-2023	Download Create Model Card Update Model

Рисунок 11 – Результаты обучения. Модели