통신 효율적인 연합 학습 기법에 관한 연구 동향

서상원, 이재욱, 고한얼, 백상헌
고려대학교

{sw_seo, iioiioiio123, heko, shpack}@korea.ac.kr

A Study on the Communication-Efficient Federated Learning

Sangwon Seo, Jaewook Lee, Haneul Ko, and Sangheon Pack
Korea Univ.

요 약

연합학습에서는 서버와 클라이언트가 학습할 심층 모델을 교환하면서 학습을 진행한다. 하지만 최근 심층 모델의 크기가 점차 커짐에 따라 통신 부하가 급격히 증가하는 문제가 발생하였고, 이러한 문제를 해결하고자 통신 효율적인 연합 학습에 대한 연구가 활발히 진행되고 있다. 이에 따라 본 논문에서는 통신 효율적인 연합 학습 기법들에 대해 소개하고 해당 기법들의 한계점을 분석한다.

Ⅰ. 서 론

최근 모바일 기기 등의 컴퓨팅 능력이 발달함에 따라 연합학습에 대한 연구가 각광받고 있다. 연합 학습은 클라이언트가 가지고 있는 개인 정보가 담긴 데이터를 서버로 전송해야 하는 기존의 머신 러닝 기술과 달리 클라이언트에서 직접 학습하고, 모델을 전송한다 [1]. 이러한 특징으로 데이터의 프라이버시를 보존하며, 학습하기 위한 컴퓨팅 부하를 줄일 수 있는 장점이 존재한다.

그림 1은 연합 학습 구조와 절차를 나타낸다. 연합 학습을 수행하기 위해서는 연합 학습 서버와 스마트폰, 태블릿 PC, 스마트 워치 등과 같은 컴퓨팅이 가능한 모바일 기기와 같은 클라이언트들이 필요하다. 연합 학습은 그림 1 과 같이 크게 3 가지의 단계로 수행된다 [2]. 우선, 1 단계 (Task Initialization 단계)에서는 서버가 학습에 참여할 클라이언트, 학습할 데이터의 종류, 학습할 모델의 형태 등에 대해서 정하고, 브로드캐스팅을 통해 연합 학습에 참여할 클라이언트들에게 학습할 심층 모델을 전송한다. 2 단계 (Local Model Training and Update 단계)에서는 학습에 참여하는 클라이언트들이 서버로부터 전송 받은 심층 모델을 자신의 데이터를 통해 학습을 수행하고, 학습이 완료된 모델 (Local Model)을 서버로 업로드 한다. 그 후, 3 단계 (Global Model Aggregate and Update 단계) 에서는 서버가 모든 클라이언트들로부터 전송 받은 모델을 수집하고, 수집된 모델들을 통해 다음에 학습할 심층 모델 (Global Model)을 만든다. 연합학습에서는 1 단계부터 3 단계의 과정을 진행하는 것을 라운드라고 정의하고, 해당 라운드는 원하는 성능이 될 때까지 반복적으로 수행된다.

한편, 연합학습에서는 원하는 성능의 심층 모델을 학습시키기 위해 많은 라운드가 필요하다. 더불어, 최근 심층 모델의 크기가 커짐에 따라 (e.g., VGG19 의 경우

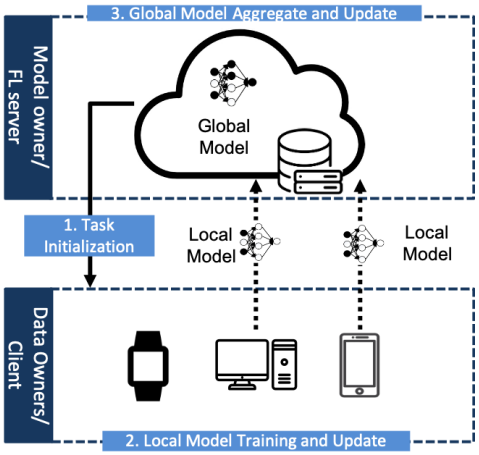

그림 1. 연합 학습의 구조

548 Mbyte [3]), 한 라운드의 통신 부하가 증가하는 문제가 대두되고 있다. 또한, 통신 부하의 증가로 서버와 클라이언트간의 모델 교환 시간이 증가되며, 이는 곧 학습 시간을 증가하는 문제를 야기시킨다. 따라서, 최근 해당 문제를 해결하기위한 통신 효율적인 연합 학습 기법들이 제안되었다.

따라서 본 논문에서는 통신 효율적인 연합 학습 기법들을 II 장에서 소개하고, III 장에서 해당 기법들의 한계점을 제시하면서 결론을 맺는다.

Ⅱ. 통신 효율적 연합 학습

연합 학습에서 통신 비용을 줄이기 위한 방법으로 Quantization, Sparsification, Temporally Update 등의 방법을 제안했다.

[4]에서는 앞에서 언급한 서버와 클라이언트 사이에 모델을 교환할 때 높은 통신 비용이 소요된다는 점을

해결하기 위해 CE-FedAVG 라는 기법을 제안했다. CE-FedAVG 는 서버에서 클라이언트로 모델을 전송할 때는 기존 연합 학습의 방법과 같이 압축을 하지 않고 전송하지만, 클라이언트에서 학습을 마친 이후 서버로 전송할 때 압축하는 방법을 이용한다. 이 과정은 먼저 클라이언트에서 서버로부터 지시 받은 압축 비율에 따라 학습된 모델의 파라미터 중 일부를 선정하는 Sparsification 과정을 진행한다. 그 뒤, 선정된 파라미터는 32bit float 값에서 8bit int 값으로 변환하는 Quantization 과정을 통해 다시 압축된다. 또한, 선정된 모델의 파라미터 인덱스 값을 Golomb Encoding 을 통해 압축하여 서버로 전송한다. 이후 서버는 각 클라이언트로부터 전송 받은 압축된 모델의 파라미터와 파라미터의 인덱스 값을 디코딩하여 Aggregation 한다. CE-FedAVG 방법을 사용하면 기존의 연합 학습에서 사용하는 FedAVG 보다 최대 6 배 더 빠르게 수렴하고, 클라이언트에서 서버로 업로드하는 데이터의 양은 3 배 더 적은 결과를 보인다. 하지만, 서버에서 클라이언트로 모델을 전송하는 과정은 기존 FedAVG 와 같기 때문에 통신 오버헤드가 존재한다.

[5]에서는 기존의 연합 학습에서 사용한 동기식 모델 업데이트 방법이 아닌 비동기 방식을 사용한 TW-FedAVG 방법을 제안했다. TW-FedAVG 는 크게 두가지의 알고리즘으로 구성되어있다. 먼저, Layerwise Model Update 는 모델의 학습을 진행할 때 깊은 계층이 얕은 계층보다 값이 변경되는 빈도가 적다는 점에 착안하여 클라이언트에서 학습한 얕은 계층과 깊은 계층의 업데이트 횟수를 다르게 진행한다. 두번째로, 비동기 업데이트 방식인 Temporally Weighted Update 방법을 사용한다. 기존의 동기식 방식에서는 서버가 모든 클라이언트의 모델을 전송 받았을 때 Aggregation 을 진행한다. 하지만, 비동기 방식에서는 일부 클라이언트의 모델만 전송 받아도 서버에서 Aggregation 을 진행하기 때문에, 자주 모델을 업데이트하는 클라이언트가 존재하면 서버의 모델이 수렴이 늦어지거나 정확도가 떨어지는 등의 문제점이 존재한다. Temporally Weighted Update 방식은 모델을 업데이트 하지 않은 기존의 클라이언트의 모델을 서버에서 저장해 두고, 이를 Aggregation 할 때마다 다시 사용하는 방식이다. 또한, Aging Factor 를 두어 라운드가 진행될 때 업데이트 하지 않은 클라이언트의 모델의 비중을 줄이게 된다. 하지만, 서버에서 모든 클라이언트의 모델을 저장해야 하는 오버헤드가 존재한다.

[6]에서는 서버가 매 라운드마다 학습된 모델이 기존 모델에서 얼마나 변할 지 예측을 진행하고 이 값을 모델과 함께 클라이언트에게 전송한다. 클라이언트는 학습한 이후 모델의 파라미터 값을 서버로부터 전송 받은 학습하기 이전 모델의 파라미터 값과 비교하여 변화량을 계산한다. 이후, 계산한 변화량과 서버로부터 전송 받은 임계값을 비교했을 때, 임계값보다 작으면 학습된 모델을 전송하지 않고, 임계값보다 크면 학습된 모델을 전송한다. 하지만, 이 방법의 경우, 모델의 변화량이 크더라도 정확도가 향상되지 않거나, 하락할 수 있다는 단점도 존재한다.

[7]에서는 클라이언트에서 서버로 모델을 전송할 때, 모델의 파라미터를 일렬로 만들고, 10 진수 값을 ASCII 코드로 변환하는 Polyline Encoding 과정을 거친다. 그 뒤, 압축된 모델과 모델을 일렬로 만들기 전의 형태에 대한 정보를 서버로 전송하고, 서버에서는 다시 이 값을 디코딩하여 Aggregation 한다. 또한, 낙오자 문제를 해결하고 빠른 수렴을 하기 위해 각 클라이언트를 컴퓨팅 속도 별로 등급을 나누고, 같은 등급의 클라이언트끼리 동기식 업데이트를 진행하며, 각 등급은 비동기 업데이트 방식을 사용한다. 이 방법의 경우, 기존 기법 대비 최대 6.41 배 빠르게 수렴하고, FedAsync 대비 8.5 배의 통신을 줄일 수 있지만, FedAVG 와 비교했을 때, 최대 0.1 배의 통신이 감소한 근소한 차이를 보인다.

Ⅲ. 결론

본 논문에서는 연합 학습에서 서버와 클라이언트 사이에 모델을 전송할 때 통신 오버헤드를 줄이는 방법인 통신 효율적 연합 학습에 대한 연구 동향에 대해 살펴보았다. 살펴본 기법들이 기존 FedAVG 대비 클라이언트에서 서버로 모델을 전송할 때의 통신 비용이 줄어들었지만, 서버에서 클라이언트로 전송하는 과정에서 통신 오버헤드, 서버에 저장 공간을 두어야 하는 문제 등 개선해야 할 문제점이 여전히 존재한다. 향후에는 이러한 문제점을 해결하는 기법을 통해 연합학습에서의 통신 등 여러 오버헤드를 더욱 효과적으로 줄일 수 있을 것이라 전망한다.

ACKNOWLEDGMENT

본 연구는 방위사업청과 국방과학연구소가 지원하는 미래전투체계 네트워크기술 특화연구센터 사업의 일환으로 수행되었습니다. (UD190033ED)

참 고 문 헌

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. Conf. Machine Learning Research*, Fort Lauderdale, FL, USA, April 2017.

[2] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031-2063, April 2020.

[3] A. Sapio, M. Canini, C. Ho, J. Nelson, P. Kalnis, C. Kim, A. Krishnamurthy, M. Moshref, D. Ports and P. Richtark, “Scaling Distributed Machine Learning with In-Network Aggregation”, *arXiv*, February 2019.

[4] J. Mills, J. Hu, and G. Min, "Communication-Efficient Federated Learning for Wireless Edge Intelligence in IoT," *IEEE Internet of Things Journal,* vol. 7, no. 7, pp. 5986-5994, July 2020.

[5] Y. Chen, X. Sun, and Y. Jin, "Communication-Efficient Federated Deep Learning With Layerwise Asynchronous Model Update and Temporally Weighted Aggregation," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 31, no. 10, pp. 4229-4238, October 2020.

[6] M. Ribero, and H. Vikalo, “Communication-Efficient Federated Learning via Optimal Client Sampling,” *ArXiv,* pp. 1-27, October 2020.

[7] H. Chai, Y. Chen, L. Zhao, Y. Cheng, and H. Rangwala, “FedAT: A Communication-Efficient Federated Learning Method with Asynchronous Tiers under Non-IID Data,” *ArXiv,* pp. 1-16, October 2020.