Data Privacy와 Federated Learning

CM 1-2팀 박주원

딥러닝과 Data Privacy

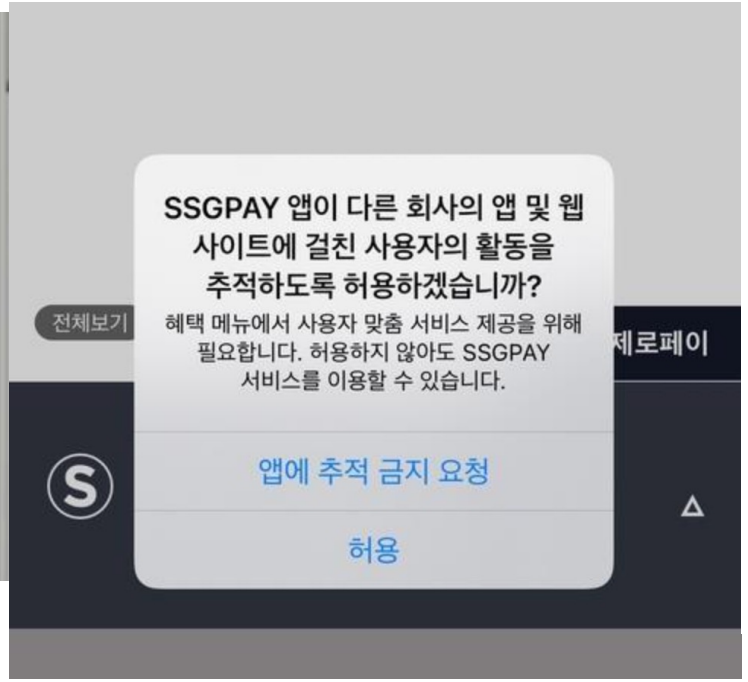
- 딥러닝 모델의 성능 향상을 위해 매우 크고 다양한 데이터셋이 필요
- 다양한 로컬 디바이스에서 생성된 데이터 수집을 요구
- 사용자의 로컬 디바이스에 저장되어 있는 데이터를 서버로 전송되고, 수집된 데이터를 활용하여 서버에서 딥러닝 모델을 학습
- 의료데이터와 같이 법적규제로 인공지능에 활용하기 어려운 민감한 데이터 수집 어려움
- 데이터 누출은 주로 로컬 – 서버로 전송하는 과정에서 일어나고, 클라우드와 같은 서버가 해킹 당할 경우 데이터 누출이 발생

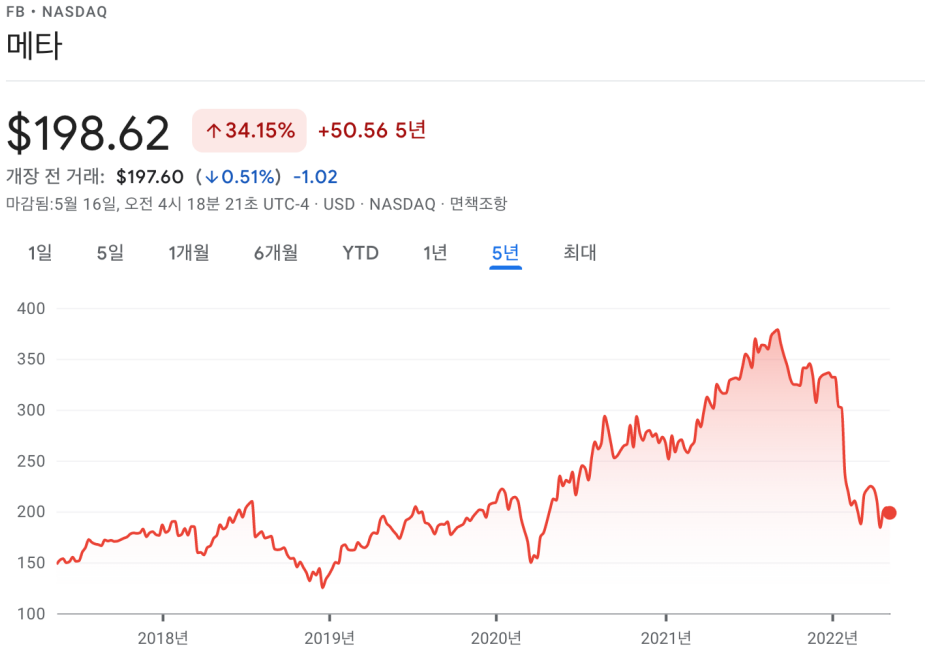
국내 마이데이터 서비스

- 주문내역 수집은 사생활 침해
- 나이키 에어포스 69,000원 구매
- 스포츠 브랜드 69,000원 구매

애플 개인정보보호 정책과 페이스북

- iOS 14.5 개인정보보호 정책 업데이트



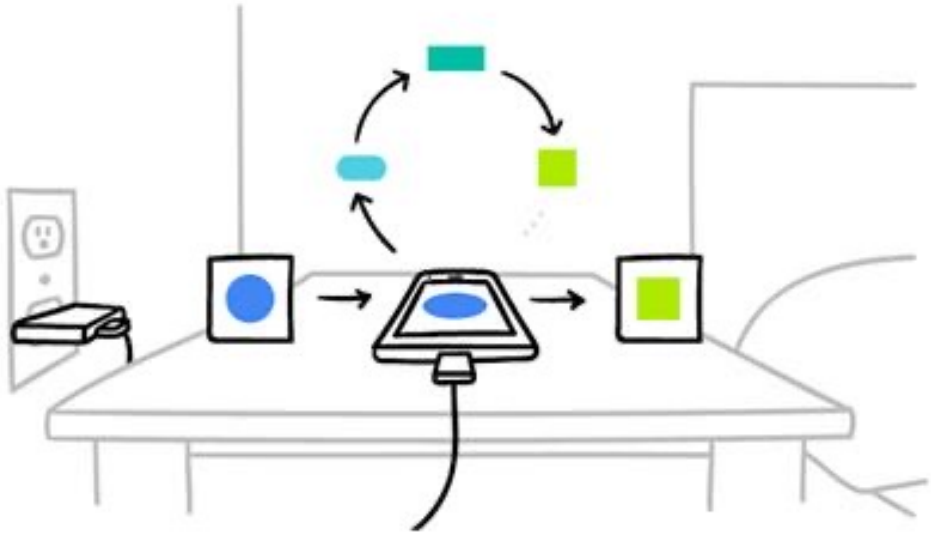


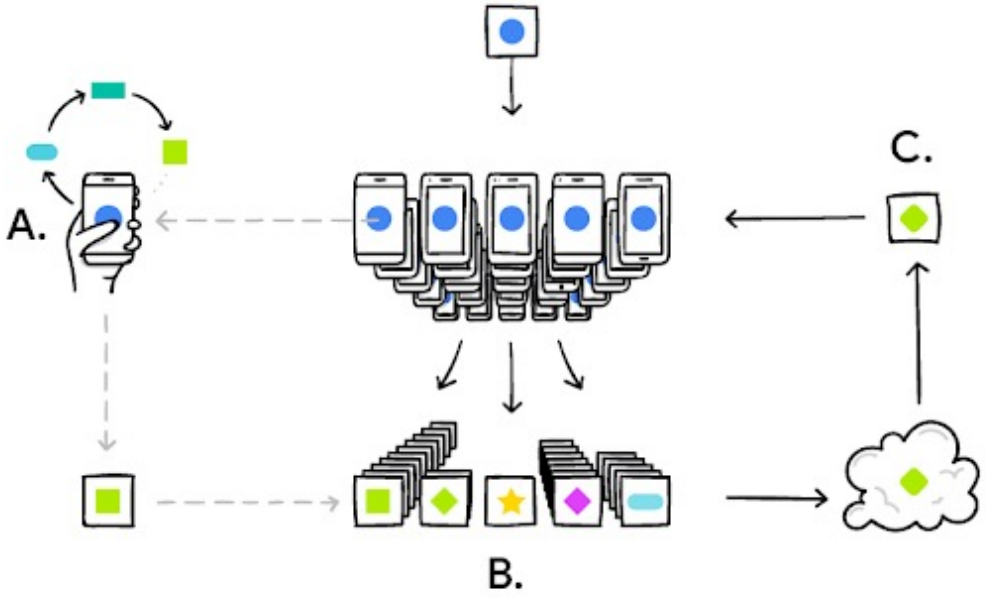
그외 딥러닝 이슈

- 방대한 데이터를 처리하기 위한 막대한 컴퓨팅 자원 필요
- 수집하는 데이터가 사용자 개인 맞춤형이 될 수 없을까?

Federated Learning

- Data Privacy를 위해 구글에서 2017년에 논문을 통해 Federated Learning 발표





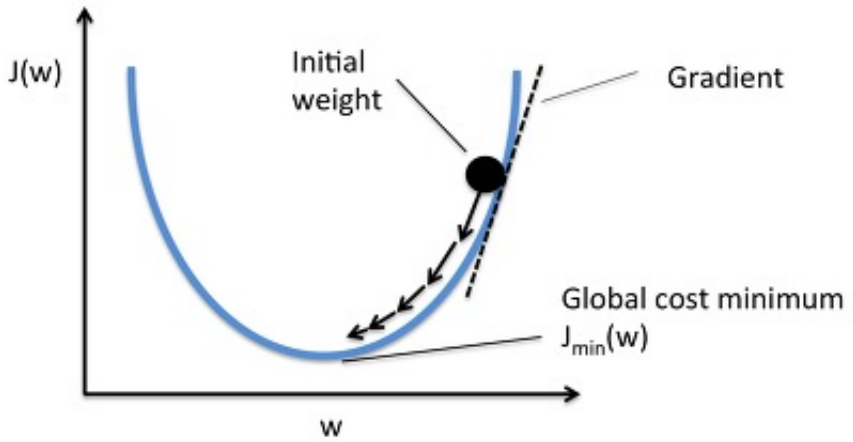
Federated Learning 사용 사례

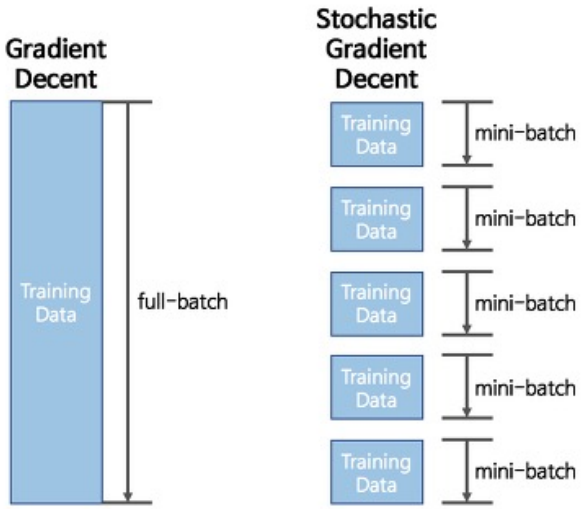
- 구글 키보드 “Gboard”
- 구글 Assistant “Hey, Google”
- 인텔 랩 x 펜실베니아 대학교 페럴만 의대
- 엔비디아의 의료 인공지능 시스템 ‘클라라 연합학습’
- 그 외 금융, 생체인증, 제조업, 헬스케어 등 다양한 분야에서 활용 중

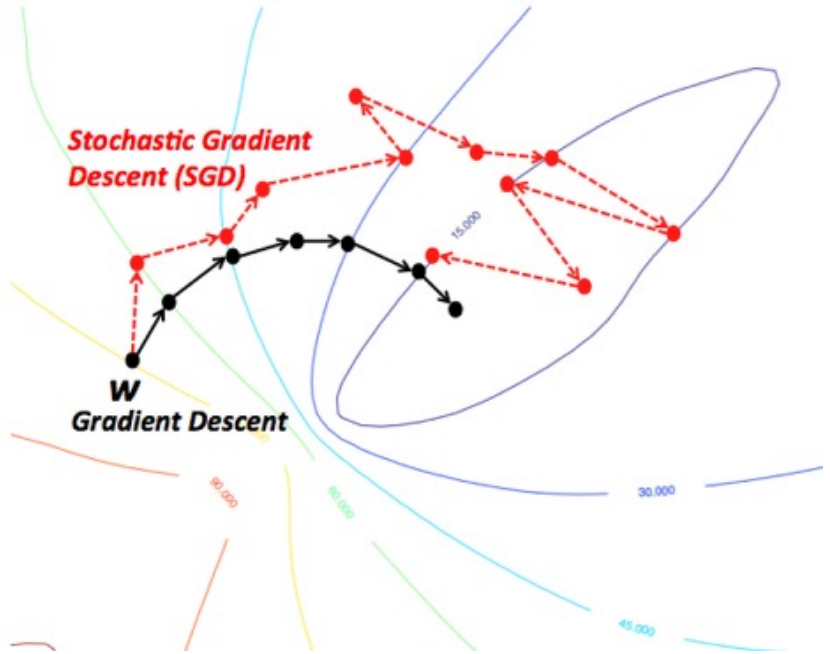
Federated Learning 주요 이슈

- Communication Overhead
  - 모든 디바이스가 중앙화 서버와 통신하므로 Bottleneck 발생
- System Heterogeneity
  - 디바이스 마다 CPU, 배터리 수명, 셀룰러 타입 등이 다름
- Statistical Heterogeneity
  - 디바이스 마다 데이터셋의 크기, 유형 등이 다름 (Non-IID)
  - the training data are not identically and independently distributed

Stochastic Gradient Decent (SGD)



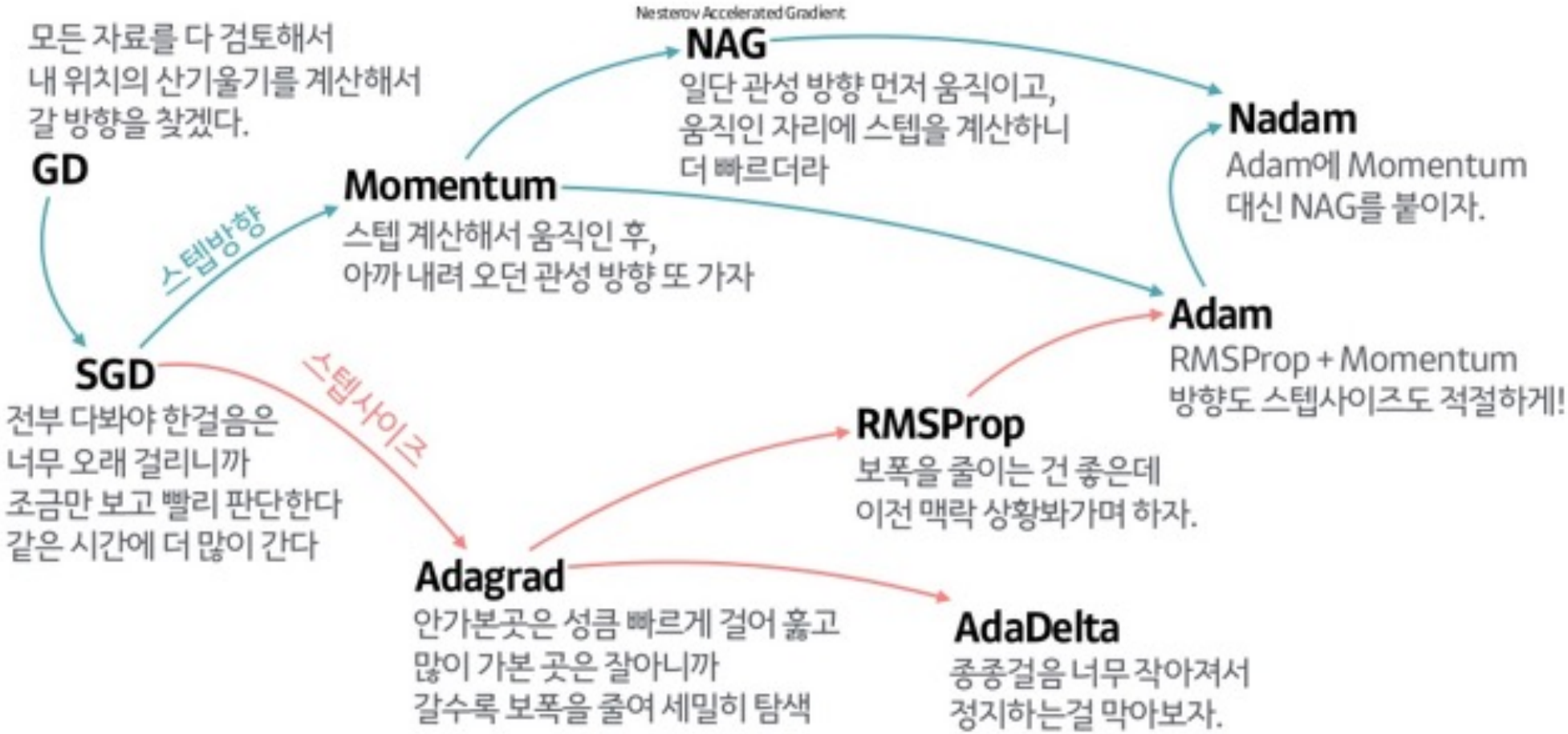




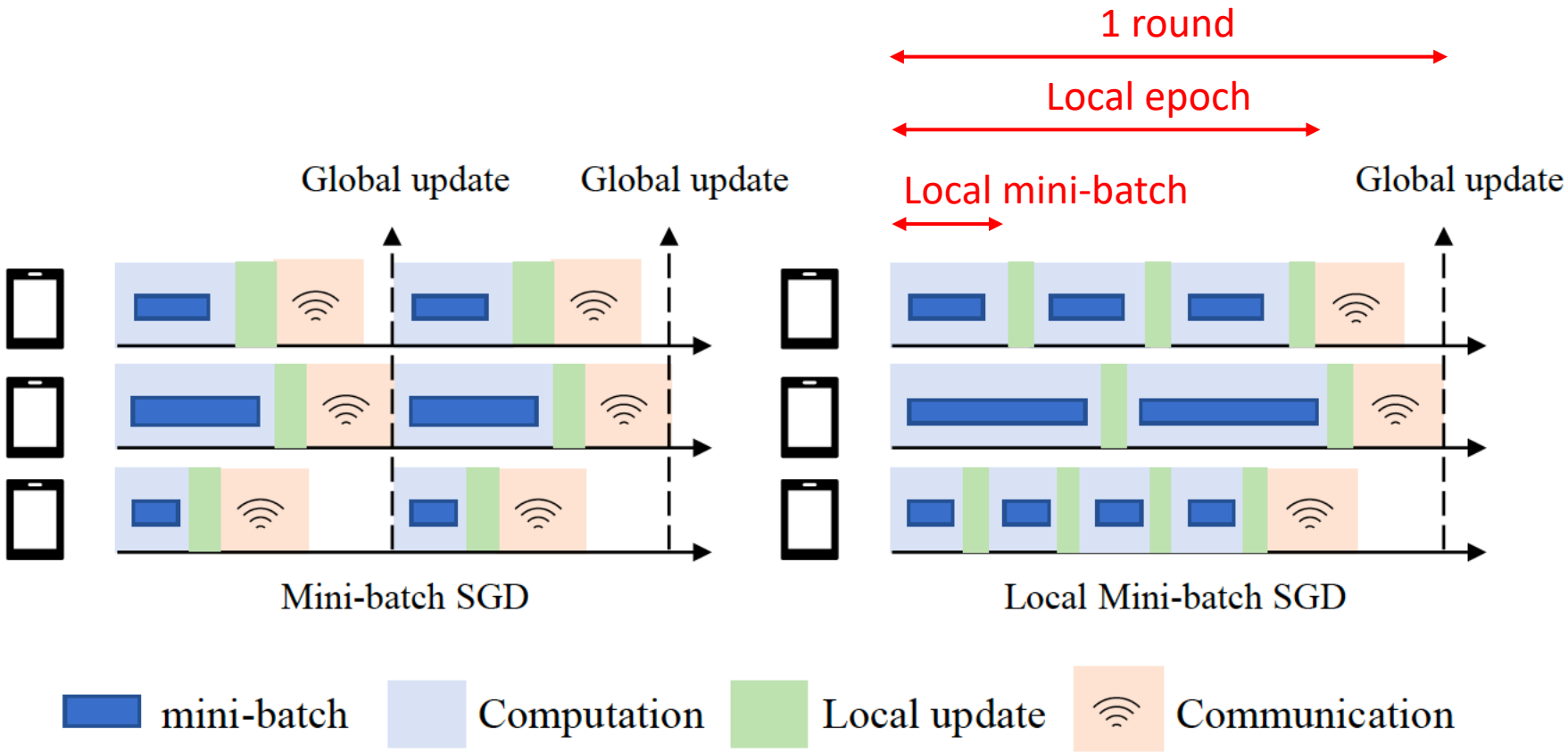
weight의 업데이트 = **에러 낮추는 방향 (decent)** X **한발자국 크기 (learning rate)** X **현 지점의 기울기 (gradient)**

$$-\gamma \nabla F(\mathbf{a^n})$$

$$- \quad \gamma \quad \nabla F(\mathbf{a^n})$$


모든 자료를 다 검토해서
내 위치의 산기울기를 계산해서
갈 방향을 찾겠다.
GD
스텝방향
SGD
전부 다봐야 한걸음은
너무 오래 걸리니까
조금만 보고 빨리 판단한다
같은 시간에 더 많이 간다
Momentum
스텝 계산해서 움직인 후,
아까 내려 오던 관성 방향 또 가자
Nesterov Accelerated Gradient
NAG
일단 관성 방향 먼저 움직이고,
움직인 자리에 스텝을 계산하니
더 빠르더라
Nadam
Adam에 Momentum
대신 NAG를 붙이자.
스텝사이즈
Adagrad
안가본곳은 성큼 빠르게 걸어 훑고
많이 가본 곳은 잘아니까
갈수록 보폭을 줄여 세밀히 탐색
RMSProp
보폭을 줄이는 건 좋은데
이전 맥락 상황봐가며 하자.
Adam
RMSProp + Momentum
방향도 스텝사이즈도 적절하게!
AdaDelta
종종걸음 너무 작아져서
정지하는걸 막아보자.

Local Mini-batch SGD



Federated Averaging (FedAvg)

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:

initialize w_0

for each round $t = 1, 2, \dots$ **do**

$m \leftarrow \max(C \cdot K, 1)$

$S_t \leftarrow$ (random set of m clients)

for each client $k \in S_t$ **in parallel do**

$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$

$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w): *// Run on client k*

$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)

for each local epoch i from 1 to E **do**

for batch $b \in \mathcal{B}$ **do**

$w \leftarrow w - \eta \nabla \ell(w; b)$

return w to server

FedAvg 성능 실험

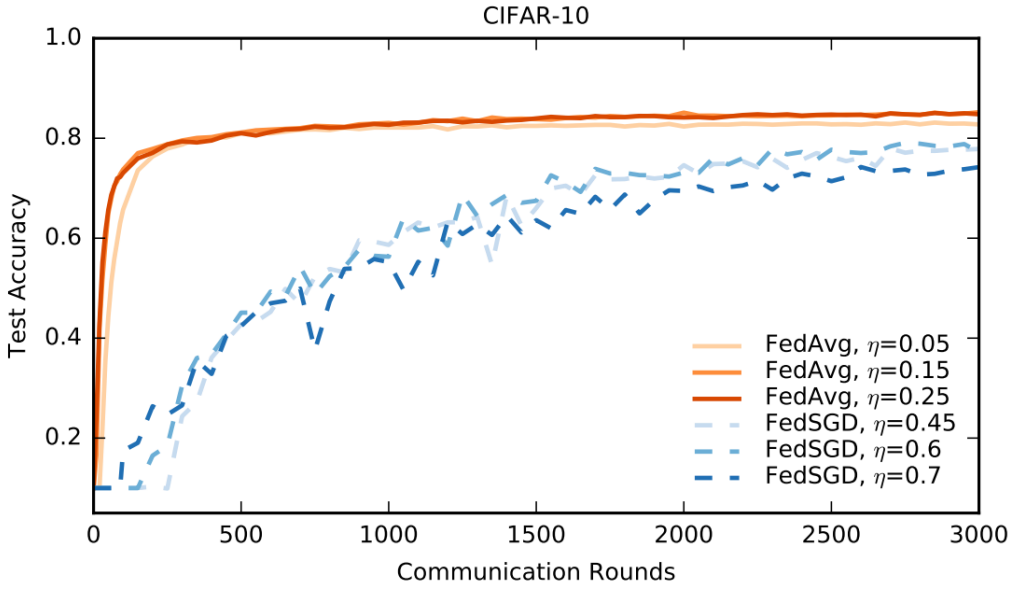


Figure 4: Test accuracy versus communication for the CIFAR10 experiments. `FedSGD` uses a learning-rate decay of 0.9934 per round; `FedAvg` uses $B = 50$, learning-rate decay of 0.99 per round, and $E = 5$.

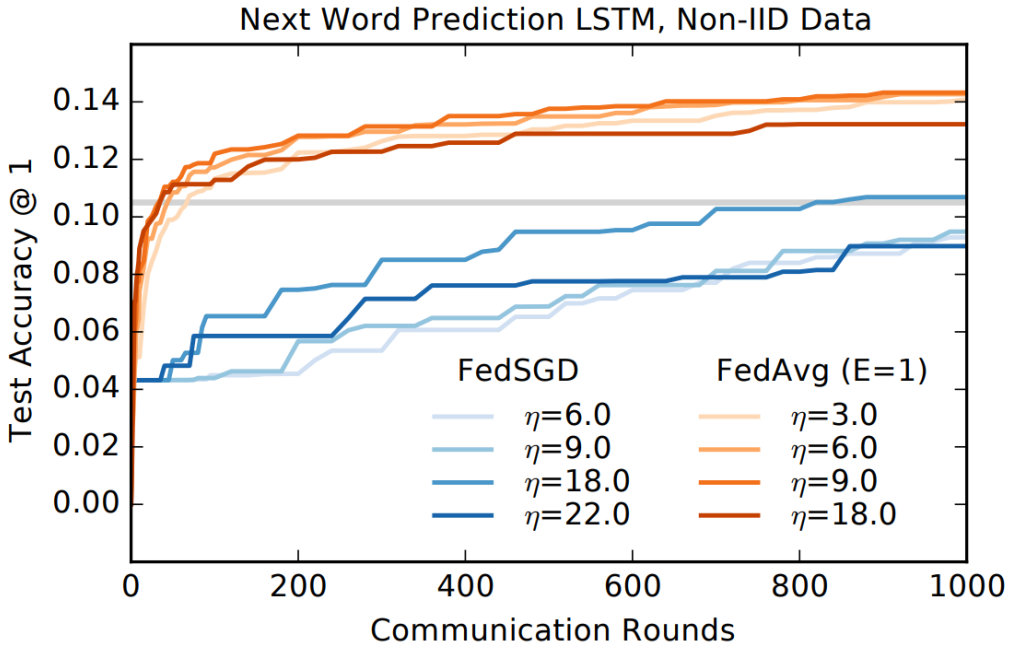


Figure 5: Monotonic learning curves for the large-scale language model word LSTM.

FedAvg 이슈

- System Heterogeneity
  - Synchronous update 방식을 사용하기 때문에 straggler로 인해 학습시간이 느려질 수 있음
  - Related work
    - Asynchronous update 방법 사용시 빠르지만 bounded-delay 이슈로 정확도가 synchronous에 비해 낮게 수렴
    - Device Selection
    - Adaptive mini-batch size
- Communication Overhead
  - Local epoch 도입으로 중앙화 서버와의 통신을 대폭 줄였지만, 디바이스 수가 많아질 수록 Bottleneck 이슈는 피하기 어려움
  - Related work
    - Edge Computing 활용
- Statistical Heterogeneity
  - Non-IID 케이스에서 FedSGD, SGD에 비해 높은 성능을 보였지만 여전히 IID 케이스 보다 낮은 정확도 수렴

Reference

- https://arxiv.org/pdf/1602.05629.pdf