

Unification of Deep Learning Model trained by Parallel Learning in Security environment

Jong-Lark Lee*

*Professor, Faculty of Cyber Security, Yeungnam University College, Daegu, Korea

[Abstract]

Recently, deep learning, which is the most used in the field of artificial intelligence, has a structure that is gradually becoming larger and more complex. As the deep learning model grows, a large amount of data is required to learn it, but there are cases in which it is difficult to integrate and learn the data because the data is distributed among several owners and security issues. In that situation we conducted parallel learning for each users that own data and then studied how to integrate it. For this, distributed learning was performed for each owner assuming the security situation as V-environment and H-environment, and the results of distributed learning were integrated using Average, Max, and AbsMax. As a result of applying this to the mnist-fashion data, it was confirmed that there was no significant difference from the results obtained by integrating the data in the V-environment in terms of accuracy. In the H-environment, although there was a difference, meaningful results were obtained.

▶ **Key words:** Security, Distributed Learning, Data Parallel, Deep Learning

[요 약]

최근 인공지능 분야에서 가장 많이 사용하는 딥러닝은 그 구조가 점차 크고 복잡해지고 있다. 딥러닝 모델이 커질수록 이를 학습시키기 위해서는 대용량의 데이터가 필요하지만 데이터가 여러 소유 주체별로 분산되어 있고 보안 문제로 인해 이를 통합하여 학습시키기 어려운 경우가 발생한다. 우리는 동일한 딥러닝 모형이 필요하지만 보안 문제로 인해 데이터가 여러곳에 분산되어 처리될 수 밖에 없는 상황에서 데이터를 소유하고 있는 주체별로 분산 학습을 수행한 후 이를 통합하는 방법을 연구하였다. 이를 위해 보안 상황을 V-환경과 H-환경으로 가정하여 소유 주체별로 분산학습을 수행했으며 Average, Max, AbsMax를 사용하여 분산학습된 결과를 통합하였다. mnist-fashion 데이터에 이를 적용해 본 결과 V-환경에서는 정확도 면에서 데이터를 통합시켜 학습한 결과와 큰 차이가 없음을 확인할 수 있었으며, H-환경에서는 차이는 존재하지만 의미있는 결과를 얻을 수 있었다.

▶ **주제어:** 보안, 데이터병렬화, 분산학습, 딥러닝

-
- First Author: Jong-Lark Lee, Corresponding Author: Jong-Lark Lee
 - *Jong-Lark Lee (jllee@ync.ac.kr), Faculty of Cyber Security, Yeungnam University College
 - Received: 2021. 11. 22, Revised: 2021. 12. 15, Accepted: 2021. 12. 15.

I. Introduction

최근 딥러닝 모델이 음성인식, 자연어처리, 컴퓨터비전 등에서 폭넓게 사용되고 있다.

딥러닝 모델은 설계된 다수의 layer에 존재하는 가중치(weights)를 파라미터로 가지며 데이터 학습을 통해 최적값을 찾게 된다. 이때 가중치의 개수가 많아지고 학습 데이터의 양이 증가하게 되면, 컴퓨터의 계산 양은 가파르게 증가한다. 최근에 사용되는 딥러닝 모델의 경우 가중치의 개수가 수십만개에 달하고 학습 데이터의 양도 하나의 컴퓨터에 저장하기 어려울 정도로 큰 경우가 많으므로 한 대의 컴퓨터로 이러한 파라미터를 추정하기 위한 계산이 불가능한 경우도 존재한다.

또한 동일한 딥러닝 모델이 필요하지만 이를 학습시킬 데이터의 소유 주체가 상이한 경우가 존재한다. 이 경우 개인정보보호법의 문제로 인해 데이터를 하나로 통합하여 처리하기 불가능한 경우가 있다. 예를 들어 홍채인식 인증을 위한 딥러닝 모델이 필요한 다수의 기업이 있고, 각 기업마다 제한된 수의 홍채 관련 데이터를 가지고 있지만 보안 문제로 인해 그 정보를 한곳에 모아 통합된 환경에서 딥러닝 모델을 생성하는 것은 불가능한 상황이 이에 해당한다. 이 경우 소유 주체별로 분리되어 저장된 데이터에 동일한 딥러닝 모델을 따로 학습시킬 수밖에 없다.

우리는 보안 문제로 인해 분리된 환경하에서 분산 추정된 딥러닝 모델들을 하나로 통합시키는 방법에 관심을 가지며 이 통합된 모델이 분리되지 않은 환경에서 추정된 모델과 어느 정도의 차이가 나는지를 연구하고자 한다.

기존의 딥러닝 모델에서의 분산학습은 학습 시간의 비효율성을 개선하기 위한 목적으로 사용되고 있는데, 여러 대의 범용 컴퓨터가 분산하여 모델을 학습시키면 짧은 시간에 슈퍼 컴퓨터 한 대가 수행한 학습 효과를 낼 수 있기 때문이다. 이를 위한 연구는 크게 세 가지 분야로 구분할 수 있다. 첫째는 분산학습을 위한 프레임워크의 설계 및 구현에 관한 것으로서 텐서플로우(TensorFlow)[1], 하둡(Hadoop)[2], 스파크(Spark)[3], 카페(Caffe)[4]등의 프레임워크가 개발되어 사용되고 있다.

둘째는 분산학습 시 개별 컴퓨터에 분산되어 저장되는 개체에 대한 연구이다. 처리해야 하는 데이터의 크기가 너무 커서 이를 한 대의 컴퓨터에 저장하여 처리하기 불가능한 경우 이를 여러대의 컴퓨터에 분산하여 처리하기 위한 데이터 병렬화(data parallel)와 추정해야 하는 모델의 크기가 너무 커서 하나의 컴퓨터에서 계산하는데 너무 오랜 시간이 소요되는 경우에 사용하는 모델 병렬화(model parallel)이다[5].

세 번째는 분산학습 시에 개별 로컬 컴퓨터에서 추정된 파라미터들을 서버로 전송하여 새로운 글로벌 파라미터를 생성하게 되는데 파라미터 업데이트를 어떻게 하는 것이 효율적인지에 관한 연구이다. 이때 글로벌 파라미터를 업데이트하기 위한 알고리즘에 대한 연구와 파라미터의 버전관리를 위한 방법-동기식(synchronous)과 비동기식(asynchronous)-에 대한 연구[6]로 구분된다.

본 연구에서는 동일한 딥러닝 모델을 상이한 컴퓨터 환경에 분산되어 저장된 제한된 데이터만으로 학습시킨 후 이를 중앙에서 통합하여 통합된 딥러닝 모델을 만드는 과정을 살펴본다. 이때 로컬에서 컴퓨터에서 생성된 파라미터를 통합하여 새로운 글로벌 파라미터를 생성하는 방법으로서 average, max, absmax 등을 제안하며 이를 사용하여 생성된 통합모델이 정확도(accuracy)의 측면에서 기존의 방법과 어느 정도 효율적인지를 실험을 통해 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련된 연구를 설명한다. 3장에서는 개별 학습모델 통합을 위한 방법을 제안하며, 4장에서는 제안된 방법을 fashion-mnist 데이터에 CNN 모델을 적용한 실험의 과정과 그 결과를 설명한다. 그리고 5장에서는 연구 내용의 요약과 함께 향후 연구 방향에 대해 설명하는 결론으로 끝을 맺는다.

II. Related Works

1. Parallel Learning

최근 사물이나 이미지 인식 등에서 많이 사용되고 있는 딥러닝 모델에서는 인식오차를 줄이기 위해 네트워크를 구성하는 layer의 수를 증가시키는 것이 보편화되어 있으며, 이에 따라 추정해야 하는 파라미터의 개수도 기하급수적으로 증가하고 있다. 예를 들어 이미지 인식 대회인 ILSVRC에서 2013년 우승을 차지한 AlexNet 모델의 경우 8개의 layer를 사용하여 오차율 16.4%를 기록한 반면 2015년 우승을 차지한 ResNet 모델은 152개의 layer를 사용하여 오차율을 3.57%로 줄였다.[7]

문제는 모델에서 사용하는 layer의 수와 추정해야 하는 파라미터의 개수가 큰 대형 네트워크 모델의 경우 이를 학습시키기 위해 필요한 데이터셋의 크기가 매우 커야하며, 이를 계산하기 위한 컴퓨터의 성능도 매우 좋아야 한다는 것이다. 성능이 좋은 CPU나 GPU를 사용하며 멀티쓰레딩을 지원하는 컴퓨터라 할지라도 대형모델을 학습시키기 위해서는 많은 시간이 소요된다.

이를 해결하기 위한 방법으로 분산학습을 사용한다. 분산학습은 일반적으로 데이터 병렬화(Data Parallelism)와 모델 병렬화(Model Parallelism)로 구분된다. 데이터병렬화는 학습에서 사용되는 데이터셋이 너무 커서 단일 컴퓨터에 저장되어 처리되기 어려운 환경에서 이를 여러개의 컴퓨터로 분산하여 처리하기 위한 방법이며, 모델 병렬화는 딥러닝 모델이 단일 컴퓨터에서 처리되기 어려울 정도로 큰 경우에 layer별로 개별 컴퓨터에 분산시켜 처리하기 위한 방법이다.

우리의 문제에서는 데이터의 소유 주체가 서로 달라 이를 한 번에 통합하여 처리하기 불가능한 환경이므로 데이터 병렬화에 연구의 초점을 맞춘다.

1.1 Data Parallelism

데이터 병렬화에서는 동일 구조의 모델을 사용하지만, 계산은 데이터셋이 분산되어 있는 컴퓨터에서 개별적으로 수행된다. 개별 컴퓨터에서는 계산을 통해 파라미터의 로컬 최적값을 찾는다. 그리고 이 로컬 파라미터의 값을 중앙의 파라미터 서버로 전송한다. 중앙의 파라미터 서버에서는 개별 컴퓨터에서 전송된 로컬 파라미터의 값을 활용하여 글로벌 파라미터를 생성한 후 이를 다시 개별 컴퓨터로 전송한다. 이때 개별 컴퓨터에서 계산된 파라미터들을 서버에서 어떻게 업데이트하는지에 따라 다음과 같이 구분된다.

- 동기식업데이트(synchronous update)
- 비동기식업데이트(asynchronous update)

동기식업데이트에서는 개별 컴퓨터에서 추정된 파라미터들이 모두 서버로 도착할 때까지 기다린 후 업데이트를 수행한다. 이 경우 마지막 컴퓨터가 자신의 파라미터 추정값을 전송할 때까지 업데이트를 대기해야 하는 문제점이 존재한다. 반면 비동기식 업데이트에서는 로컬 컴퓨터가 새로운 파라미터의 추정값을 계산하는 즉시 서버로 전송하여 파라미터의 값을 업데이트하게 하고, 이를 개별 컴퓨터로 리턴하는 방식이다. 하지만 일반적으로 로컬 컴퓨터에서 계산된 파라미터의 값만을 가지고 계산된 글로벌 파라미터의 값이 전체를 대표하기 어렵다는 측면에서 최적화하는데 더 많은 반복이 필요하다는 단점이 존재한다. 본 연구에서는 각 기업의 하드웨어의 성능이 유사하고 네트워크의 속도도 유사하다는 가정하에 동기식 업데이트를 사용하기로 한다.

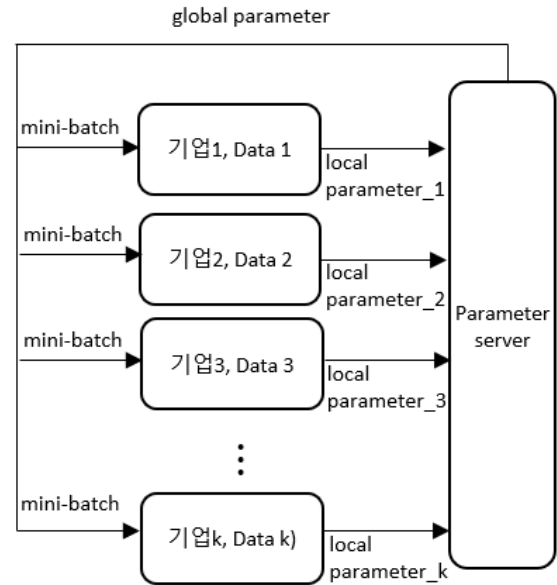


Fig. 1. The structure of data parallel

1.2 Stochastic Gradient Descent

딥러닝 모델에서 학습 알고리즘의 가장 큰 목적은 손실 함수(loss function)를 최소화 하는 방식으로 각 계층의 가중치를 추정하는 것이다. 이를 위한 기본적인 아이디어는 손실함수의 기울기와 반대되는 방향으로 일정크기만큼 이동하면서 손실함수의 최소값을 찾는 방식이다. 이를 수식으로 표현하면 수식(1)과 같다.

$$w' \leftarrow w - \alpha \nabla_w L_{total} \quad (1)$$

여기서 $L_{total} = \frac{1}{n} \sum_{i=1}^n L_i$ 은 모든 데이터에 대한 총 손실함수값이고, $\nabla_w L_{total}$ 은 가중치에 대한 총 손실함수의 기울기(gradient)이며 α 는 학습률(learning rate)이다.

하지만 이동지점마다 손실함수의 값(L_{total})을 계산하기 위해 모든 데이터를 사용하여 계산하게 되면 계산량이 너무 많아지므로 전체 데이터 대신 데이터의 일부그룹(mini-batch)를 사용하여 손실함수의 값을 계산하도록 하는 방법인 Stochastic Gradient Descent를 사용한다.

$$w' \leftarrow w - \alpha \nabla_w L_{minibatch} \quad (2)$$

여기서 $L_{minibatch} = \frac{1}{m} \sum_{i \in M} L_i$ 로서 mini-batch에 있는 데이터를 사용한 손실함수값이며, $\nabla_w L_{minibatch}$ 는 mini-batch를 사용한 손실함수의 기울기 값이다. 또한 m 은 mini-batch내에 있는 데이터의 갯수이다.

우리는 각 이미지에 대한 손실함수로서 식(3)의 logistic loss function을 사용하였다.

$$L_i = -f_{y_i} + \log \sum_j e^{f_j} \quad (3)$$

여기서 f_j 는 class score vector \mathbf{f} 의 j 번째 요소이다.

분산학습에서는 Fig.2와 같이 가중치의 업데이트 과정을 병렬화하여 처리함으로써 학습의 속도를 보다 빠르게 수행하도록 한다.

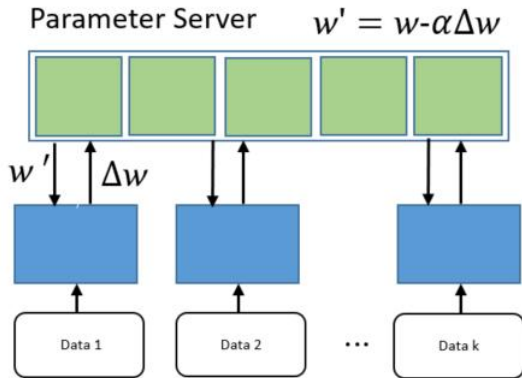


Fig. 2. data parallelism for updating parameters[5]

2. CNN Model

CNN(Convolutional Neural Network)는 딥러닝 모델 중에서 사물이나 이미지 인식등과 같은 컴퓨터비전 분야에서 가장 많이 사용되고 있는 인공지능 기법으로서 인간의 시각에 비취지는 사물이 대뇌피질에 존재하는 뉴런에 의해 어떻게 처리되어 인식되는지를 모방하여 설계되었다[8].

CNN이 다른 딥러닝 모델과 다른 점은 크게 3가지로 구분할 수 있는데, 첫 번째는 고정된 크기의 Sliding Kernel을 사용하여 일정한 간격만큼(이를 stride라 부름) 움직여 가며 이미지를 읽어 들여 입력데이터를 생성하는 convolution layer를 사용한다는 점이다. 두 번째는 입력된 이미지의 크기를 축소하는 Pooling(down sampling이라고도 부름) layer를 사용한다는 것이고, 세 번째는 비선형 activation 함수인 sigmoid, tanh, ReLU등을 사용한다는 점이다. 이 중에서 ReLU는 다른 함수에 비해 SGD(Stochastic gradient Descent)의 수렴속도가 빨라 계산시간이 현저히 줄어드는 장점이 있어 많이 사용한다고 알려져 있다.[9]

III. Unification of Local Learning Models

본 연구에서 사용하는 딥러닝 모델의 구조는 Fig.3과 같다. 즉 개별 기업에서는 상이한 데이터를 지나 동일한

구조의 딥러닝 모델을 이용하여 학습을 수행한다. 개별 기업들은 자신들이 소유하고 있는 모든 데이터를 활용하여 학습을 수행한다. 로컬 학습이 완료되면 여기서 생성된 로컬 파라미터를 파라미터 서버로 전송하게 되며, 파라미터 서버에서는 이를 이용하여 글로벌 파라미터를 생성한다. 그리고 이 글로벌 파라미터를 개별 기업으로 전송하여 다음 학습을 수행하도록 한다.

이렇게 생성된 여러개의 모델을 물리적으로 통합하기 위해 우리가 제안하는 학습 알고리즘은 다음과 같다.

1. create a deep learning model of same structure in each environment and initialize the same weights
- do {
2. Each model is trained to a certain accuracy or higher
3. Create a new model by combining the weight values of the trained model
4. Copy the new created model to each local learning environment
5. If the accuracy of the copied model in each environment is more than the reference value, the training ends
- }

한 번의 epoch가 완성된 후 모든 기업에서 학습한 결과로서 발생한 로컬 파라미터들을 이용해 새로운 글로벌 파라미터를 생성하는 방법은 다음과 같다.

(1) average. 식 (4)과 같이 모델에 존재하는 동일한 위치의 가중치들을 모아 평균을 구한다.

$$w^{i_{avg}} = \frac{w^{i_{m1}} + w^{i_{m2}} + \dots + w^{i_{mk}}}{k} \quad (4)$$

(2) max. 식(5)와 같이 모델에 존재하는 동일한 위치의 가중치들을 모아 최대값을 구한다.

$$w^{i_{max}} = \max(w^{i_{m1}}, w^{i_{m2}}, \dots, w^{i_{mk}}) \quad (5)$$

(3) 식 (6)과 같이 모델에 존재하는 동일한 위치의 가중치들을 모아 절대값의 최대값을 구한다.

$$w^{i_{absmax}} = \max(|w^{i_{m1}}|, |w^{i_{m2}}|, \dots, |w^{i_{mk}}|) \quad (6)$$

여기서 m 은 epoch의 순서를 의미하며, k 는 기업의 수, i 는 가중치의 번호를 의미한다.

즉, 우리의 모델에서는 Fig.2의 파라미터 서버에서 업데이트되는 w' 대신, w_{avg} , w_{max} , w_{absmax} 를 사용한다는 점이 차이점이다.

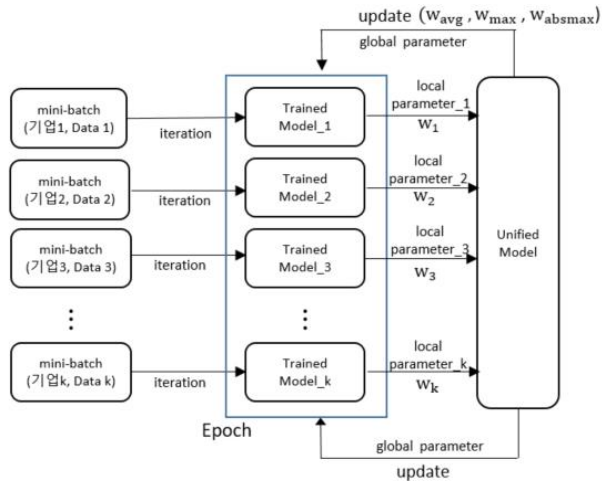


Fig. 3. The Structure of Unified model in data parallel environment.

IV. Experiments and Results

1. Data and Experimental Environment

본 연구에서 실험을 위해 사용한 데이터는 fashion-mnist로서 28×28 크기의 흑백으로 이루어진 이미지로서 t-shirt, top, trouser, pullover 등 10개의 클래스로 구성되어 있으며, 0~9의 레이블이 붙여져 있다.

Table 1. fashion-mnist data set

| Dataset | Train | Test | Total |
|---------------|--------|--------|--------|
| fashion-mnist | 60,000 | 10,000 | 70,000 |

실험을 위해 사용한 컴퓨터는 파라미터 서버로 사용할 컴퓨터 1대와 분산학습을 위해 사용할 컴퓨터 2대로 구성하였고, 시스템의 사양은 3대 모두 다음과 같이 동일하다.

Table 2. The Computer specification used in model training

| CPU | GPU | RAM | Disk |
|----------|-------------|------|---------|
| i7-6500U | RTX-2080 6G | 16GB | SSD 1TB |

2. Process of Experiments

본 연구에서는 실험을 위해 두 가지 방법으로 실험용 데이터셋을 생성하여 실험하였다.

(1) V-환경의 데이터셋을 활용한 실험
실험을 위해 70,000개의 fashion-mnist 데이터를 랜덤하게 섞은 후 학습용 데이터 60,000개와 테스트용 데이터 10,000개로 분리하였다. 그리고 두 대의 컴퓨터에 학습용 데이터 30,000개, 테스트용데이터 5,000개씩 배치하여 학습을 진행하였고, 파라미터 서버에서는 테스트용 데이터 10,000개를 이용하여 테스트를 진행하였다.

(2) H-환경의 데이터셋을 활용한 실험

70,000개의 데이터 중에서 label 0~4를 가진 데이터와 5~9를 가진 데이터를 분리하여 각각 데이터셋을 생성하였으며, 각각의 데이터셋을 랜덤하게 섞은 후 학습용 30,000개, 테스트용 5,000개를 구분하여 학습을 진행하였다. 그리고 각각의 서버에 존재하는 테스트용 이미지 10,000개를 파라미터 서버에 저장하여 테스트를 진행하였다.

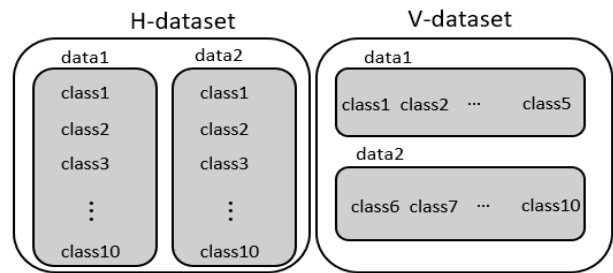


Fig. 4. The Structure of H-dataset and V-dataset in security environment

실험에서 사용한 딥러닝모델은 CNN으로서 딥러닝의 구조는 다음과 같다.

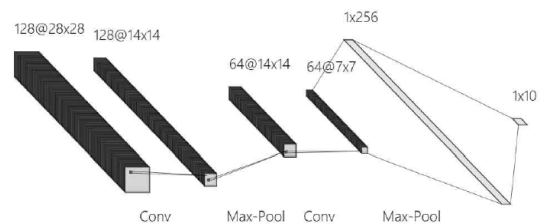


Fig. 5. The layers of deep learning model

또한 적용된 활성화 함수(activation function)는 ReLU를 사용하였다.

학습의 과정은 개별 학습환경에서 각각 30,000장의 이미지를 가지고 학습하도록 하였으며, 이때 mini-batch의 크기는 64로, epoch의 크기는 5,000으로 설정하였다.

개별 학습환경에서 mini-batch에 의해 생성된 파라미

터들을 통합하여 글로벌 파라미터로 생성하기 위한 알고리즘으로서 average, max, absmax를 사용하였다.

3. Result of Experiments

타 연구에서 fashion-mnist 전체 데이터셋을 이용하여 학습한 모델의 정확도(accuracy)는 90%~95%의 범위를 갖는다.[10]

이와 비교하여 본 연구에서 제안한 방법의 결과는 실험 환경에 따라 [Table 3] 및 [Table 4]와 같다.

local accuracy는 각 epoch에서 두 개의 분산환경에서 학습한 결과에 의해 생성된 정확도이며, Unified model accuracy는 식(4), (5), (6)에 의해 글로벌 파라미터를 계산한 통합모델에서의 정확도이다.

V-환경에서의 실험은 average와 absmax에서 우수한 성능을 보였으며, 이는 전체 데이터셋을 이용한 실험의 결과와 비교해도 큰 차이가 없다. 특히 average에서는 통합모델의 정확도가 로컬 모델의 정확도보다 높은 성능을 보였다.

반면 H-환경에서는 통합모델의 정확도가 모두 70%대로서 낮게 나타났다. 이는 서로 상이한 class를 가지고 학습한 로컬 모델을 통합한 후 전체 class 데이터를 이용하여 테스트한 것이어서 어쩌면 당연한 결과라 판단된다. 하지만 avg인 경우 78.68%로서 비교적 높은 정확도를 보였으며 데이터의 class를 서로 다르게 학습한 두 개의 모델을 통합하여 사용할 수 밖에 없는 환경에서는 의미있는 결과라 할 수 있다.

Table 3. Model accuracy result in V-environment

| epoch | local accuracy (trained model 1) | local accuracy (trained model 1) | Unified model accuracy | | |
|-------|----------------------------------|----------------------------------|------------------------|--------|--------|
| | | | avg | max | absmax |
| 1 | 91.41% | 91.51% | 76.13% | 18.28% | 71.10% |
| 2 | 92.43% | 92.13% | 93.00% | 85.32% | 91.53% |
| 3 | 92.46% | 92.96% | 93.44% | 58.03% | 92.50% |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 5,000 | 93.26% | 92.43% | 93.73% | 89.60% | 93.22% |

Table 4. Model accuracy result in H-environment

| epoch | local accuracy (trained model 1) | local accuracy (trained model 1) | Unified model accuracy | | |
|-------|----------------------------------|----------------------------------|------------------------|--------|--------|
| | | | avg | max | absmax |
| 1 | 93.71% | 98.33% | 47.33% | 11.49% | 47.08% |
| 2 | 94.58% | 98.28% | 56.40% | 49.51% | 56.95% |
| 3 | 95.01% | 98.48% | 64.49% | 27.14% | 62.76% |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 5,000 | 95.02% | 98.41% | 78.68% | 71.49% | 70.73% |

V. Conclusions

본 연구에서는 보안 문제로 인해 데이터를 통합할 수 없는 환경에서 제한된 데이터만으로 학습한 결과를 활용하여 통합된 딥러닝 모델을 생성하기 위한 방법을 제안하였다. 연구의 핵심은 딥러닝 모델의 구조는 동일하나 데이터의 소유 주체별로 상이한 데이터를 가지고 있는 경우에 개별적으로 학습한 모델을 어떻게 통합하여 좋은 성능을 낼 수 있도록 하는가에 있다.

본 연구에서는 데이터 보안에 대한 두 가지의 상황으로 문제를 분리하여 실험을 수행하였다. 첫 번째는 데이터의 소유 주체별로 동일한 class의 데이터를 가지고 있으나 통합하기 어려운 상황(V-환경)이고, 두 번째는 데이터의 소유 주체별로 상이한 class의 데이터를 가지고 있으나 통합하기 어려운 상황(H-환경)이다.

우리는 데이터의 소유 주체 별로 로컬 학습을 수행하도록했으며 학습된 결과로 생성된 파라미터를 통합하여 새로운 글로벌 파라미터를 생성한 후 다시 로컬 학습에 사용하도록 했다. 이때 글로벌 파라미터를 생성하기 위한 방법으로 average, max, absmax 방법을 사용하였다.

V-환경에서 개별학습을 수행한 후 통합한 모델은 원래 통합된 데이터 환경에서 학습한 모델의 성능과 유사할 정도로 좋은 성능을 보였다. 특히 average는 개별 환경에서 학습한 모델의 성능보다 더 좋은 성능을 보일 정도로 정확도 면에서 우수했다.

H-환경에서는 통합된 데이터 환경에서의 학습결과에 미치지 못했지만 78.86%라는 의미가 있는 결과를 보여주었다. 특히 개별 기업마다 상이한 class의 데이터만 가지고 있으나 보안문제로 인해 이를 통합하기 어려운 환경에서 개별 데이터만을 가지고 학습한 결과를 가지고 통합모형을 만들어야 하는 상황에서는 의미있는 결과라고 판단된다.

향후 과제는 개별 학습 모델의 파라미터들을 통합하는 과정에서 보다 성능이 좋은 알고리즘을 개발할 필요가 있으며, 이를 이미지 인식 이외의 다양한 문제에서 실험하고 분석하는 것이다.

ACKNOWLEDGEMENT

This work was supported by the Yeungnam University College Research Grants in 2021

REFERENCES

- [1] <https://www.tensorflow.org/deploy/distributed>
- [2] <http://hadoop.apache.org>
- [3] <https://spark.apache.org>
- [4] <https://caffe.berkeleyvision.org/>
- [5] Vishakh Hegde, Sheema Usmani, “Parallel and Distributed Deep Learning”, Proceedings of the 11th Annual conference on Genetic and evolutionary computation, pp. 1569-1576, Jul. 2009, DOI:10.1145/1569901.1570111
- [6] B. Ooi, K. Tan, S.Wang, W. Wang, Q. Cai, G. Chen, J. Gao, Z. Luo, A. Tung, Y. Wang, Z. Xie, M. Zhang, K. Zheng“ A distributed deep learning platform.”, ACM Multimedia, 2015. DOI:doi.org/10.1145/2733373.2807410
- [7] Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015, DOI:10.1007/s11263-015-0816-y
- [8] Kien Nguyen, Arun Ross, “Iris Recognition with Off-the-Shelf CNN Features: A Deep Learning Perspective”, IEEE Access · Dec. 2017, DOI: 10.1109/ACCESS.2017.2784352
- [9] Abien Fred Agarap, “An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification”, Computer Science, Mathematics, 2019, arXiv:1712.03541
- [10] Jason Brownlee, “Deep Learning CNN for Fashion-MNIST clothing classification”, 2019, <https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-fashion-mnist-clothing-classification/>

Authors



Jong-Lark Lee received Ph.D. degree in Statistics from SungKyunKwan University, Korea, in 2012. He is currently an Assisntent Professor in the Faculty of Cyber Security, YeungNam University College.

He is interested in Network Security, Machine Learnig, and Cyber Weapon.