**Asynchronous Online Federated Learning for Edge Devices with Non-IID Data**

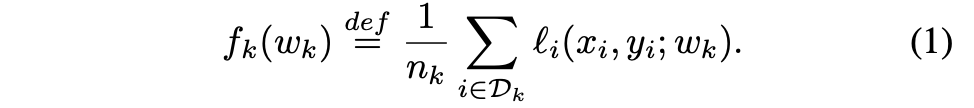
1.

*FedAvg*는 연합학습에서의 non-convex한 모델을 학습하기위한 대표적인 최적화 기법이다. 하지만 *FedAvg* 알고리즘은 homogeneous한 edge device의 가정에 기반한 알고리즘이며, 이는 현실에서 다음과 같은 세가지 이유에서 적합하지 못하다. 첫째, 수집된 데이터의 양과 분포는 학습과정에서 상이하다. 둘째, edge device 또한 latency와 메모리크기와 프로세서 속도와 같은 system configuration에서 상이하다. 셋째, 디바이스의 availability가 보장되지 않은 경우 글로벌 모델 학습에 특정 디바이스가 제외될 수 있다. 본 논문은 edge device가 non-IID한 streaming local data를 활용하여 online learning을 수행할 수 있는 프레임워크를 제안한다. 본 구조에서는 중앙모델이 asynchronous한 방식으로 학습되게 됩니다. 최종적으로 본 논문에서는 모델의 수렴속도와 예측 성능에서 좋은 결과를 보였음을 실험을 통해서 증명한다. Diagram

Description automatically generated

2.

먼저 본 논문에서 제안하는 구조의 소개에 앞서 기존의 *FedAvg*알고리즘을 간략히 소개하겠습니다. K개의 분산된 클라이언트를 가정하였을때, (1)번 수식에서 Dk는 k번째 클라이언트에서 수집된 데이터 집합, nk는 해당 클라이언트에 존재하는 데이터 샘플의 개수이다. 이때 k번째 클라이언트에서 local loss는 (1)번과 같이 정의 된다. 로컬 모델 파라메터인 Wk는 (2)번 수식과 같이 중앙 모델에서 전체 디바이스에 존재하는 데이터 샘플의 개수로 나누어 평균값으로 계산된다. 최종적으로 (3)의 수식과 같이 loss값을 최소화하는 w\*을 찾게 된다.



Text

Description automatically generated with low confidence



3.

기존의 synchronized한 방식에서의 federated optimization은 다음과 같이 수행되었습니다. 앞선 설명 드린 *FedAvg*알고리즘의 경우 각 device는 (1)번 수식의 local objective function의 최적화를 위해 Gradient descent optimization을 수행합니다. 하지만 이경우 로컬 모델의 epoch수가 증가할 수록 각 클라이언트는 자신의 local objective function의 최적점을 찾아가며 이는 central objective와 상이할 가능성이 있습니다. 또한 각 디바이스에서 생성되는 데이터는 local gradient의 변화의 증가로 이어질 수 있습니다. 따라서 본 논문에서는 뒤에서 설명 드리겠지만 local 편차의 양에 제한을 두어 글로벌 모델의 큰 변화를 방지합니다. 하지만 synchronized한 manner에서의 optimization은 앞서 설명한 것과 같이 local device간의 heterogeneity의 특성으로 인하여 컴퓨팅 리소스와 시간의 낭비로 이어진다는 한계점이 존재합니다.

4.

지금부터는 Asynchronous한 online 연합 학습에 기반하고 있는 본 논문에서 제안하는 구조에 대해서 설명하겠습니다. 여기서 주의해야할 점은 각 클라이언트별로 가지고 있는 글로벌 모델의 copy는 다른 클라이언트와 다를 수 있다는 점입니다. 위의 그림은 논문에서 제안하는 ASO-Fed의 전체적인 업데이트 과정을 설명합니다. 중앙 서버는 하나의 클라이언트로부터 업데이트를 수신한 경우 aggregation을 수행하고 client사이의 feature representation을 추출합니다. 글로벌 모델의 학습 후 서버는 next iteration을 시작하고 새로운 모델을 준비된 client들에게 분배합니다. Clients간의 관련성을 포착하기 위해 본 논문의 저자들은 iterative한 local computation을 통한 새로운 모델과 이전 모델 간의 균형을 위해 decay coefficient를 활용하였습니다. 해당 알고리즘은 다음 장에서 알고리즘을 통해 소개해드리도록 하겠습니다. 추가로 그림에서의 client 2의 학습결과를 중앙 서버가 수신한 시점에서는 이미 서버에서는 2번의 모델 학습이 이루어진것을 확인할 수 있는데요 이러한 asynchronous한 manner에서의 inconsistency를 해결하기 위해서 저자들은 local client의 학습과정에서 dynamic step size에 기반하여 global feature representation을 중앙서버에서 학습하여 해결하였다고 언급합니다.

Diagram

Description automatically generated

4.

중앙 서버에서의 학습 과정에 대해서 먼저 설명 드리겠습니다. 각 iteration별로 글로벌 모델은 aggregation을 수행하였는데 이때 다음 (4)번 수식은 t+1번째의 global iteration에서 클라이언트 k로부터 업데이트를 수신한 경우를 나타냅니다. Wt+1로 표현되는 해당시점에서의 새로운 글로벌 모델은 오른쪽 수식에서 보이는 것처럼 이전 t시점에서의 글로벌 모델에서 클라이언트 k의 이전 모델과 현재모델의 차이에 업데이트된 전체 데이터 샘플 중 해당 k번째 client에 존재하는 데이터의 portion을 곱한 값을 빼 줍니다. 즉 위의 수식은 클라이언트 k의 learning rate에 local gradient를 곱한 값을 각 클라이언트가 가진 데이터의 비중으로 곱하여 업데이트를 수행하게 됩니다.

Text, letter

Description automatically generated

5.

앞서 설명 드린 중앙 서버에서의 feature representation의 학습은 다음과 같이 설명됩니다. Asynchronous한 update로 인한 model 성능의 potential effect를 해결하기 위해서, 본 논문에서는 server에서 feature representation learning을 수행합니다. 기존의 많은 연구가 증명하듯이 attention mechanism은 key feature를 잡아내는데 효과적입니다. 따라서 본 저자들 또한 이에 기반하여computational cost를 줄이기 위한 weight normalization를 적용하여 feature learning을 수행한다고 주장합니다. Mobile device의 제한된 컴퓨팅 리소스의 특성상 cnn이나 lstm을 feature extraction layer로 활용한 simple한 network 구조를 채택하여 학습합니다. 밑의 5번과 6번 수식에서 표현된 일련의 과정을 통해 feature extraction layer의 weight값을 업데이트 합니다.

6.

반대로 local model들의 글로벌 모델로부터의 편차를 경감시키기 위한 Local Client의 학습과정은 다음과 같습니다. Local objective function인 fk를 단순히 감소하는 것이 아니라 클라이언트에서 gradient기반 update를 적용하여 7번 수식과 같은 수식을 활용한다. 보시면 regularization term의 형태와 비슷하게 글로벌 모델과 로컬모델간의 차이의 거리가 사용되었음을 확인할 수 있습니다.

7.

학습과정에서 Local client로 지속적으로 유입되는 데이터를 처리하기 위해서는 각 client는 online learning을 수행해야 합니다. 이 과정을 위해서 각 client는 서버로 해당시점에서 마지막으로 업데이트된 최신 모델을 받아 새로운 데이터로 업데이트 합니다. 따라서 현재 모델과 이전모델 사이의 균형을 유지하는 것이 중요합니다. 8번과 9번 수식이 이를 묘사합니다. Global iteration t에서 device k는 최신모델인 wt를 서버로부터 수신합니다. 델타 Sk기호는 local gradients를 의미하는데요 device k에서의 optimization은 다음과 같이 현재 local gradients에서 이전값을 뺀후에 decay coefficient인 베타값을 이용하여 구해진 hk값을 다시 더하여 표현됩니다. 이 hk라는 것이 앞서 설명드린 previous model과 local current model간의 균형을 맞추는 역할을 한다고 저자들은 주장합니다. 오른쪽 알고리즘 2에서 표현된것과 같이 hk는 지속적으로 업데이트 됩니다. 최종적으로 10번의 수식과 같이 클라이언트 k에서의 learning rate에 기반하여 모델의 업데이트 과정이 이루어 지게 됩니다.

8.

Local client에서의 균일한 학습을 위한 또다른 strategy는 dynamic learning step size입니다. Real world에서는 클라이언트간의 각기다른 이유에 근거하여 performance에 차이와 lagging이 생기며 본논문에서는 다음 근거에 기반하여 dynamice learning step size를 활용했다고 주장합니다. 만약 하나의 클라이언트가 더 많은 데이터를 가지고 있거나 bandwidth가 비교적 낮은 경우 중앙모델에 기여하는 정도는 더 낮을 것이며 따라서 learning step size는 커야할 것 입니다. 즉, asynchronous한 환경에서 learning performance의 향상을 위해서 도입된 전략입니다. Step size는 초기에는 모든 클라이언트에 대해 동일한 step size로 초기화 한 후 조정되게 될것입니다. (10)번 수식에서 설명된 update process는 따라서 다음 (11)번 수식과 같이 수정될것입니다. 보시면 다른점은 rtk라는 변수가 learning rate에 곱해진 것을 알 수 있습니다. 이 변수는 과거 t번째 iteration에서 발생한 average time cost와 1사이에 더 큰값을 이용하여서 설정되게 됩니다. 결국 저자들은 이러한 strategy의 채택을 통해 model 수렵의 stragglers의 효과를 감소하는데 도움이 되었음을 실험을 통해서 증명합니다. (long delay를 가진 클라이언트들에게 large step size를 할당함으로써 글로벌 모델로의 기여도의 균형을 유지한다)

9.

본 논문에서는 실험을 위해서 다음 우측 상단에 있는 3가지 real world data set과 마지막 benchmark dataset인 fashion mnist를 사용하였습니다. Real world와 유사한 실험 관경 구축을 위해 이전에 설명드린 decay coefficient는 0.001 learning rate는 0.001로 설정하였습니다. 모델은 single layer LSTM과 fcn 을 통합하여 사용하였고요 mnist의 경우lstm 대신에 cnn layer 2개를 사용하였다고 하였습니다. 세부적인 hyperparameter 조정값들은 논문에 작성되어있습니다. 또한 straggler와drop out 의 simulation을 위해 random offset parameter를 10초부터 100초사이의 서로다른 값으로 실험에 사용하였습니다. 우측 테이블은 baseline approach와 본 논문에서 제안한 ASO-Fed의 예측 성능을 나타냅니다. 확인하시다시피, aso-fed가 regression problem에서 가장 낮은 MAE와 SMAPE를 FITREC와 AIR QUALITY 데이터에서 보여주었으며EXTRASENSORY와 FASHION MNIST데이터에서 또한 전체적으로 가장 높은 성능을 보여주었습니다. 그림 3은 running time에 따른 평가지표의 변화를 보여줍니다. 보시다시피 fedavg와 fedprox 모델의 경우 fluctuation이 큰폭으로 발생하였으며 이는 synchrnonus한 프레임 워크가 skewed하고 non-IID한 스트리밍 데이터에 적합하지 않음을 나타 냅니다. 역시 이 실험에서도 aso-fed가 running time에 따라서 가장 steady한 improvements를 보였습니다. 테이블 5에서는 central feature learning과 dynamice learning step size의 효과로 인하여 학습의 effecitveness를 늘리고 computation cost이 감소되었음을 나타냅니다.

10.

Time efficiency

Client의 drop out rate에 따른 성능 비교

전체적인 performance 그래프