연합학습에서의 활성화 함수에 대한 연구

신재우 ○ 김동규 김태현 오재훈 윤세영

KAIST 김재철 AI 대학원

(yimsungen5, eaststar, potter32, jhoon.oh, yunseyoung)@kaist.ac.kr

An Empirical Study of the Activation Function in Federated Learning

Jaewoo Shin○ Donggyu Kim Taehyeon Kim Jaehoon Oh Seyoung Yun

KAIST AI

요 약

연합학습(Federated Learning; FL)은 대규모로 분산되어 있는 장치들로부터 모델을 학습하는 기계 학습 패러다임 중 하나이다. 최근 FL 연구의 경향 및 발전은 최적화 방법을 변경하여 학습 절차를 개선하는 방향으로 이루어져왔다. 그러나 기존 문헌들에서 간과되고 있는 사실은 중앙 집중식 접근 방식에서 사용되는 모델 구조 및 활성화 함수 등을 검증없이 연합학습 접근 방식에 적용하고 있다는 점이다. 본 논문에서는 이 문제를 지적하고 연합학습 환경에서 활성화 함수의 변화에 따른 성능 변화와 다양한 분석 결과를 제공한다. 첫째, 중앙 집중식 접근 방식에서 성능이 좋은 활성화 함수가 연합학습 접근 방식에서 다른 경향을 가짐을 보인다. 둘째, 연합학습 환경에서 활성화 함수 설정과 네트워크 깊이 변화에 따라 성능저하가 일어남을 보인다. 마지막으로, 연합학습의 주요한 두 요소인 클라이언트 참여 비율과 로컬 학습 횟수를 다양하게 변화시키며 적합한 활성화 함수를 선택하는 방법을 제시한다. 관찰을 기반으로 한 우리의 분석은 연합학습을 위한 모델 설계에 대한 통찰력을 제공한다.

1. 서 론

연합학습(FL)은 데이터의 개인 정보 보호할 수 있는 협업 기계 학습 기술의 보편적인 패러다임이다[1,2]. FL 프레임워크에서 각 클라이언트 (예를 들어, 모바일 장치)는 데이터가 아닌 모델의 가중치 값들을 중앙 서버와 통신한다; 모든 로컬 업데이트가 집계되어 중앙 서버 모델이 된다. 데이터가 한곳에 모여 있는 중앙 집중식 접근 방식과 달리, FL 접근 방식은 데이터 이질성 (불균형), 각 클라이언트의 리소스 용량 및 모델 통신으로 인해 다른 양상을 보인다.

이러한 문제를 완화하기 위해 FL 연구자들은 최적화 알고리즘에 제약 조건을 추가하여 중앙 서버 모델의 정확도를 개선했다. [1,2]는 로컬 업데이트 시 안정성을 향상하기 위해 중앙 서버 모델과 각 클라이언트 모델 간 가중치 값의 차이를 proximal term 으로 목적 함수에 추가한다.

그러나, FL 의 인기에도 불구하고 성능에 큰 영향을 미칠 수 있는 훈련과정의 설정에 관한 연구는 크게 다루어지지 않고 있다. 다시 말해, 중앙 집중식 접근 방식에서 사용되는 신경망 구조 및 활성화 함수(예를 들어, ReLU) 등을 검증 없이 연합학습 접근 방식에 적용하고 있다.

본 논문은 이러한 취약 부분을 연구하고자 한다. 가장 먼저, 중앙 집중식 환경에서 대중적인 ReLU 활성화 함수가 FL 환경에서는 최적이 아닐 수도 있음을 보인다. 다음으로, 활성화 함수의 선택과 깊이의 변화에 따른 성능 변화를 관찰하고, 적합한 활성화 함수를 제시한다. 마지막으로, 다양한 클라이언트 참여 비율과 로컬 학습 횟수에 대해서 적합한 활성화 함수를 선택하는 방법을 제시한다.

2. 이론적 배경

2.1 활성화 함수

활성화 함수는 복잡한 표현을 학습하는데 필요한 신경망의 비선형성을 제공하는 층이다. 일반적으로 입력 신호를 계층 방식으로 0 을 중심으로 하는 비선형 출력값으로 변환한다. 출력 신호는 0 근처에 분포되어 있으므로 극단적으로 큰 수로 발산하지 않는다. 본 논문의 실험에서는 다음과 같은 활성화 함수들을 사용하였다: Tanh, HardTanh, ReLU[3], LeakyReLU[6]. 추가적으로 활성화 함수를 사용하지 않는 것은 Identity 로 표기하였다.

2.2 Federated Averaging (FedAvg)

FedAvg[7]는 다음 목적 함수를 최소화한다:

$$\min_{w} f(w) \quad \text{where } f(w) = \frac{1}{N}\sum_{i=1}^{N} f^{(i)}(w)$$

여기서 $f^{(i)}$ 는 클라이언트 i 의 손실 함수이며, N 은 총 클라이언트 수다. 각 라운드에서 클라이언트 참여 비율을 따라 $K \ll N$ 클라이언트가 선택되고 선택된 클라이언트는 로컬 학습 횟수 (epoch) 동안 SGD 를 사용하여 모델을 학습시키며 마지막으로 서버에서 모델들을 집계한다. 모델들을 집계하기 위해 FedAvg 는 다음과 같이 클라이언트 모델들의 가중치를 합산하고 평균을 낸다:

$$w^t = \frac{1}{K}\sum_{k \in S_t} w_k^t$$

2.3 실험 설정

실험 데이터 CIFAR-10 데이터셋[4]을 사용하며, 불균형 데이터를 임의로 만들기 위해 Dirichlet 분포[5]를 사용한다.

모델 아키텍쳐 ConvNet 이라는 간단한 네트워크를 사용한다. ConvNet 은 residual connection 없이 convolution layer 들이 쌓여 있는 구조이다. 아키텍쳐의 이름은 층의 수에 따라 명명한다(예를 들어, ConvNet4 는 4 개의 convolution layer 가 쌓여 있는 구조이다).

특별한 언급이 없으면 ConvNet4 를 모델로 사용한다. 총 20 명의 클라이언트 중에서 각 라운드에서 클라이언트 참여 비율은 0.2(즉, 4 명의 클라이언트)이다. 각 로컬 모델은 5 번의 로컬 epoch 에 대해 학습되며 중앙 서버 모델은 총 100 번의 라운드를 통해 집계된다.

3. 실험 결과

3.1 중앙 집중식 접근 방식 vs. 연합학습(FL) 접근 방식

3.1.1 활성화 함수

표 1 활성화 함수에 따른 중앙 집중식 환경과 FL 환경에서 중앙 서버 정확도

Activation Function	Centralized Setting	FL Setting
Identity	72.00	60.41
Tanh	79.79	62.18
HardTanh	79.10	**63.54**
ReLU	**87.02**	56.03
Leaky ReLU	**87.03**	56.27

표 1 을 보면, 중앙 집중식 환경에서는 ReLU 와 LeakyReLU 가 가장 높은 정확도를 보여준다. 반면에 FL 환경에서는 활성화 함수에 따라 상당히 다른 경향성을 보인다. 결과가 뒤집혀 HardTanh 의 정확도가 가장 높으며 ReLU 와 LeakyReLU 의 정확도가 가장 낮은 것을 확인 할 수 있다.

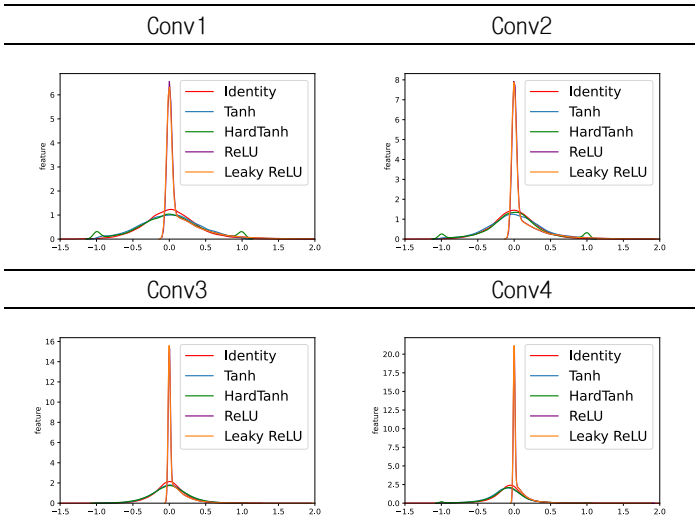

그림 1 Convolution layer 를 통과한 후의 feature distribution

그림 1 은 ConvNet4 에서 각 convolution layer 와 활성화 함수를 거친 후의 feature distribution 을 보여준다. ReLU 와 LeakyReLU 가 0 에 가까운 분포를 갖는 것을 볼 수 있다. 이는 0 에서의 이산성으로 인해 발생한다. 그리고 feature 가 더 깊은 층을 지날수록 0 근처의 밀도가 높아진다. 중앙 집중식 환경에서 ReLU 및 LeakyReLU 는 0 에서의 불연속성과 선형 함수와 유사한 형태로 인해 기울기 소실 및 기울기 폭주를 예방하며 잘 작동된다. 그러나 FL 환경에서 0 에서의 불연속성은 부정적인 영향을 준다. FL 의 집계 단계에서 불연속성을 고려하지 않고 평균화를 수행하며, 이로 인해 ReLU 및 LeakyReLU 는 0 근처에서 높은 민감도를 얻는다. 간단히 말해, 이미지가 더 깊은 층을 통과함에 따라 0 의 밀도가 높아지는 것을 보아 0 이 아닌 값들이 0 으로 바뀐다. 그러나 Identity, Tanh, HardTanh 는 정규 분포와 유사한 분포를 보여 FL 의 집계 단계에 대한 민감도가 상대적으로 낮다. 더불어 Tanh 와 HardTanh 는 비선형성으로 인해 Identity 보다 더 나은 정확도를 보여주며 HardTanh 는 선형적인 형태로 Tanh 보다 나은 성능을 보여준다. 결과적으로 활성화 함수의 모양은 FL 의 집계 단계에 대해 다른 민감도를 제공한다.

3.1.2 모델의 깊이(Depth)

표 2 깊이에 따른 중앙 집중식 환경에서의 정확도

Model	Centralized Setting
ConvNet3	84.63
ConvNet4	87.02
ConvNet5	88.96
ConvNet6	89.47
ConvNet7	**90.38**

표 2 는 중앙 집중식 환경에서 깊은 모델을 씀에 따른 최종 정확도를 보여준다. 중앙 집중식 환경에서는 모델이 깊어질수록 더 높은 정확도를 가진다.

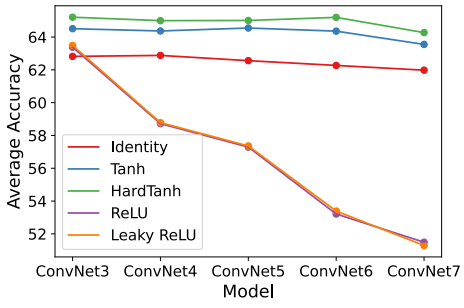

그림 2 깊이에 따른 FL 환경에서의 정확도

그림 2 는 FL 환경에서 중앙 서버의 최종 정확도를 보여준다. HardTanh, Tanh 및 Identity 는 모든 깊이의 모델에서 비슷한 정확도를 보인다. 반면에 ReLU 와 LeakyReLU 는 모델이 깊어질수록 정확도가 낮아진다. 이러한 경향성은 3.1.1 에서 언급한 민감도 때문에 나타난다. 그림 1 에서 볼 수 있듯이 ReLU 와 LeakyReLU 는 깊은 층에서 0 근처의 feature density 가 더 크게 나타나며 이는 높은 민감도를 의미하며 결과적으로 모델이 깊어짐에 따라 심각한 정확도 저하가

발생한다. 따라서 단순히 convolution layer 를 쌓는 경우 ReLU 와 LeakyReLU 는 이점이 없는 것을 확인할 수 있다.

3.2 FL 환경에 따른 성능 비교

3.2.1 클라이언트 참여 비율

환경적인 제한으로 인해 연합학습에서 클라이언트의 참여는 제한된다. 예를 들어, cross-silo 설정의 경우 많은 클라이언트의 참여가 가능하며, cross-device 설정의 경우에는 적은 클라이언트만이 참여 가능하다. 이에, 이 부분에서는 다양한 클라이언트 참여도에 따른 활성화 기능을 설정하는 방법에 대해 설명한다.

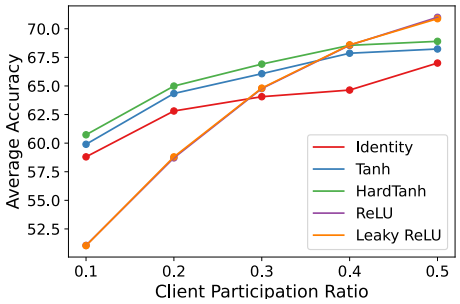

그림 2 클라이언트 참여 비율에 따른 FL 환경에서의 정확도

그림 3 은 다양한 클라이언트 참여 비율에 대한 중앙 서버의 최종 정확도를 보여준다. 클라이언트 참여도가 낮은 경우, HardTanh 가 가장 높은 정확도를 보인다. 그러나 클라이언트 참여가 증가함에 따라 ReLU 및 LeakyReLU 가 더 높은 정확도 상승율을 통해 HardTanh 를 추월한다. 클라이언트의 참여율이 높을수록 중앙 집중식 환경과의 유사도가 높아지게 되며, ReLU 와 LeakyReLU 의 정확도가 높아진다. 결과적으로 HardTanh, Tanh 및 Identity 는 ReLU 및 LeakyReLU 보다 FL 의 집계 과정으로 인한 정확도 저하의 영향이 적기 때문에 낮은 클라이언트 참여도에서 더 잘 작동한다. 그리고 클라이언트 참여도가 높을수록 중앙 집중식 환경에서의 정확도가 높은 ReLU 와 LeakyReLU 가 더 잘 작동한다

3.2.2 로컬 학습 횟수 (Epoch)

연합학습은 환경으로 인해 로컬 epoch 설정에 제한이 있다. 이 부분에서는 서로 다른 로컬 epoch 에서 활성화 함수를 설정하는 방법에 대해 설명한다.

그림 4 는 서로 다른 로컬 epoch 로 학습한 중앙 서버의 최종 정확도를 보여준다. 일반적으로 로컬 epoch 가 증가함에 따라 정확도는 증가한다. 그러나 작은 로컬 epoch 에서는 HardTanh, Tanh 및 Identity 가 ReLU 및 LeakyReLU 보다 더 높은 정확도를 보여준다. 그리고 로컬 epoch 가 증가할수록 ReLU 와 LeakyReLU 의 정확도가 HardTanh 보다 높아진다. 3.1.1 에서 말했듯이 FL 의 집계 단계는 특히 ReLU 및 LeakyReLU 에 심각한 정확도 저하를 일으킨다. 그러나 로컬 epoch 가 증가함에 따라 로컬 모델의 성능향상이 중앙 집중식 환경과 같이 ReLU 와 LeakyReLU 에서 더 많이 일어나고 집계 단계에서의 정확도 저하는 동일하게 적용되어 전체적인 정확도가 증가하게 된다. 결과적으로 낮은 로컬 epoch 에서는 HardTanh, 높은 로컬 epoch 에서는 ReLU 및 LeakyReLU 가 정확도 측면에서 우위에 있다.

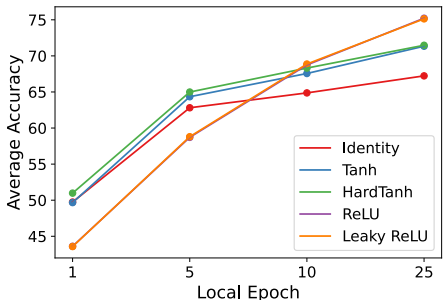

그림 4 로컬 epoch 에 따른 FL 환경에서의 정확도

4. 결론

본 논문에서는 FL 환경에서 활성화 함수의 능력을 명확히 한다. 주요 발견은 ReLU 및 LeakyReLU 의 개별 기능으로 인해 FL 환경에서 정확도가 떨어지고 HardTanh 는 대부분 환경에서 다른 활성화 함수보다 성능이 우수하다는 것이다. 또한 FL 설정에서 성능을 높이는 데 도움이 되는 다양한 로컬 epoch, 클라이언트 참여도, 깊이에서 활성화 기능을 선택하기 위한 정보를 제공한다. FL 환경에서 최적화된 활성화 함수를 찾는 연구가 필요하며, 이는 우리의 연구가 뒷받침한다.

[참고 문헌]

[1] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems, 2:429-450, 2020.

[2] Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S.,and Suresh, A. T. Scaffold: Stochastic controlled averag-ing for federated learning. In International Conference on Machine Learning, pp. 5132-5143. PMLR, 2020.

[3] Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Dou-glas, R. J., and Seung, H. S. Digital selection and ana-logue amplification coexist in a cortex-inspired silicon circuit. Nature, 405(6789):947-951, 2000

[4] KRIZHEVSKY, Alex, et al. Learning multiple layers of features from tiny images. 2009.

[5] HSU, Tzu-Ming Harry; QI, Hang; BROWN, Matthew. Measuring the effects of non-identical data distribution for federated visual classification. arXiv preprint arXiv:1909.06335, 2019.

[6] Maas, A. L., Hannun, A. Y., Ng, A. Y., et al. Rectifier nonlinearities improve neural network acoustic models. In Proc. icml, volume 30, pp. 3. Citeseer, 2013.

[7] McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In Artificial intelli-gence and statistics, pp. 1273-1282. PMLR, 2017.