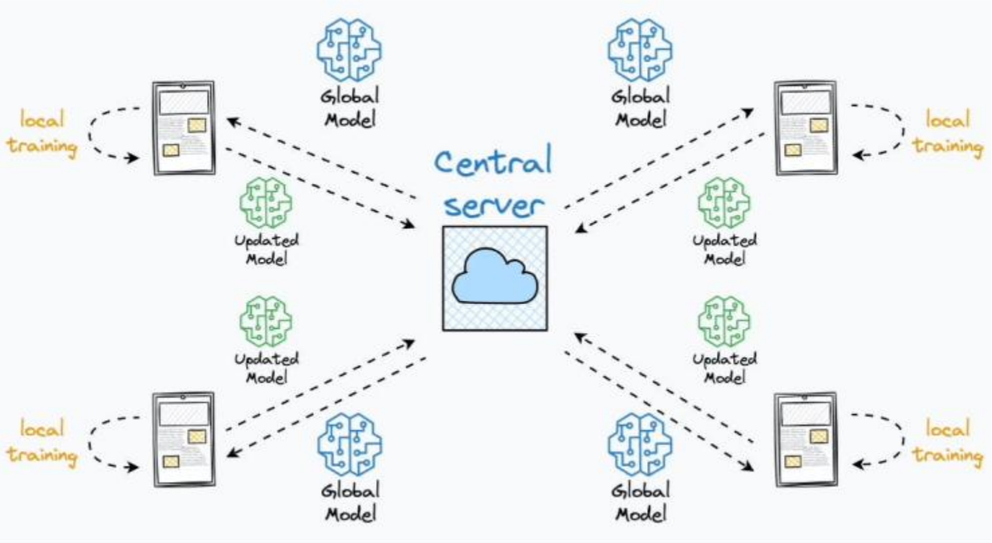
# Ditto(Fair and Robust Federated Learning Through Personalization)

20201780 이승재

20237140 남기용

# 목차

# Federated Learning 기본 구조




연합학습(FL)은 클라이언트 장치들이 데이터를 직접 공유하지 않고, 로컬에서 모델을 학습한 후 서버에 모델 업데이트만 전달하여 글로벌 모델을 학습하는 분산 학습 패러다임입니다.

대표적인 알고리즘인 FedAvg는 각 클라이언트의 로컬 모델 업데이트를 평균하여 글로벌 모델을 갱신합니다.

# 문제 정의

## Federated Learning 환경의 핵심 문제

Federated Learning(FL)은 프라이버시를 유지하며 모델을 공동 학습하기 위한 기술

그러나 현실의 FL 환경은 client 간 데이터가 매우 다름(non-IID) → 하나의 Global Model로 모든 클라이언트를 만족시키기 어려움

## Fairness 문제

분포가 다른 데이터(non-IID) → 특정 클라이언트 성능 극히 낮음

클라이언트 간 성능 편차가 크다

FL 평가에서 중요한 지표: Worst-case(Client-level) accuracy

## Robustness 문제

FL은 악의적 클라이언트의 공격에 취약

Label poisoning, Model Replacement
Random updates

악성 업데이트 하나로 전체 모델(global model)이 붕괴될 수 있음

Fairness를 위해 특정 그룹에 대해 성능을 조정하면, 전체 worst-case 성능에는 손해가 될 수 있음.
Robustness를 높이면 보통 모델이 보수적으로 학습되어 일부 소수 그룹 성능이 희생될 수 있음.
상충되는 제약 조건이다.

# Ditto: Fair and Robust Federated Learning Through Personalization

Ditto의 목적
Non-IID 데이터 환경에서 모든 클라이언트를 만족시키는 공정한(fair) 글로벌 모델 학습
공격자(adversarial client) 존재 시에도 robust 하면서, 각 클라이언트에게 개인화(personalization) 된 모델 제공

Ditto의 동작 방식
(1) 서버 → 클라이언트
서버는 현재 글로벌 모델을 클라이언트들에게 전송.

(2) 클라이언트: 두 가지 작업을 병행 수행
A. 전체를 위한 글로벌 학습 (FedAvg 동일)
1.서버가 준 모델로 로컬 SGD 수행
2.업데이트된 글로벌용 gradient/모델을 서버에 다시 전송 > 이는 전체 네트워크 성능을 위한 공익 활동

B.Ditto 추가 개인화 학습 단계 (핵심 부분)
서버 모델을 개인화 모델의 초기값으로 설정
로컬 데이터로 목적함수 최적화 : $F_k(v_k) + \frac{\lambda}{2}\|v_k - w^t\|^2$

로컬에서 개인화 모델만 업데이트(SGD)
이 개인화 모델은 서버로 보내지 않고 클라이언트에 저장

- $F_k(v_k)$ :
  → *내 데이터로 손실을 줄이는 개인화 학습(로컬 목표)*
  → 각 기기가 자기 데이터에 최적화된 "나만의 모델"을 학습
- $\|v_k - w^{\backslash *}\|^2$ :
  → 글로벌 모델과 너무 멀어지지 않게 하는 **정규화 항**
- $\lambda$ (균형 조절 파라미터):
  - **작을수록:** 개인화 강화 ("내 데이터 위주로 갈래" → Robustness ↑)
  - **클수록:** 글로벌 모델에 더 가까워짐 ("일반화가 더 필요해" → Fairness/Global consistency ↑)

## 실험 3종류

**실험1** — 기본 성능 + Fairness + Robustness 비교

**실험2** — 다른 personalization 방법들과의 직접 비교 (clean / 일부 adversarial setting)

**실험3** — Robustness 향상을 위한 확장 실험 — Ditto + robust aggregation or robust solver

## 실험1

### Methods (기법)

- global : FedAvg 전역 모델
- local : 각 클라이언트 로컬 모델
- fair (TERM, t=1) : fairness 최적화 baseline
- Ditto : 제안 방법 (개인화 + fairness + robustness)

### 데이터셋

FEMNIST, Fashion-MNIST

### 실험 구성

- clean 환경 + 다양한 공격(adversary ratio: 10~80%, 공격 유형 A1/A2/A3)
(A1 Label poisoning, A2 Random updates, A3 Model replacement)
- 각 방법의 평균 test accuracy ± 표준편차 측정
- Fairness 정의(Definition 2) 기준으로 평가

## 실험1 결과

| **Fashion** | | **A1** (ratio of adversaries) | | | **A2** (ratio of adversaries) | | | **A3** (ratio of adversaries) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Methods | clean | 20% | 50% | 80% | 20% | 50% | 80% | 10% | 20% | 50% |
| global | .911 (.08) | .897 (.08) | .855 (.10) | .753 (.13) | .900 (.08) | .882 (.09) | .857 (.10) | .753 (.10) | .551 (.13) | .275 (.12) |
| local | .876 (.10) | .874 (.10) | .876 (.11) | .879 (.10) | .874 (.10) | .876 (.11) | .879 (.10) | .877 (.10) | .874 (.10) | **.876 (.11)** |
| fair (TERM, $t$=1) | .909 (.07) | .751 (.12) | .637 (.13) | .547 (.11) | .731 (.13) | .637 (.14) | .635 (.14) | .653 (.13) | .601 (.12) | .131 (.16) |
| Ditto | **.943 (.06)** | **.944 (.07)** | **.937 (.07)** | **.907 (.10)** | **.938 (.07)** | **.930 (.08)** | **.913 (.09)** | **.921 (.09)** | **.902 (.09)** | .873 (.11) |
| **FEMNIST** | | **A1** (ratio of adversaries) | | | **A2** (ratio of adversaries) | | | **A3** (ratio of adversaries) | | |
| Methods | clean | 20% | 50% | 80% | 20% | 50% | 80% | 10% | 15% | 20% |
| global | .804 (.11) | .773 (.11) | .727 (.12) | .574 (.15) | .774 (.11) | **.703 (.14)** | .636 (.15) | .517 (.14) | .487 (.14) | .314 (.13) |
| local | .628 (.15) | .620 (.14) | .627 (.14) | .607 (.14) | .620 (.14) | .627 (.14) | .607 (.14) | .622 (.14) | .621 (.14) | **.620 (.14)** |
| fair (TERM, $t$=1) | .809 (.11) | .636 (.15) | .562 (.13) | .478 (.12) | .440 (.15) | .336 (.12) | .363 (.12) | .353 (.12) | .316 (.12) | .299 (.11) |
| Ditto | **.834 (.09)** | **.802 (.10)** | **.762 (.11)** | **.672 (.13)** | **.801 (.09)** | .700 (.15) | **.675 (.14)** | **.685 (.15)** | **.650 (.14)** | .613 (.13) |

Ditto는 clean 환경에서도 baseline 대비 높은 accuracy 유지

공격 환경에서도 성능 안정적 → robustness 확보

TERM baseline은 fairness는 높지만 accuracy 저하 → Ditto는 accuracy와 fairness 동시 확보

## 실험2

### Methods (기법)

- plain finetuning
- L2SGD
- EWC
- SKL
- Per-FedAvg (HF)
- mapper
- APFL
- Ditto

### 데이터셋

FEMNIST, CelebA

### 실험 구성

- clean + 일부 adversarial (50% 공격자, A1) 조건에서 각 방법 성능 평가
- 평균 test accuracy ± 표준편차 비교

## 실험2 결과

| | Clean | | 50% Adversaries (A1) | |
|---|---|---|---|---|
| Methods | **FEMNIST** | **CelebA** | **FEMNIST** | **CelebA** |
| global | .804 (.11) | .911 (.19) | .727 (.12) | .538 (.28) |
| local | .628 (.15) | .692 (.27) | .627 (.14) | .682 (.27) |
| plain finetuning | .815 (.09) | .912 (.18) | .734 (.12) | .721 (.28) |
| L2SGD | .817 (.10) | .899 (.18) | .732 (.15) | .725 (.25) |
| EWC | .810 (.11) | .910 (.18) | .756 (.12) | .642 (.26) |
| SKL | .820 (.10) | **.915 (.16)** | .752 (.12) | .708 (.27) |
| Per-FedAvg (HF) | .827 (.09) | .907 (.17) | .604 (.14) | **.756 (.26)** |
| mapper | .792 (.12) | .773 (.25) | .726 (.13) | .704 (.27) |
| APFL | .811 (.11) | .911 (.17) | .750 (.11) | .710 (.27) |
| Ditto | **.836 (.10)** | .914 (.18) | **.767 (.10)** | .721 (.27) |

Ditto는 대부분 환경에서 다른 개인화 방법 대비 높은 accuracy

클라이언트별 데이터 heterogeneity에 강함

최신 personalization baseline 대비 성능 우수성 입증

fairness 문제(클라이언트별 성능 편차 최소화)에 긍정적 효과

개인화 방법으로 robustness/accuracy 균형을 잘 잡음

글로벌 모델에서 시작하여 각 로컬 장치에서 50 에포크 동안 파인튜닝
최적 하이퍼파라미터를 사용
Ditto의 단순성에도 불구하고, 아래 표 2에서 볼 수 있듯이 Ditto는
다른 개인화 방법들과 비교해 유사하거나 더 우수한 테스트 정확도를
달성하며 표준 편차는 약간 낮습니다.

## 실험3

### Methods (기법)

- Ditto 단독
- Ditto + clipping
- Ditto + multi-Krum
- robust aggregation variants

### 데이터셋

FEMNIST, Fashion-MNIST

### 실험 구성

- 다양한 adversarial 환경에서 평가
- α, β 등 모델 hyperparameter 변경 → 성능 변화 관찰
- Ditto 단독 대비, 추가 방어기법 효과 분석

## 실험3 결과

| FEMNIST | A1 | | A2 | | A3 | |
|---|---|---|---|---|---|---|
| Methods | 20% | 80% | 20% | 80% | 10% | 20% |
| global | .773 | .574 | .774 | .636 | .517 | .364 |
| clipping | .791 | .408 | .791 | .656 | .795 | .061 |
| Ditto | .803 | **.669** | .792 | .681 | .695 | .650 |
| Ditto + clipping | **.810** | .645 | **.808** | **.684** | **.813** | **.672** |

Ditto 단독으로도 Robustness 확보 가능

Robust aggregation 적용 시 성능 소폭 향상

Ablation을 통해 α, β 등 구성 요소가 성능에 미치는 영향 확인

Robustness 향상 가능성 + 구성 요소 기여도 분석

# 재구현

1.anaconda를 사용하여 로컬pc로 재구현

2.최신 버전의 tensorflow와 파이썬이 사용 불가하여 tensorflow의 버전을 1.1x 버전으로 낮추고 파이썬 버전을 낮춤

3.ai 도움으로 main.py 수정하여 경로인식

4.원본 깃허브에 다운로드가 가능했던 femnist와 외부에서 찾은 vehicle데이터 셋을 사용

5.최신 그래픽 카드가 TensorFlow 1.x가 의존하는 CUDA를 사용할 수 없어 논문에 나온 하이퍼 파라미터 값으로 진행하니 학습에 많은 시간이 소요

6.가장 가벼운 vehicle로 먼저 진행하고자 학습 가능한 형태로 전처리해서 실행

7.femnist를 기본 하이퍼 파라미터 값인 5000라운드를 진행하니 점수가 꾸준히 향상된다는 것을 확인.

vehicle데이터 셋 결과

At round 2000 training accu: 0.43885865734344603,
loss: 0.6438063433284127
At round 2000 test accu: 0.276460894916973
At round 2000 malicious test accu: 0.09011396766498807
At round 2000 benign test accu: 0.44727891156462585
variance of the performance:  0.0005743898281233917

femnist데이터 셋 결과

At round 5000 training accu: 0.8915515344174104,
loss: 0.33905118488465513
At round 5000 test accu: 0.7974987974987975
At round 5000 malicious test accu: nan
At round 5000 benign test accu: 0.7974987974987975
variance of the performance:  0.011218454876114696

# 개선점

1.핵심 파라미터 개선: 동적  스케줄링

Ditto의 핵심은 균형 조절 파라미터의 값 논문은 균형 조절 파라미터 =1 등 고정된 값을 사용하거나 제한된 범위에서 선택

문제점: 고정된 값은 모델 학습의 유연성을 떨어뜨린다.

개선 목표: 시간 경과에 따라  값을 변화시켜 공정성/강건성 이점을 극대화

구현 방법 : ditto.py 파일을 수정하여  값이 초기에는 작았다가 후반으로 갈수록 커지는 스케줄을 구현

2.  견고성 확장: Ditto + 손실 기반 가중치 재조정

현재는 Ditto가 공격을 당했을 때 생존하는 능력만 봤지만 서버가 '누가 악성 클라이언트인지' 미리 판단하고 가중치를 다르게 주면 방어력이 훨씬 높아질 수 있다.

문제점: 현재의 Ditto/FedAvg는 모든 클라이언트 업데이트를 동일하게 평균화

개선 목표: 클라이언트의 학습 결과에 따라 신뢰도 점수를 부여하고, 신뢰도가 높은 클라이언트의 업데이트를 더 많이 반영

구현 방법 (Aggregation 수정): ditto.py의 집계(Aggregation) 로직을 수정합니다.

아이디어: 클라이언트가 보고한 로컬 손실(Loss)을 기준으로 가중치를 주고 Loss가 비정상적으로 높거나 낮은 클라이언트(악성 유저일 가능성 높음)의 업데이트는 가중치를 낮춘다.