# 연합학습 과정에서의 양자화 매개변수 학습

김유준○ 홍충선*

경희대학교 컴퓨터공학과

{yj4889○, cshong* }@khu.ac.kr

# Learning Quantization Parameters in Federated Learning

Youjun Kim○, ChoongSeon Hong*

Department of Computer Science and Engineering, Kyung Hee University

**요 약**

연합학습은 중앙 서버의 데이터 수집 없이 네트워크 말단 디바이스에서 딥 네트워크를 학습시키는 분산 학습 방법이다. 중앙 서버의 데이터 부재는 데이터를 활용한 딥 네트워크 압축이 불가능하다는 것을 의미한다. 본 논문은 연합학습 과정에서 사용자 데이터 유출 없이 딥 네트워크의 부동 소수점 연산을 현저히 줄이는 양자화 방법(LQFL)을 새롭게 제시한다. 딥 네트워크의 weight, bias와 동시에 양자화 매개변수는 일반적인 학습 Loss에 의해 학습된다. 본 논문에서 제시한 새로운 방법인 LQFL은 딥 네트워크를 압축시킴과 동시에 일반 연합학습보다 Test 데이터셋에 대한 정확도가 2% 더 높은 것을 볼 수 있다.

## 1. 서 론

연합학습은 사용자의 데이터 프라이버시를 보존할 수 있는 분산학습 방법이다. 모든 데이터는 중앙 서버가 아닌 사용자 디바이스 내에 저장되어 있으며 딥 네트워크는 사용자 디바이스에서 학습된다. 따라서 일반적인 딥러닝 과정과 달리 연합학습은 학습 과정에서 데이터가 사용자 디바이스 외부로 유출되지 않기 때문에 프라이버시를 완벽히 보존할 수 있다[1]. 딥 네트워크는 중앙 서버로 모아진 데이터뿐만 아니라 보안이 중요한 데이터까지 학습할 수 있게 되어 모든 데이터를 활용할 수 있다.

이러한 연합학습은 네트워크 말단 디바이스의 컴퓨팅 및 통신 능력이 월등히 증가함에 따라 자원할당과 통신 효율 등에서 많은 연구가 진행되고 있지만, 연합학습 과정에서의 딥 네트워크 압축과 관련한 연구는 미약하다. 딥 네트워크 압축 분야는 네트워크 말단 디바이스에서 적은 용량을 가지고 더욱 빠른 추론을 가능하게 하며 현재 각광받고 있는 'On device AI' 에 꼭 필요한 기술이다.

특히 압축 분야 중에서 양자화 기술은 네트워크의 weight 및 activation을 2 bits ~ 16 bits로 표현하여 부동 소수점 연산을 줄여 딥 네트워크의 추론을 더욱 빠르게 한다. [2]는 약간의 정확도 손실을 감수하고 기존 32 bits의 네트워크를 2 bits로 양자화하였다. [2]와 같이 weight 및 activation의 데이터 분포를 분석하여 양자화하는 방식과 달리 [3], [4]는 역전파를 활용하여 새로운 변수를 학습하고 네트워크를 양자화하였다.

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(No.2019-0-01287, 분산 엣지를 위한 진화형 딥러닝 모델생성 플랫폼)과 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(2020-0-00364, AI기반 차량용 통신 기술 향상을 위한 NPU와 응용 시스템 개발)*Dr. CS Hong is the corresponding author

본 논문에서는 일반적인 연합학습 과정에서 네트워크 학습과 역전파를 활용한 양자화 변수 학습을 동시에 시행하는 새로운 학습 방법인 LQFL(Learning Quantization parameters in Federated Learning)을 제시한다. LQFL은 사용자 프라이버시를 완벽히 보존할 수 있으며 딥 네트워크 학습과 양자화를 동시에 진행할 수 있다.

## 2. 관련연구

### 2.1 연합학습

연합학습은 중앙 서버에서 글로벌 모델을 사용자 디바이스로 보내 학습하고 학습된 모델은 다시 중앙 서버로 보내져 통합된다. 통합은 ①식에 의해 이루어진다[1].

$$w_{t+1} \leftarrow \sum_{n=1}^{N} \left(\frac{k_n}{k}\right) w_{t+1}^{n} \qquad ①$$

$k$는 전체 데이터 수, $k_n$은 $n$번째 사용자의 데이터 수, $w$는 딥 네트워크의 weight이다. SGD (Stochastic Gradient Descent의 gradient $g$에 의해 weight는 $w_{t+1} \leftarrow w_t - \gamma g_n$로 업데이트되며 이는 FedSGD $w_{t+1} \leftarrow w_t - \gamma \sum_{n=1}^{N} \left(\frac{k_n}{k}\right) g_n$로 표현되고 이는 ①식과 같다고 할 수 있다[1]. 즉, 딥 네트워크가 학습할 때 gradient에 의한 역전파 알고리즘을 사용하기 때문에 연합학습이 가능하게 된다.

### 2.2 양자화

딥 네트워크 양자화는 weight 및 activation을 기존 32 bits에서 더욱 낮은 bits로 양자화하여 부동 소수점 연산(FLOPS)을 줄여 더욱 빠른 추론이 가능하게 하는 데에

목적이 있다. 단순히 학습된 네트워크와 양자화된 네트워크의 차이를 최소로, 즉 양자화 에러를 최소로 하는 기존의 방식과 달리 [3]은 일반적인 네트워크 학습 과정에서의 Loss를 이용하여 weight 학습과 동시에 양자화 매개변수를 역전파 알고리즘을 활용하여 학습하는 방법을 사용하였다. [4]도 [3]과 같이 역전파 알고리즘을 이용하여 양자화 매개변수를 학습하였고 [3]보다 더 적은 수의 양자화 파라미터를 사용하였다. weight 또는 activation은 ②식[4]에 의해 양자화된다.

$$\bar{v} = \lceil clip(\frac{v}{q}, -Q_N, Q_P) \rfloor \quad ②$$

$v$는 weight 또는 activation이며 네트워크를 $t$ bits로 양자화할 때 weight라면 $Q_N = 2^{t-1}$, $Q_P = 2^{t-1} - 1$, activation이라면 $Q_N = 0$, $Q_P = 2^t - 1$이다. $clip$은 $v/q$가 $-Q_N$보다 작으면 $-Q_N$, $Q_P$보다 크면 $Q_P$, 사이 값이면 $v/q$이다. 「」은 반올림연산이다. $q$는 양자화 변수로써 ③식[4]의 gradient (straight through estimator [5, 6]) 와 네트워크 Loss에 의해 학습된다.

$$\frac{\partial \hat{v}}{\partial q} = \begin{cases} -v/q + \lceil v/q \rfloor & \text{if } -Q_N < v/q < Q_P \\ -Q_N & \text{if } v/q \le -Q_N \\ Q_P & \text{if } v/q \ge Q_P \end{cases} \quad ③$$

또한 gradient scale을 조정하여 같은 bits로의 양자화 네트워크에서 [3]보다 더 높은 정확도를 보여줬다.

### 3. 제안사항

본 연구에서는 [4]와 같은 양자화 방식을 사용하였다. 먼저 forward pass에서 $v$(weight or activation)은 양자화 변수인 $q$에 의해 $v_q$로 양자화된다. 양자화된 $v_q$와 input data에 의해 output을 계산하고 Loss를 구한다. Backward pass에선 Loss에 의해 gradient가 구해지고 역전파 알고리즘을 시행한다. 그림 1은 $q$에 의해 양자화되는 뉴럴 네트워크를 간략히 표현한 것이다. 상위 3개 뉴런이 일반 32 bits 뉴럴 네트워크이며 하위 3개 뉴런이 양자화된 네트워크이다. 학습이 완료되면 하위 3개 뉴런만 남게 되며 forward pass에서 양자화된 activation과 weight의 곱인 $a^q * w^q$에 의해 부동 소수점 연산이 현저히 줄어들게 된다.

그림 1에서 학습되는 매개변수는 weight와 양자화 매개변수이다.

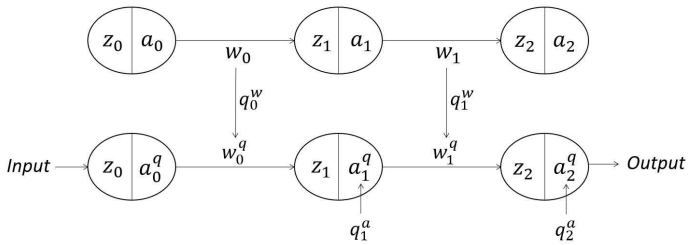


그림 1. 양자화 매개변수와 뉴럴 네트워크

양자화 매개변수는 activation을 위한 $q^a$와 weight를 위한 $q^w$가 있다. chain rule에 의해 $q$와 $w$는 다음과 같이 업데이트된다.

$$q_2^{a+1} = q_2^a - \gamma(\frac{\partial L}{\partial a_2^q}\frac{\partial a_2^q}{\partial q_2^a})$$

$$q_1^{w+1} = q_1^w - \gamma(\frac{\partial L}{\partial a_2^q}\frac{\partial a_2^q}{\partial z_2}\frac{\partial z_2}{\partial w_1^q}\frac{\partial w_1^q}{\partial q_1^w}) \quad ④$$

$$w_1^{+1} = w_1 - \gamma(\frac{\partial L}{\partial a_2^q}\frac{\partial a_2^q}{\partial z_2}\frac{\partial z_2}{\partial w_1^q}\frac{\partial w_1^q}{\partial q_1^w}\frac{\partial q_1^w}{\partial w_1})$$

즉 양자화 파라미터인 $q$도 weight 업데이트인 $w_{t+1} \leftarrow w_t - \gamma g_n$와 마찬가지로 $q_{t+1} \leftarrow q_t - \gamma g_n$ 업데이트되고 $q_{t+1} \leftarrow q_t - \gamma \sum_{n=1}^{N} (\frac{D_n}{D}) g_n$식으로 [1]의 FedSGD 알고리즘을 적용할 수 있다. 그리고 연합학습 통합식에도 다음과 같이 적용 가능하다.

$$q_{t+1} = \sum_{n=1}^{N} \frac{D_n}{D} q_{t+1}^n \quad ⑤$$

$D$는 총 데이터의 개수이며 $D_n$은 $n$번째 client가 가지고 있는 데이터의 개수다.

딥 네트워크의 양자화를 위한 $q$가 stochastic gradient descent에 의한 역전파 알고리즘에 의해 업데이트되기 때문에 연합학습 과정에서 Loss를 이용하여 weight와 양자화 파라미터를 동시에 업데이트한다. 딥 네트워크 생성과 통합(①, ⑤)을 하는 서버와 weight 및 양자화 파라미터를 학습하는 클라이언트 알고리즘인 LQFL을 Algorithm 1 과 2에 나타내었다.

### 4. 성능평가

본 논문은 cifar10 데이터를 사용하였고 모든 데이터는 중앙 서버가 아닌 사용자 디바이스에만 저장된다고 가정한다. pytorch를 이용하여 cross entropy loss 함수를 사용하였고 momentum 은 0.9이다. 총 학습참여자 수는

100명, 참여비율은 0.1, 로컬 반복은 5회, batch 크기는 100이다. 딥 네트워크 모델은 ResNet18[7]을 사용하였으며 LQFL 과정에서 모델은 4 bits로 양자화된다. 그림 2와 3은 양자화하지 않는 전통적인 연합학습(FL)과 LQFL을 2000번 학습하여 loss, accuracy를 비교하는 그래프이다. LQFL과 기존 연합학습의 학습속도가 거의 비슷하다는 것을 알 수 있다. 또한 4bits로 양자화된 LQFL의 성능이 32bits의 기존 연합학습성능과 거의 같다는 것을 표 1을 통해 확인할 수 있다.

Algorithm 1. ***Server*** side: Learning Quantization parameters in Federated Learning

Server:
  **initialize** $w_0$, $q_0$, $E$ : epochs,
  $N$ : number of clients,
  $f$ : fraction of participation in learning
  **for** epochs in E :
    $P$ = max($N*f$, 1)
    $S \leftarrow$ random set of $P$ clients
    **for** $s$ in $S$ :
      $w_{t+1}^{s}, q_{t+1}^{s} \leftarrow$ Client($w_t$, $q_t$, s)

$$w_{t+1} = \sum_{n=1}^{N} \frac{D_n}{D} w_{t+1}^{n}$$

$$q_{t+1} = \sum_{n=1}^{N} \frac{D_n}{D} q_{t+1}^{n}$$

Algorithm 2. ***Client*** side: Learning Quantization parameters in Federated Learning

Client($w, q, s$):
  **initialize** $B$: batchs,
  $E$: local epochs,
  $L$: number of deep network layers,
  func: activation function

  {forward pass}
  **for** $l$ in (1, $L$-1) :
    $w_l^q$ = quantize($w_l$, $q_l^w$) ※②
    $z_{l+1}$ = forward($a_l^q$, $w_l^q$)
    $a_{l+1}$ = func($z_{l+1}$)
    $a_{l+1}^q$ = quantize($a_{l+1}$, $q_{l+1}^a$) ※②

  {learning network}
  **for** e in $E$ :
    **for** $b$ in $B$ :

$$w = w - \gamma(\frac{\partial L}{\partial w}; b) \quad ※\frac{\partial L}{\partial w} \leftarrow ④\text{-}3$$

$$q^a = q^a - \gamma(\frac{\partial L}{\partial q^a}; b) \quad ※\frac{\partial L}{\partial q^a} \leftarrow ④\text{-}1$$

$$q^w = q^w - \gamma(\frac{\partial L}{\partial q^w}; b) \quad ※\frac{\partial L}{\partial q^w} \leftarrow ④\text{-}2$$

  **return** $w, q$

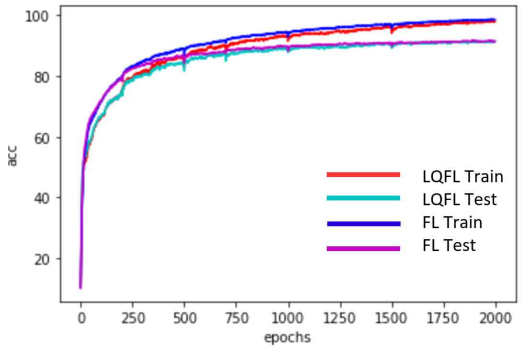


그림 2. LQFL과 Federated Learning Accuracy 그래프

표 1. Best Accuracy 및 Best Loss 비교

| Method | Test Dataset | | Train Dataset | |
|---|---|---|---|---|
| | Best Loss | Best Accuracy | Best Loss | Best Accuracy |
| FL | 0.328594 | 91.73 | 0.039707 | 98.742 |
| LQFL | 0.321792 | 91.6 | 0.054859 | 98.126 |

## 5. 결 론

본 논문은 연합학습과정에서 딥 네트워크의 weight 및 bias 학습과 동시에 양자화 매개변수를 학습하는 LQFL을 새롭게 제시하였다. LQFL은 데이터가 사용자 디바이스 외부로 유출되지 않기 때문에 프라이버시를 완벽히 보존할 수 있으며 딥 네트워크를 압축하여 더욱 빠른 추론을 가능하게 한다. LQFL에 의해 압축된 4bits 네트워크의 성능이 기존 연합학습에 의해 압축된 32bits 네트워크의 성능과 거의 비슷하다는 것을 확인할 수 있었다. 다음 연구로는 학습 파라미터를 다양하게 사용하여 더 높은 정확도의 LQFL을 연구하고 다른 딥 네트워크에 대한 실험 및 컴퓨팅, 통신부하문제를 해결하는 것이다.

[1] McMahan, H. B., Moore, E., Ramage, D., & Hampson, S. (2016). Communication-efficient learning of deep networks from decentralized data. arXiv preprint arXiv:1602.05629.
[2] Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., & Bengio, Y. (2016). Binarized neural networks. In Advances in neural information processing systems (pp. 4107-4115).
[3] Jung, S., Son, C., Lee, S., Son, J., Han, J. J., Kwak, Y., ... & Choi, C. (2019). Learning to quantize deep networks by optimizing quantization intervals with task loss. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4350-4359).
[4] Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., & Modha, D. S. (2019). Learned step size quantization. arXiv preprint arXiv:1902.08153.
[5] Bengio, Y., Léonard, N., & Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432.
[6] Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., & Zou, Y. (2016). Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160.
[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).