

HeteroSplitFed : 클라이언트의 서로 다른 학습능력 및 통신환경을 고려한 분산학습 기법 연구

류지현, 양희철

충남대학교 컴퓨터공학과, 충남대학교 컴퓨터융합학부

ryujihyunmail@gmail.com, hcayang@cnu.ac.kr

HeteroSplitFed : A Study on distributed learning techniques for heterogeneous client settings on computing capability and communication bandwidth

Ji Hyun Ryu, Heecheol Yang

Department of Computer Science Engineering, Chungnam National Univ.

요약

본 논문에서는 클라이언트가 서로 다른 학습능력 및 통신환경을 가지는 분산학습 환경을 고려한다. 컴퓨팅 능력이 제한되거나 통신환경이 열악한 클라이언트가 학습에 참여할 수 있도록 SplitFed와 ordered dropout 방식을 결합한 새로운 알고리즘을 제안한다. 제안한 방식을 통해 클라이언트가 서로 다른 크기의 모델을 학습함으로써 모두 분산학습에 참여할 수 있고 전체 모델 학습에 기여할 수 있도록 하였다. CIFAR10 데이터셋에 대한 이미지 분류를 통해 클라이언트가 서로 다른 크기의 모델을 학습하더라도 모두 전체 모델 성능 향상에 기여할 수 있음을 밝혔다.

I. 서론

연합학습은 분산학습의 대표적인 학습 방법이다. 여러 클라이언트와 서버가 학습 과정에서 학습모델의 파라미터를 전달하는 방식으로 학습을 진행한다[1]. 이를 통해서 각 클라이언트는 데이터의 유출 없이 학습할 수 있는 장점이 존재한다. 하지만 컴퓨팅 능력이 제한되거나 통신환경이 열악한 클라이언트에서는 학습 수행이 어렵고 이러한 이유로 전체 학습결과가 저하되는 결과를 얻게 된다.

분할학습은 연합학습과 다른 분산학습 방식으로 컴퓨팅 능력이 제한된 클라이언트를 활용하여 학습시키기 위해서 제안된 방식이다[2]. 연합학습과 달리 전체 학습모델을 클라이언트 모델과 서버 모델 두 부분으로 나누어 각각을 클라이언트와 서버에서 학습하게 된다. 전체 모델의 일부분만을 클라이언트에서 학습을 진행하기 때문에 컴퓨팅 능력이 제한된 클라이언트에서도 학습이 가능한 것이 장점이지만, 전체 학습 진행 과정이 순차적으로 진행되어야 하므로 학습 진행시간이 오래 걸리는 단점이 존재한다.

두 분산학습과정의 단점을 극복하기 위해서 새로운 학습 방식인 SplitFed 방식이 제안되었다[3]. SplitFed 방식에서는 분할학습을 위한 서버와 연합학습을 위한 서버가 존재하여 두 학습 방식을 활용하며 학습이 진행되며 이를 통해서 클라이언트의 제한된 학습능력과 열악한 통신환경을 극복할 수 있고, 분할학습의 느린 학습 속도 또한 개선이 가능한 장점이 있다.

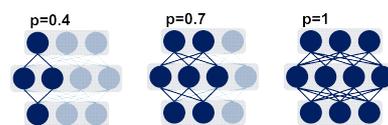
기존 연구들에서도 제안된 학습능력과 열악한 통신환경에서의 학습을 진행하고 성능을 높이기 위해서 다양한 방식이 제안되었다[4],[5],[6].

본 연구는 SplitFed 방식과 Ordered Dropout[7]을 활용하여 각 클라이언트의 학습능력과 통신환경이 다른 경우 학습이 가능한 새로운 알고리즘을 제안한다. CIFAR10 데이터셋과 ResNet18, ResNet50을 사용하여 테스트를 진행하였고 성능을 확인하였다.

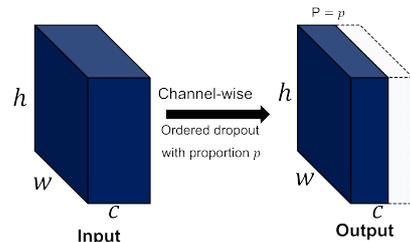
II. 본론

1. Ordered Dropout(OD)

OD에서는 노드를 랜덤하게 드롭아웃 하는 것이 아닌 [그림 1]에서와 같이 오른쪽에 있는 노드부터 순차적으로 드롭아웃을 수행한다. 이때 드롭아웃 비율인 p 는 드롭아웃에서 남아있는 비율을 나타낸다. 이를 이용해서 [그림 2]와 같이 모델의 컨볼루션 레이어에서 각 텐서에서 드롭아웃 비율 p 에 대해서 channel-wise한 방식으로 드롭아웃을 구현하여 진행하였다.



[그림 1] p 값에 따른 OD

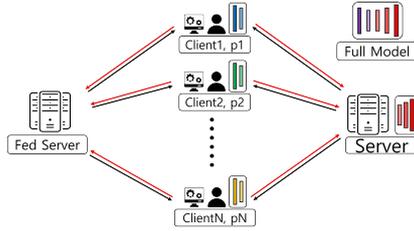


[그림 2] Channel-wise OD with p

2. 시스템 모델

본 논문에서 사용된 시스템 모델은 [그림 3]과 같다. 기존 SplitFed에서 제안된 모델에서 각 클라이언트에서 서로 다른 OD 비율을 적용하여 학습을 진행한다. 먼저 클라이언트마다 다른 OD 비율 p_n 에 따라서 클라이언트 모델을 학습한다. 이후 모델의 나머지 부분을 서버에서 학습을 진행한다. 이제 서버에서부터 각 클라이언트로 역전파를 진행한다. 이후 Fed Server로 학습이 완료된 클라이언트 모델 파라미터를 전달한다. Fed

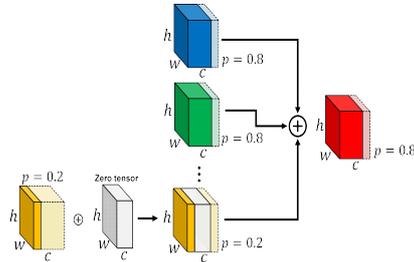
Server에서는 FedAvg 방식으로 모델 aggregation을 진행한다. aggregation이 완료된 클라이언트 모델을 클라이언트에 다시 분배하는 과정으로 한 번의 Round가 종료된다.



[그림3] SplitFed system model with Ordered dropout

3. 알고리즘

[그림 4]에서는 각 클라이언트의 OD 비율이 다른 경우 클라이언트의 학습결과를 aggregation 하는 과정을 보여준다. OD 비율이 가장 큰 클라이언트 모델의 크기에 맞춰서 aggregation 하게 된다. OD 비율이 작은 모델의 경우 각 layer에서의 결과 텐서에 0값으로 채운 텐서를 연결하여 텐서 크기를 일치시킨다. 이후 FedAvg를 사용하여 그 결과들을 aggregation 하게 된다. 서버에서 클라이언트로 결과 분배는 다시 클라이언트 모델의 크기에 맞춰서 슬라이싱하여 분배한다.



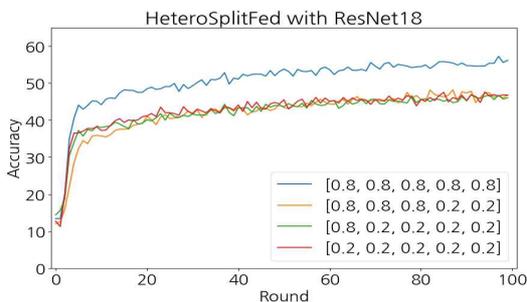
[그림4] HeteroSplitFed에서의 Aggregation 과정

4. 시뮬레이션 결과

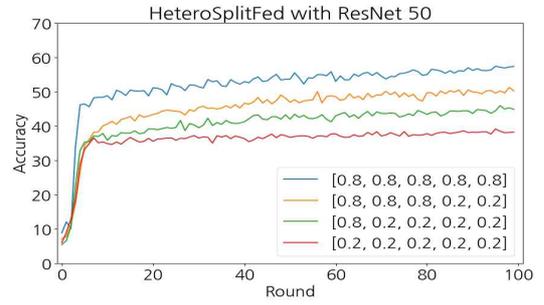
실험은 ResNet18과 ResNet50 모델을 기반으로 SplitFed 모델을 구성하였다. 학습 Round는 100, local epoch은 5로 설정하였고, 클라이언트는 5개로 구성하였다. 학습 데이터셋은 CIFAR10을 사용하여 Dirichlet 분포를 이용하여 5개의 클라이언트에 데이터셋을 분배하였다. 이때 Dirichlet 분포에서 파라미터 α 값을 0.9로 설정하였다. 데이터의 분포는 non-i.i.d.하게 분배된다. [표 1]은 클라이언트마다 p 에 대한 최고 정확도를 정리하였다.

p	ResNet18	ResNet50
0.8, 0.8, 0.8, 0.8, 0.8	57.30%	57.40%
0.8, 0.8, 0.8, 0.2, 0.2	48.24%	51.22%
0.8, 0.2, 0.2, 0.2, 0.2	46.93%	45.91%
0.2, 0.2, 0.2, 0.2, 0.2	47.64%	39.10%

[표 1] p 값의 따른 각 모델 학습 정확도



[그림 5] HeteroSplitFed with ResNet18



[그림 6] HeteroSplitFed with ResNet50

[그림 5], [그림 6]에서처럼 학습의 정확도가 Round가 진행될수록 점차 증가하는 모습을 보여준다. ResNet18을 사용한 결과에서는 OD 비율이 높은 경우 빠르게 증가하면서 수렴하는 결과를 보이지만 나머지 결과에 대해서는 큰 차이가 없지만 수렴하는 결과를 보인다. ResNet50을 사용한 경우에는 OD 비율이 작은 경우 낮은 정확도와 수렴속도를 보이지만 OD 비율이 높은 클라이언트가 포함된 경우에는 정확도가 OD 비율이 낮은 경우보다 더 빠르게 수렴하는 결과를 확인하였다.

III. 결론

본 논문에서는 분산학습에 참여하는 클라이언트의 학습능력 및 통신환경이 제한되는 상황에서 학습 진행이 어려운 한계점을 해결하기 위해 SplitFed와 OD를 활용한 분산학습 알고리즘을 제안하였다. 향후 본 논문에서 제안한 알고리즘을 바탕으로 클라이언트와 서버 모델을 적응적 방식으로 나누는 방식과 더 다양한 상황에서의 시뮬레이션 환경을 구축하여 실험 및 평가를 진행할 계획이다.

참고 문헌

- [1] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. "Communication-efficient learning of deep networks from decentralized data." AISTATS, pp. 1273-1282. PMLR, 2017.
- [2] Vepakomma, Praneeth, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. "Split learning for health: Distributed deep learning without sharing raw patient data." arXiv preprint arXiv:1812.00564 2018. (<https://arxiv.org/abs/1812.00564>)
- [3] Thapa, C., Arachchige, P. C. M., Camtepe, S., & Sun, L. "Splitfed: When federated learning meets split learning." AAAI. Vol. 36. No. 8. 2022.
- [4] Chen, H., Chen, X., Peng, L., & Bai, Y. "Personalized Fair Split Learning for Resource-Constrained Internet of Things." Sensors, 24(1), 88. 2023
- [5] Shin, J., Ahn, J., Kang, H., & Kang, J. "FedSplitX: Federated Split Learning Accuracy for Computationally-Constrained Heterogeneous Clients." arXiv preprint arXiv:2310.14579. 2023 (<https://arxiv.org/abs/2310.14579>)
- [6] Kang, H., Cha, S., Shin, J., Lee, J., & Kang, J. "NeFL: Nested Federated Learning for Heterogeneous Clients." arXiv preprint arXiv:2308.07761. 2023 (<https://arxiv.org/abs/2308.07761>)
- [7] Horvath, S., Laskaridis, S., Almeida, M., Leontiadis, I., Venieris, S., & Lane, N. "Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout." NeurIPS 34, pp. 12876-12889. 2021