FL WITH MATCHED AVG (2)

FEDERATED LEARNING WITH
MATCHED AVERAGING

REFERENCE

- Wang, Hongyi, et al.
  "Federated learning with matched averaging."
  arXiv preprint arXiv:2002.06440 (2020).

RECALL

RECALL

- FedAvg에서 coordinate-wise averaging
  - 여러 문제가 생길 수 있음
  - 이는 신경망 파라미터의 치환 불변성 때문

RECALL

- 치환 분별성(permutation invariance)
  - 주어진 신경망 파라미터의 순서만 바꿔서 여러 변형을 만들 수 있음
- $\hat{y} = \sigma(xW_1)W_2$ 과 $\hat{y} = \sigma(xW_1\Pi)\Pi^T W_2$

RECALL

- 최적의 가중치가 $\{W_1, W_2\}$ 이라 가정,
- 두 이종적 데이터셋 $X_j, X_{j'}$ 에 대한 훈련된 가중치는
  - $\{W_1\Pi_j, \Pi_j^T W_2\}$ 와 $\{W_1\Pi_{j'}, \Pi_{j'}^T W_2\}$
- 높은 확률로 $\Pi_j \neq \Pi_{j'}$ 이므로
- 어떠한 Π에 대해서도 $(W_1\Pi_j + W_1\Pi_{j'})/2 \neq W_1\Pi$

RECALL

- 의미있는 평균을 위해 치환을 풀어야 함
  - $(W_1 \Pi_j \Pi_j^T + W_1 \Pi_{j'} \Pi_{j'}^T)/2 = W_1$

RECALL

- 확률적(Probability) Federated Neural Matching (PFNM)
  - Averaging 하기 전
  - 클라이언트 신경망의 뉴런들을
  - 매칭함으로써 문제를 다룸

RECALL

- $$\min_{\{\pi_{li}^{j}\}} \sum_{i=1}^{L} \sum_{j,l} \min_{\theta_i} \pi_{li}^{j} c(w_{jl}, \theta_i) \quad s.t. \quad \sum_{i} \pi_{li}^{j} = 1 \forall j, l; \sum_{l} \pi_{li}^{j} = 1 \forall i, j$$
  - 임의의 치환행렬을 적용했을 때
  - 데이터셋 j로 훈련한 신경망의 l번째 뉴런을
  - 글로벌 신경망의 i번째 뉴런과 유사도를 측정해
  - 모든 로컬 데이터셋들과 모든 로컬 뉴런의 조합
  - 모든 글로벌 신경망의 뉴런들에 대해 누계
  - 그 중 가장 작은 값을 갖는 치환행렬을 찾음

RECALL

- 최적화 문제를 어떻게 해결할 것인가?
  - 헝가리안 정합 알고리즘(Hungarian matching algorithm)을 사용
- $\{\theta_i = \arg\min_{\theta_i} \sum_{j \neq j', l} \pi_{li}^{j} c(w_{jl}, \theta_i)\}_{i=1}^{L}$

RECALL

- 유사도 함수 $c(\cdot, \cdot)$를 결정해야 함
- Probabiilistic Federated Neural Matching (PFNM) 연구에서 고려됨
  - Beta-Bernoulli process(BBP)에 기반한
  - Bayesian nonparametric 모델의
  - Maximum a posteriori (MAP) 추정을 계산
- 앞으로 BBP-MAP이라 부름

RECALL

- 기존 PFNM은 단순한 신경망만 상정
  - 가령, 전연결(fully connected) feedforward 네트워크
- PFNM을 CNN과 LSTM에 적용
  - FedAvg 대비 매우 적은 수준의 향상만 있었음

RECALL

- Federated Matched Averaging (FedMA) 방법을 제안
  - 최신 CNN이나 LSTM을 위한
  - 새로운 layer-wise 연합 학습 알고리즘
- 커뮤니케이션 비용을 줄이고, SOTA의 성능을 보임

PERMUTATION INVARIANCE

PERMUTATION INVARIANCE OF KEY ARCHITECTURES

- 치환 불변성을 깊은(deep) 전연결 네트워크의 관점으로 확장
  - 또한, Matched averaging 방식에 대해 고찰
- 이후 LSTMs과 CNN 구조로 확장

PERMUTATION INVARIANCE OF DEEP FCS

- 기존의 식:

$$\hat{y} = \sigma(xW_1\Pi)\Pi^T W_2, \text{ where } \Pi \text{ is any } L \times L \text{ permutation matrix.}$$

PERMUTATION INVARIANCE OF DEEP FCS

- Deep FC 네트워크를 위해 재귀적으로 확장하면

$$x_n = \sigma(x_{n-1}\Pi_{n-1}^T W_n \Pi_n),$$

  - 레이어 인덱스 : $n = 1,...,N$
  - $\sigma(\cdot)$: 임의의 비선형 함수, 마지막 층 제외

PERMUTATION INVARIANCE OF DEEP FCS

- J 클라이언트의 Deep FCs의 matched averaging을 하기 위해
  - 모든 클라이언트의
  - 모든 레이어의 치환을 찾아야 함

PERMUTATION INVARIANCE OF DEEP FCS

- 연결된 레이어의 치환은 서로 영향을 끼쳐
  - NP-hard 결합 최적화 문제를 야기

PERMUTATION INVARIANCE OF DEEP FCS

- 대신에 레이어별로 재귀적으로 생각
  - $\{\Pi_{j,n-1}\}$이 이미 주어졌다고 할 때
  - $\{\Pi_{j,n}\}$을 계산
- $\{\Pi_{j,0}\}$은 모든 j에 대해 I로 할당

PERMUTATION INVARIANCE OF CNNS

- 뉴런 대신에 채널에 대한 불변성을 고려해야 함
- $Conv(x, W)$: 입력 x, 가중치 W에 대한 컨볼루션 연산
  - $W \in \mathbb{R}^{C^{in} \times w \times h \times C^{out}}$

PERMUTATION INVARIANCE OF CNNS

- Deep FCs의 식

$$x_n = \sigma(x_{n-1}\Pi_{n-1}^T W_n \Pi_n),$$

- 처럼 표현하면:

$$x_n = \sigma(\mathbf{Conv}(x_{n-1}, \Pi_{n-1}^T W_n \Pi_n)).$$

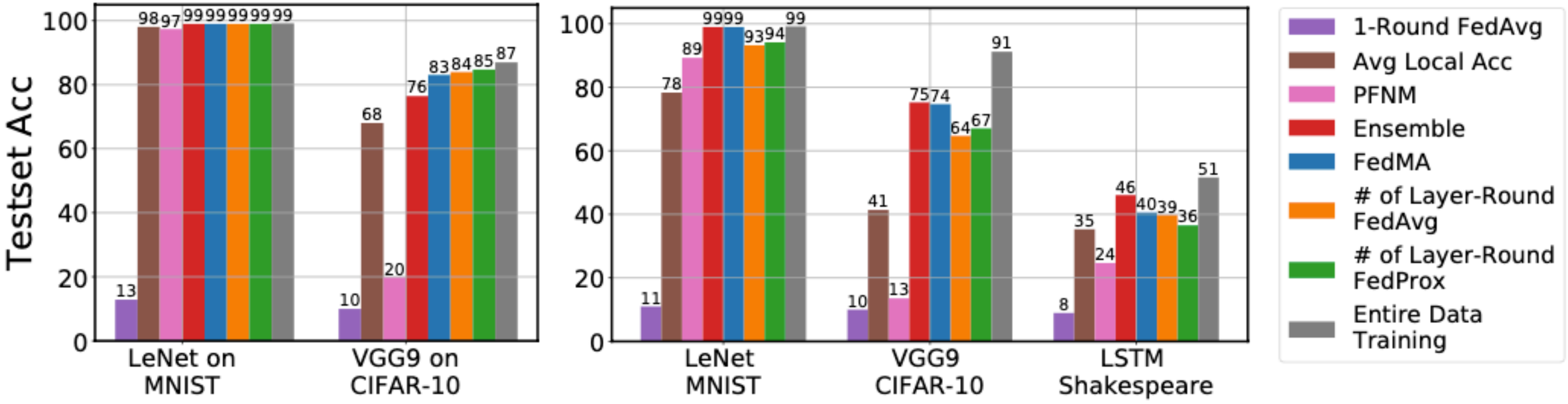
- 유사하게 풀링(pooling) 연산도 포괄 가능

PERMUTATION INVARIANCE OF CNNS

- 마찬가지로 재귀적으로 matched avg에 적용
- 곧장 PFNM을 CNN으로 확장해보니:
  - LeNet과 같은 간단한 CNN 구조에서 잘 동작함 (4 레이어)
  - VGG-9와 같은 복잡한 구조에서는 붕괴됨 (9 레이어)
- 다른 관점이 필요

PERMUTATION INVARIANCE OF CNNS

- 다양한 FL 방법의 비교



PERMUTATION INVARIANCE OF LSTMS

- 순환(Recurrent) 구조의 신경망에서의 치환 불변성
  - 은닉 상태(Hidden state)의 순서와도 연관 있음

PERMUTATION INVARIANCE OF LSTMS

- Input-to-hidden 가중치 W
- Hidden-to-hidden 가중치 H
  - $H \in \mathbb{R}^{L \times L}$
  - L : 은닉 상태의 수

PERMUTATION INVARIANCE OF LSTMS

- 기본 RNN의 은닉 상태에 대해
  - $h_t = \sigma(h_{t-1}H + x_tW)$

PERMUTATION INVARIANCE OF LSTMS

- RNNs를 match하기 위해서는
  - 두 클라이언트의 Hidden-to-hidden 가중치
  - $||\Pi^T H_j \Pi - H_{j'}||_2^2$ 를 최소로 하기 위한
  - Euclidean 유사도 계산이 요구됨
- 이는 quadratic 할당 문제
  - NP-hard 문제

PERMUTATION INVARIANCE OF LSTMS

- 재귀적인 방법(을 조금 바꾸는 것)으로 해결할 수 있음
- 우선 input-to-hidden 가중치인 $\{W_j\}$ 로부터 $\{\Pi_j\}$를 구함
- 연합된 hidden-to-hidden 가중치는 다음과 같이 계산
  - $H = (1/J)\Sigma_j \Pi_j H_h \Pi_j^T$

PERMUTATION INVARIANCE OF LSTMS

- LSTM은 여러 셀(cell) 상태를 가지고 있으며
  - 각각은 개별적인 hidden-to-hidden과
  - input-to-hidden 가중치를 가짐

PERMUTATION INVARIANCE OF LSTMS

- Input-to-hidden 가중치들을
  - $SD \times L$ 가중치 행렬로 누적(stack) 가능
  - S : cell state의 개수
  - D : 입력 차원
  - L : 은닉 상태의 수
- 이후 동일하게 치환을 계산하면 됨

PERMUTATION INVARIANCE OF LSTMS

- LSTM은 종종 임베딩(embedding) 레이어를 가지는데
  - 임베딩: 입력을 밀집 벡터(dense vector)로 벡터화
  - Dense vs. Sparse
- 이는 FC 레이어처럼 취급

FEDMA

FEDERATED MATCHED AVERAGING ALGORITHM

- PFNM을 CNNs와 LSTMs로 확장할 수 있지만
  - 깊은 구조들에서 효과를 보지 못함
  - 반복적인 matched averaging이
  - 전체적으로 좋지 못한 결과를 야기한 것으로 추정

FEDERATED MATCHED AVERAGING ALGORITHM

- Layer-wise한 매칭 스킴을 제안
  - 데이터 센터(서버)가 클라이언트로부터 레이어의 첫 층에 해당하는 가중치를 수집
  - 한 레이어에 해당하는 매칭을 수행
  - 연합 모델의 가중치를 클라이언트들에게 브로드캐스트
  - 클라이언트는 반영 후 연속되는 모든 레이어를 고유 데이터셋으로 학습
    - 이 때 연합 레이어는 동결함
  - 마지막 레이어에 이르기까지 반복

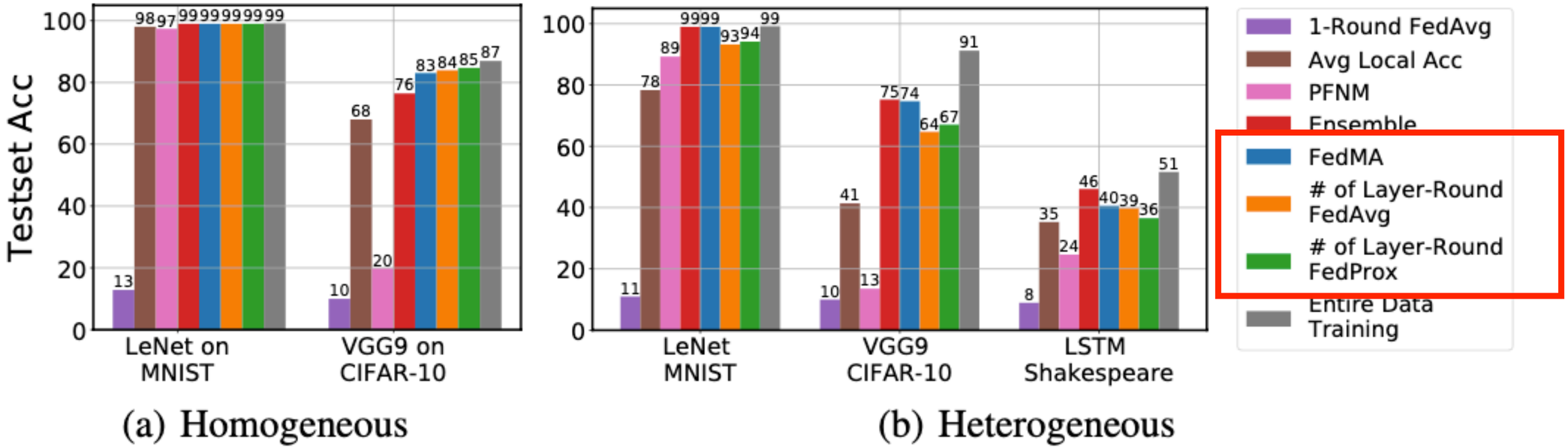
FEDERATED MATCHED AVERAGING ALGORITHM

Algorithm 1: Federated Matched Averaging (FedMA)

Input : local weights of N-layer architectures $\{W_{j,1}, \ldots, W_{j,N}\}_{j=1}^{J}$ from J clients
Output: global weights $\{W_1, \ldots, W_N\}$
$n = 1$;
while $n \leq N$ **do**
  if $n < N$ **then**
    $\{\Pi_j\}_{j=1}^{J} = \text{BBP-MAP}\big(\{W_{j,n}\}_{j=1}^{J}\big)$; // call BBP-MAP to solve Eq. 2
    $W_n = \frac{1}{J} \sum_j W_{j,n} \Pi_j^T$;
  else
    $W_n = \sum_{k=1}^{K} \sum_j p_{jk} W_{jl,n}$ where p_k is fraction of data points with label k on worker j;
  end
  for $j \in \{1, \ldots, J\}$ **do**
    $W_{j,n+1} \leftarrow \Pi_j W_{j,n+1}$; // permutate the next-layer weights
    Train $\{W_{j,n+1}, \ldots, W_{j,L}\}$ with W_n frozen;
  end
  $n = n + 1$;
end

FEDERATED MATCHED AVERAGING ALGORITHM

- FedMA는 네트워크의 레이어 수와 동일한 커뮤니케이션 라운드가 요구됨
  - 동종(Homo)에서 FedAvg와 FedProx와 유사한 성능
  - 이종(Hetero)에서 상회하는 성능



EXPERIMENTS

EXPERIMENTAL SETUP

- VGG 구조에서 배치 정규화는 배제함
  - Future work로 남겨둠

COMMUNICATION EFFICIENCY AND CONVERGENCE RATE

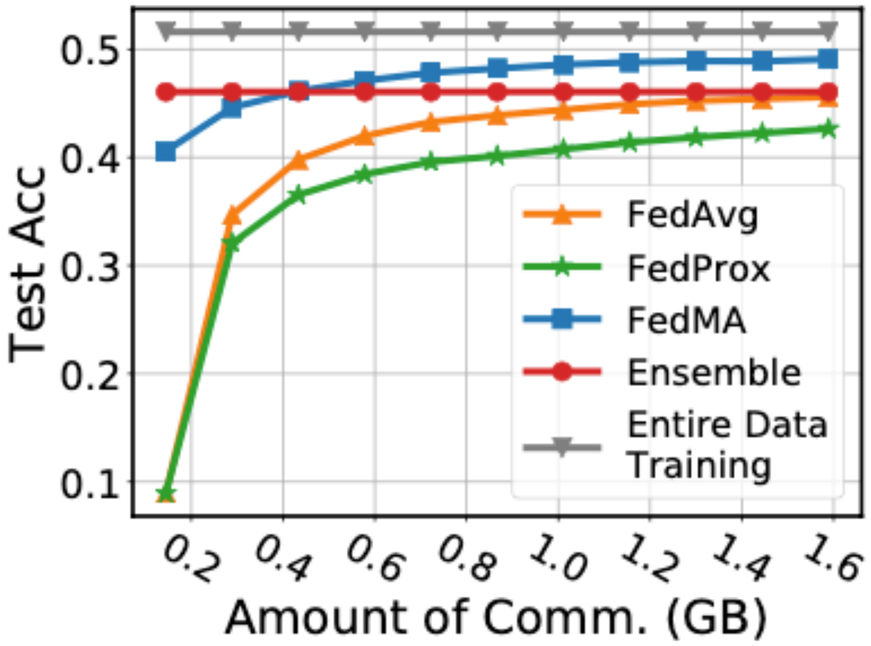
- 데이터 센터와 클라이언트 사이의 비교
  - 주고받는 메시지 크기 (GiB)
  - 글로벌 모델이 좋은 성능을 얻기 위한 커뮤니케이션 라운드 수
  - 앙상블 방법과도 비교

COMMUNICATION EFFICIENCY AND CONVERGENCE RATE

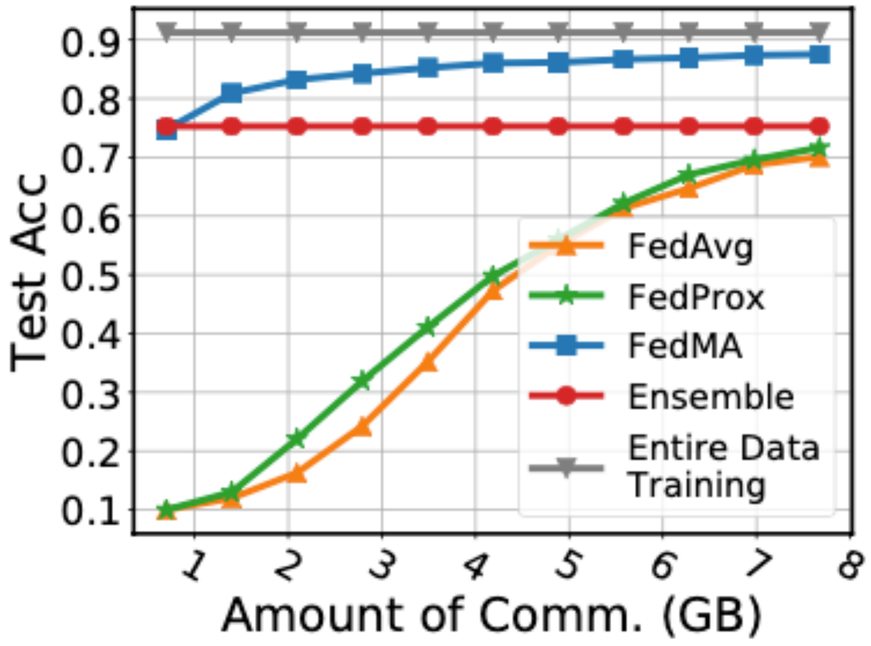
- 이종적 연합 학습 시나리오에서 평가
  - CIFAR-10 / 16 클라이언트 / VGG-9
  - Shakespeare / 66 클라이언트 / 1-layer LSTM

COMMUNICATION EFFICIENCY AND CONVERGENCE RATE

- FedMA가 FedAvg와 FedProx 성능을 상회



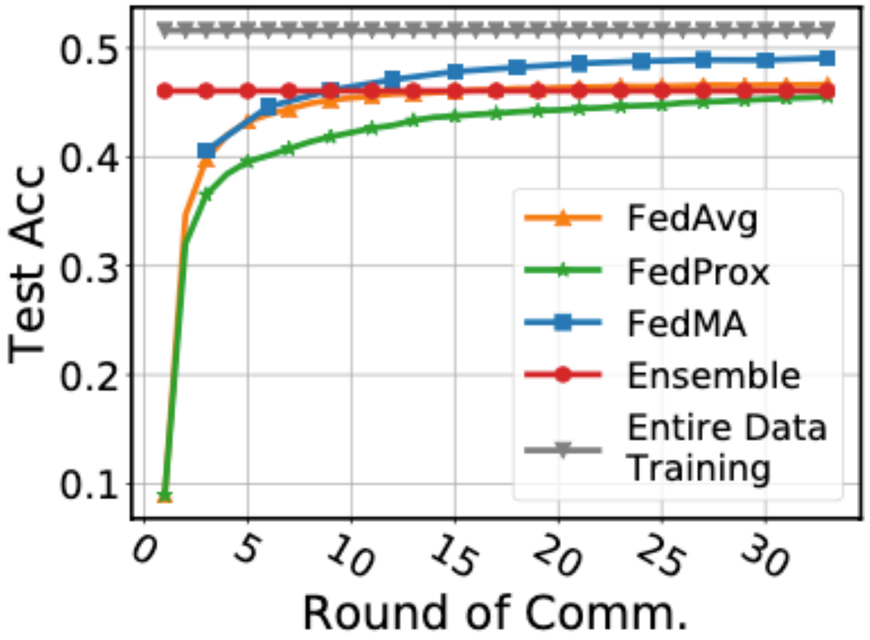
(a) LSTM, Shakespeare; message size



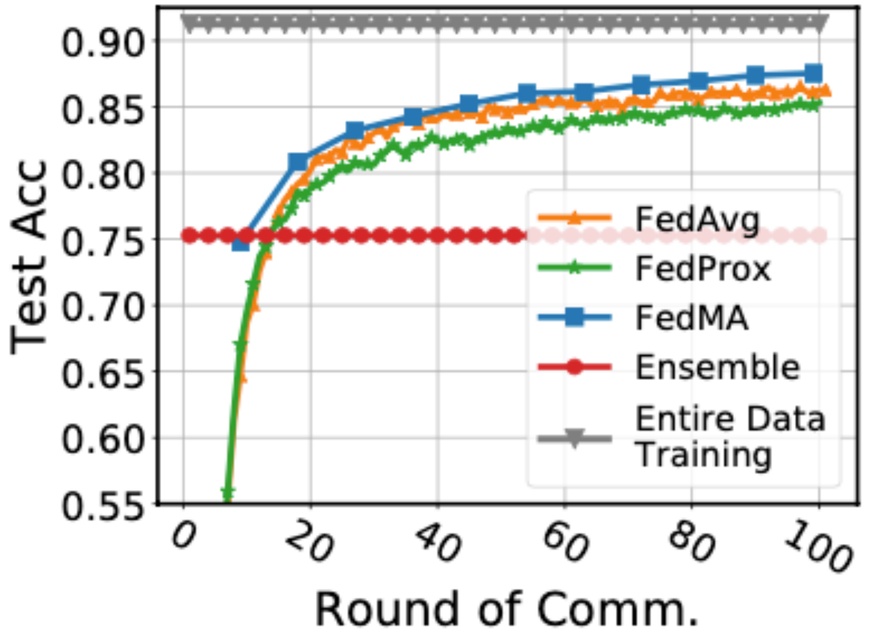
(c) VGG-9, CIFAR-10; message size

COMMUNICATION EFFICIENCY AND CONVERGENCE RATE

- FedMA가 FedAvg와 FedProx 성능을 상회



(b) LSTM, Shakespeare; rounds



(d) VGG-9, CIFAR-10; rounds

EFFECT OF LOCAL TRAINING EPOCHS

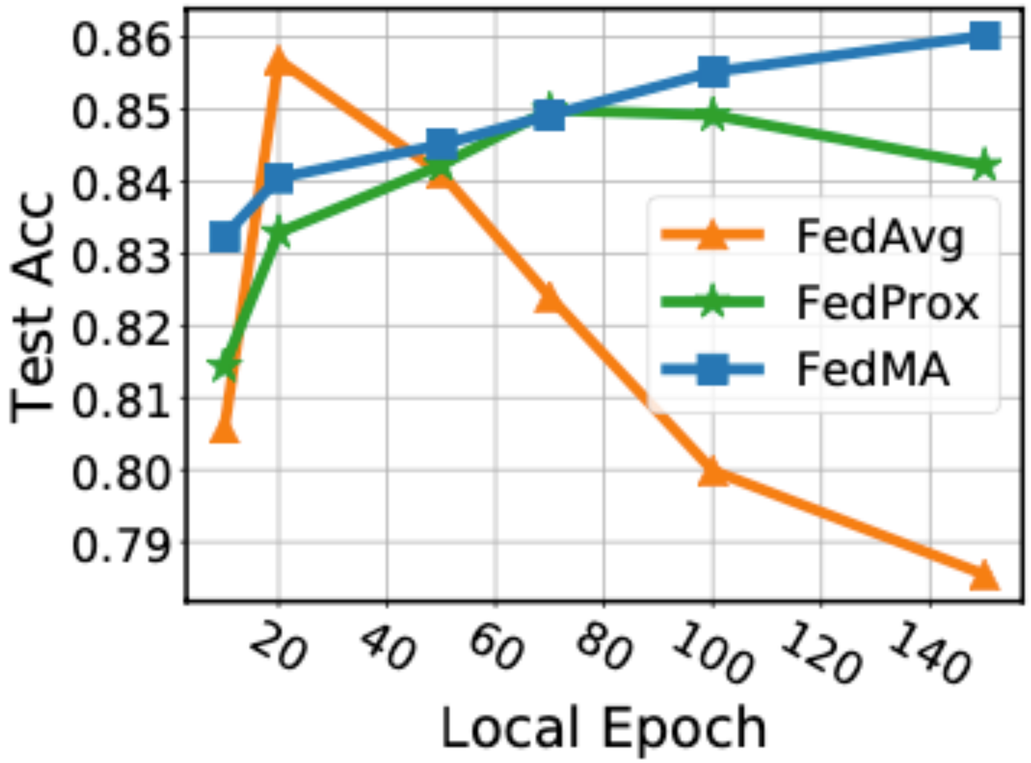
- 사전 연구들에 따르면
- 로컬(local) 학습에서의 에폭 E에 따라
  - FedAvg의 성능에 영향을 끼침은 물론
  - 때로는 발산하는 결과를 야기

EFFECT OF LOCAL TRAINING EPOCHS

- 이종적 세팅에서 FedAvg, FedProx, FedMA 실험
  - CIFAR-10
  - VGG-9
  - $E = \{10,20,50,70,100,150\}$

EFFECT OF LOCAL TRAINING EPOCHS

- 이종적 세팅에서 FedAvg, FedProx, FedMA 실험



EFFECT OF LOCAL TRAINING EPOCHS

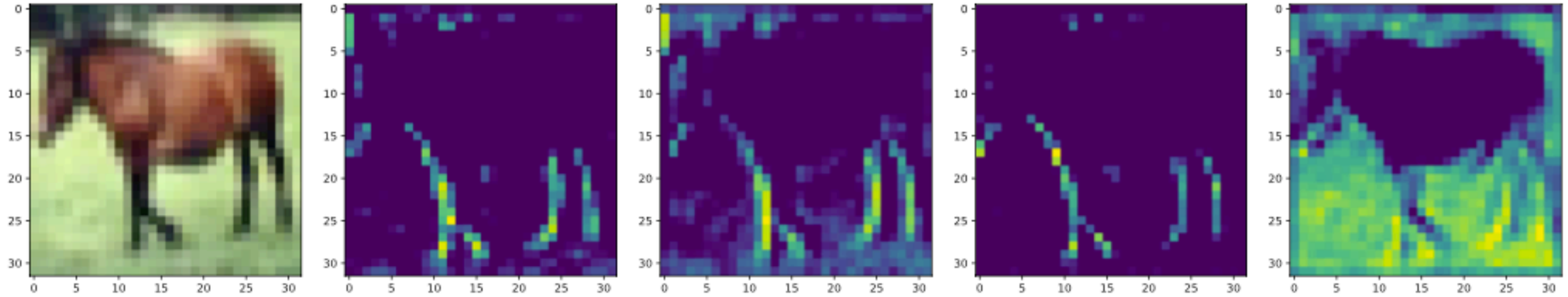
- Local Epoch이 클 수록 FedMA에서 유리
  - 로컬 클라이언트가 원하는 만큼 학습할 수 있는 오직 유일한 방법
- FedAvg는 성능 저하를 야기
- FedProx는 부분적으로만 문제를 경감
  - 더 큰 에폭에서는 성능 저하

INTERPRETABILITY

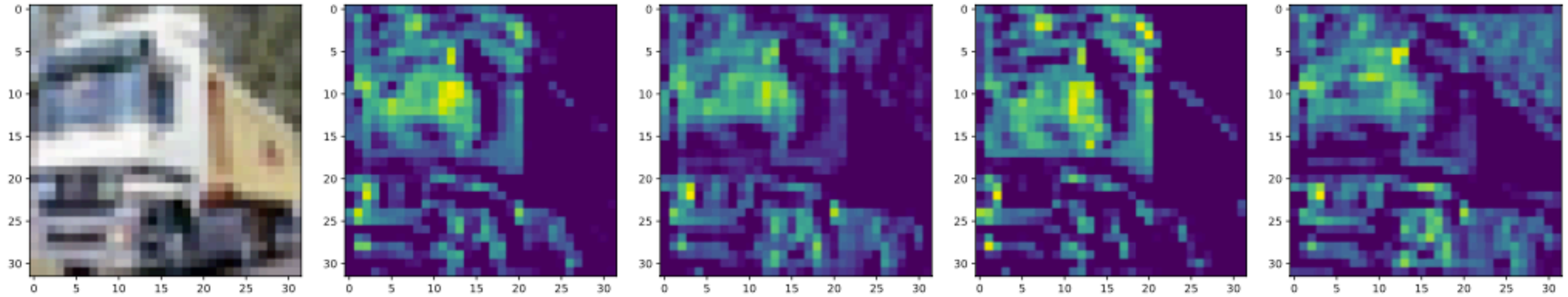
- FedMA의 강점 중 하나는
- FedAvg보다 커뮤니케이션을 효율적으로 행한다는 점
  - Element-wise한 단순한 평균이 아닌,
  - 컨볼루션 필터의 매칭 후 평균

INTERPRETABILITY

- 필터의 매칭을 시각화해서 살펴보면 더 직관적:



(a) Raw input1 (b) FedMA filter 0 (c) Client1, filter 0 (d) Client2, filter 23 (e) FedAvg filter 0



(f) Raw input2 (g) FedMA filter 0 (h) Client 1, filter0 (i) Client 2, filter23 (j) FedAvg filter0

CONCLUSION

CONCLUSION

- FedMA의 소개
  - 치환 불변성과 적응형 글로벌 모델 크기를 다루고자
  - 최신 CNNs나 LSTMs 구조를 위해 설계된
  - Layer-wise한 연합 학습 알고리즘

CONCLUSION

- FedMA가 이전의 FL 알고리즘들 성능을 상회
  - 수렴성 등
- 데이터 편향 문제, 로컬 에폭 수 등을 해결

CONCLUSION

- 향후 연구로 FedMA를 확장해
  - Residual connection이나
  - 배치 정규화 레이어 등에 적용 예정

CONCLUSION

- Fault tolerance한 FedMA
- 더 큰 데이터셋에 대한 성능 연구 등

FL WITH MATCHED AVG (2)

FEDERATED LEARNING WITH
MATCHED AVERAGING