다른 환경에서 학습된 신경망 모델의 통합

이윤호*, 이수항*, 주혜진*, 이종락**,원일용*
*서울호서전문학교 사이버해킹보안과
**영남이공대학교 정보보안과
e-mail : dbsgh3025@gmail.com, op778@daum.net, jennifer0604@naver.com,
jllee@ync.ac.kr, clccclcc@shoseo.ac.kr

Integration of neural network models trained in different environments

Yun-Ho Lee*, Su-Hang Lee*, Hye-Jin Ju*, Jong-lack Lee**, Ill-Young Weon*
*Dept. of Cyber Hacking Security, Seoul Hoseo College
**Dept of Security, YongNam Techincal Colleag

요 약

신경망은 주로 전체 데이터를 중앙에서 학습시키거나 상황에 따라 데이터나 모델을 나누어 분산학습 방법으로 처리해 왔다. 그러나 데이터의 양의 증가와 보안적 이유로 인해 모든 환경에서 기존의 방법을 쓰기에 어려움이 있다. 본 연구에서는 제한된 데이터만으로 모든 데이터로 학습한 것과 같은 학습 효과를 내기 위한 방법을 제안한다. 데이터의 구성이 다른 두 가지 환경인 V-환경과 H-환경에서 학습한 모델을 어떤 방법으로 통합해야 기존의 성능과 비슷한 성능을 낼 수 있는지 연구한다. 우리는 가중치를 합치는 방법을 avg, max, absmas 3가지 방법으로 실험하였으며, 실험 결과로 V-환경에서는 기존의 성능과 비슷한 성능을 보였으며, H-환경에서는 기존의 성능에는 부족하지만, 의미 있는 성능을 보였다.

1. 서론

신경망은 인공지능 분야의 여러 곳에 효과적으로 이용되고 있다. 근래 데이터양의 폭발적인 증가와 컴퓨터 하드웨어의 발전에 힘입어 다양한 분야에서 좋은 성과를 보여 주고 있다.

전통적인 신경망의 학습 방법은 수집할 수 있는 모든 데이터를 중앙에서 수집하여 만족할만한 정확도가 나올 때 까지 학습하는 방법을 주로 사용한다. 데이터의 양이 너무 많아 단독 컴퓨터로 학습할 수 없는 경우는 데이터를 나누거나 모델을 나누는 분산학습 처리 방법으로 발전해 왔다. 특히 데이터 분산은 모든 데이터를 한 컴퓨터가 처리하지 않는 것일 뿐 전체 데이터를 볼 수 있다는 점에서는 1대로 처리하는 것과 동일하다고 할 수 있다.

그러나 보안적 이유로 각각의 컴퓨터가 자신의 제한된 데이터만 볼 수 있는 환경에서는 전체 학습은 기존의 분산학습 방법을 그대로 적용하기는 어려움이 있다.

우리는 이러한 환경에서 제한된 데이터만으로 모든 데이터로 학습한 것과 같은 학습 효과를 내기 위한 방법을 제안한다. 제안의 핵심은 각각의 환경에서 자신의 데이터만으로 학습하고 학습된 모델을 중앙에서 물리적으로 통합하여 다시 각 환경으로 내려보내 개별적으로 검증하는 방법이다. 통합된 모델은 각 환경의 데이터에는 접근하지 못한다. 즉 학습한 모델 웨이트의 물리적 통합이 핵심이다.

본 논문은 2장에 관련 연구, 3장에 모델 통합의 방법론을 제안하였다. 4장에서는 제안한 시스템의 실험 및 결과를 분석하였다. 5장에서는 결론 및 향후 과제를 언급하였다.

2. 관련 연구

2.1 분산학습

딥러닝 모델의 분류 정확도는 훈련예시, 모델 파라미터 수 혹은 둘 다의 증가에 의해 상향될 수 있으며, 전형적인 신경망에서는 모델을 정의하고 학습을 위해 대량의 데이터를 필요로 하는 수많은 매개변수가 있다. 가끔은 데이터셋의 크기 때문에 단일 기계에 저장조차 할 수 없다. [2]

그러나 대형 네트워크 훈련은 많은 비용이 들고, 멀티 스레딩을 지원해도 단일 머신에서의 훈련은 오랜 시간이 걸린다. 따라서 훨씬 더 빨리 실행되고,

훈련 시간을 효과적으로 줄일 수 있는 병렬 및 분산 알고리즘이 필요하다. 분산 방법에는 다음의 2가지 방법이 존재한다. [3]

1) 데이터 병렬화 (Data Parallelism)

데이터가 서로 다른 기계에 분할되고, 각 작업자 노드에 복제된 전체 모델을 사용해 각 기계에서 일부 계산이 수행되는 방식이다. [2] 전체 작업자에서 가중치, 매개변수를 동기화하는 몇 가지 접근법이 있다. 가장 간단한 것은 매개변수 평균화로써 중앙 집중화 된 매개변수 서버의 전역 모델 매개변수를 반복 후 각 작업자의 매개변수 평균으로 설정한다. 이후 훈련을 위해 업데이트된 전역 모델 매개변수를 모든 작업자에게 보낸다. [3]

2) 모델 병렬화 (Model Parallelism)

모델이 너무 커서 단일 머신에 맞지 않으면 여러 머신에 분할한다. 신경망의 다른 노드에 해당하는 연산이 다른 기계에서 수행된다. 모델을 하나의 기계에 맞출 수 없고 훈련 과정을 너무 많이 고정시키지 않는 경우 의존한다. [2] 모델 병렬화 프레임워크에서는 노드 간의 통신 자동 관리, 훈련이나 추론 과정을 동기화할 필요가 있다. [3]

2.2 CNN

CNN(Convolutional Neural Network, 합성곱 신경망)은 다계층 네트워크를 이용하여 인간의 시각 피질 내에 뉴런의 구조와 작동을 기반으로 [4], 딥 피드-포워드 인공신경망의 한 종류이다. [5] 가장 오래된 심층 신경 구조에 속하며 [6] 원래 컴퓨터 비전을 위해 개발되었으나 이후 자연어 처리(NLP)에 효과적인 것으로 나타났다. [7]

CNN은 컨볼루션 레이어(Convolution Layer), 풀링 레이어(Pooling Layer)로 구성된다. 컨볼루션 레이어는 입력 이미지의 너비와 높이를 통해 “슬라이드”하고 입력 영역의 내적과 가중치 학습 매개 변수를 계산하는 데 사용된다. 반면 풀링 레이어는 컨볼루션 필터의 결과에 따라 입력 이미지의 크기를 줄이므로 모델 내의 파라미터의 수도 감소하여 다운 샘플링(Down-Sampling)이라고도 한다. [5]

이미지와 비디오 같은 영상을 처리하도록 고안되었는데, 이미지 특징 표현을 자동으로 학습할 수 있어 기존의 수작업으로 제작된 많은 기술을 능가한다. [4]

3. 학습 모델 통합

망의 구조는 동일하지만 학습 환경이 달라 다른 데이터로 학습한 두 개의 모델을 물리적으로 통합하는 방법으로 우리가 제안하는 학습 알고리즘은 다음과 같다.

1. Create a model of the same structure in each environment and initialize the weight value
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
}

각각의 환경에서 학습이 완료된 모델로 새로운 모델을 만들 때 구조는 동일하고 웨이트 값은 아래의 방법과 같이 통합한다.

average. 식 (1)과 같이 가중치들의 평균을 구해 모델을 하나로 합치는 방법이다.

$$w_{new} = \frac{w_{m_1} + w_{m_2} + \dots + w_{m_n}}{n} \quad (1)$$

max. 가중치들의 최댓값만 뽑아 모델을 하나로 합친다. 식으로 표현하면 식 (2)와 같다.

$$w_{\neq w} = \max(w_{m_1}, w_{m_2}, \dots, w_{m_n}) \quad (2)$$

absmax. 식 (3)과 같이 가중치들에 절댓값을 취한 후 최댓값만 뽑아 모델을 하나로 합친다.

$$w_{\neq w} = \max(|w_{m_1}|, |w_{m_2}|, \dots, |w_{m_n}|) \quad (3)$$

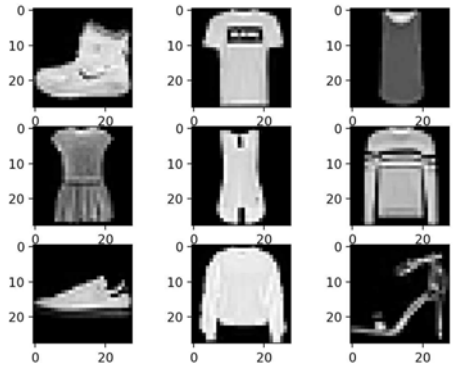
제안하는 모델에서 통합된 신경망은 통합적 모델과 동일한 노드와 웨이트 구조를 가지며, 모든 웨이트 값을 일괄적으로 위의 3가지 방법에 따라 각각 설정한다.

4. 실험 및 결과

4.1 데이터 및 실험 환경

<표 6> dataset

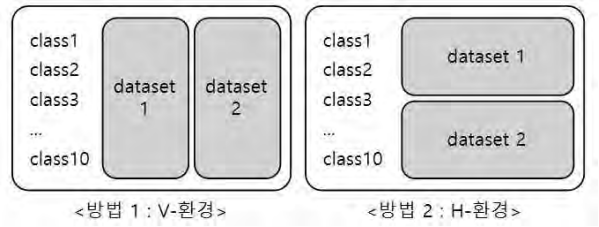
Data set	Train	Test	Total
fashion-mnist	60000	10000	70000



(그림 1) fashion-mnist 데이터셋

사용한 데이터는 fashion-mnist이며, 이 데이터는 0~9의 레이블이 붙여진 t-shirt/top, trouser, pullover 등 10개의 클래스로 이루어져 있다.

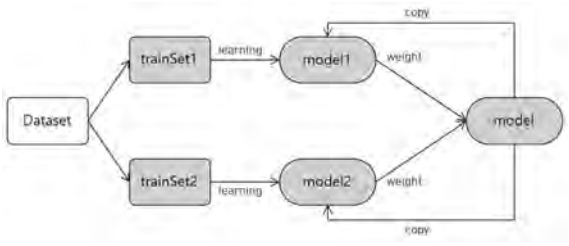
실험의 환경은 그림 2와 같이 2가지로 구분하였는데, 각각의 환경이 모든 클래스를 포함하지만 서로 다른 데이터로 구성되어 있는 환경(V-환경), 각각의 환경이 특정 클래스의 데이터만을 포함하는 환경(H-환경)으로 나누어 진행한다.



(그림 2) 실험 환경

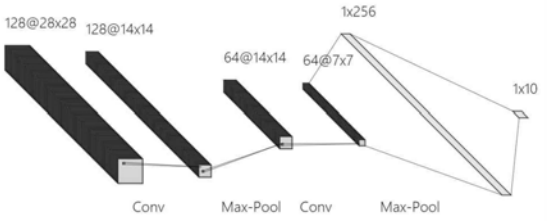
4.2 실험 과정

실험 방법은 그림 3와 같다. 먼저 학습 전, 전체 학습 훈련 셋을 두 그룹으로 나누고, 나눠진 데이터 셋에 대해 각각 학습을 진행한다. 나눠진 학습용 데이터로 각 모델을 학습하고 이후 만들어진 각각의 모델을 3가지 방법으로 통합한 새로운 모델을 만든다. 통합된 모델은 각 환경으로 복사하고 재학습한다. 이 과정을 반복하여 수행한다. 이렇게 만들어진 통합된 모델의 성능은 테스트 데이터로 검증한다.



(그림 3) 실험 흐름

실험에서 사용한 신경망의 구조는 그림 4과 같다.



(그림 4) 모델 구조

4.3 실험 및 분석

타 실험에서 fashion-mnist 전체 데이터셋을 가지고 학습한 모델의 테스트셋 결과는 약 90% ~ 95% 정도의 정확도를 가진다. [8] 이와 비교해서 본 논문에서 제안한 방법에서의 결과는 아래와 같다.

V-환경에서 실험 결과는 표 1과 같다. 표 1에서 지역정확도는 나눠진 데이터셋이 학습되는 각각의 환경에서의 validation 정확도이다. 테스트 정확도는 통합된 모델의 테스트 정확도이다. 테스트 정확도는 웨이트를 합치는 방법에 따라 avg, max, absmax 3가지가 있다.

<표 1> V-환경 실험 결과

회수	지역정확도 (모델1)	지역정확도 (모델2)	테스트 정확도 (avg)	테스트 정확도 (max)	테스트 정확도 (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%	91.96%	93.44%	58.03%	92.50%
...	...	...	...	...	...
n	**93.26%**	**92.43%**	**93.73%**	**89.60%**	**93.22%**
...	...	...	...	...	...

V-환경은 모든 방법이 우수한 성능을 보였다. max 방법은 비교적 저조한 성능을 보였지만, avg로 웨이트를 통합했을 때는 지역 정확도보다 높은 성능을 보였다.

H-환경에서 실험 결과는 표 2과 같다.

<표 2> H-환경 실험 결과

회수	지역정확도 (모델1)	지역정확도 (모델2)	테스트 정확도 (avg)	테스트 정확도 (max)	테스트 정확도 (absmax)
1	93.71%	98.33%	47.33%	11.45%	47.08%
2	94.58%	98.28%	56.40%	49.41%	56.95%
3	95.01%	98.48%	64.49%	27.14%	62.76%
...	...	...	...	...	...
n	**95.2%**	**98.41%**	**78.68%**	**71.49%**	**70.73%**
...	...	...	...	...	...

H-환경은 지역 정확도만큼의 성능이 나오지는 않았지만, 처음 모델을 통합했을 때의 정확도인 11% ~ 47% 정확도보다 약 30%가 성능이 오른 것을 확인할 수 있었다. 특히 avg 방법 같은 경우는 거의 80%에 가까운 성능을 보였다. 이는 최적화 등을 거치면 성능이 더 오를 수 있을 것이라 기대할 수 있다.

V-환경 같은 경우는 기존의 논문들만큼 본 논문에서도 좋은 성능을 보였다. H-환경의 경우는 지역 정확도만큼 정확도가 높지 않았지만 통합 모델의 테스트 성능에서 유의미한 성능을 보였다. 또한 처음의 통합 모델의 테스트 성능은 40%대에 머물렀지만, 반복적으로 학습을 하며 70%대의 성능으로 오른 것을 확인하였다.

5. 결론 및 향후 과제

학습하는 데이터를 통합하지 못하는 보안 환경에서 제한된 데이터만으로 모든 데이터로 학습한 것과 같은 학습 효과를 내기 위한 방법을 제안하였다.

제안의 핵심은 각각의 개별적 환경에서 얻은 데이터로 학습하고, 학습된 모델을 통합하는 방법이다. 모델의 통합은 각각 신경망의 웨이트들의 물리적 연산을 통하여 수행 되었다. 제안된 방법의 성능 검증을 위해 2개의 실험 환경을 나누어서 실험하였다.

실험 결과는 제안한 방법이 어느 정도 의미가 있음을 보여 주었다. 특히, V-환경 실험 결과는 단독 학습과 비슷한 성능을 보였다. H-환경 실험 결과는 의미 있는 성능을 보였지만, 단독 학습과 비교 한다면 아직 더 많은 최적화가 필요해 보인다.

향후 과제는 모델의 물리적인 웨이트 조절을 위한 다양한 연산자를 개발할 필요가 있으며, 다양한 도메인에서 성능을 실험하고 분석하는 것이 필요하다.

참고문헌

[1] Jakub Konecnˇ y´, H. Brendan McMahan, Felix X. Yu, Ananda Theertha Suresh, Dave Bacon & Peter Richtarik “Federated Learning: Strategies for Improving Communication Efficiency”, pp. 1-10, 2017

[2] Vishakh Hegde and Sheema Usmani “Parallel and Distributed Deep Learning”, pp. 1-4, 2016

[3] Coviam Technologies “Distributed Training In Deep Learning Models”, pp. 1-5, 2018

[4] Kien Nguyen, Clinton Fookes, Arun Ross, Sridharan, “Iris Recognition With Off-the-Shelf CNN Features: A Deep Learning Perspective”, IEEE Access, vol.6, pp. 18848-18855, 2017

[5] Abien Fred M. Agarap, “An Architecture Combining Convolutional Neural Network(CNN) and Support Vector Machine (SVM) for Image Classification”, pp.1-4, 2017

[6] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, Dong Yu “Convolutional Neural Networks for Speech Recognition”, IEEE/ACM Trans. Audio Speech Language Processing, vol.22, pp.1533-1545, 2014

[7] Yoon Kim “Convolutional Neural Networks for Sentence Classification”, Empirical Methods in Natural Language Processing (EMNLP), pp. 1746 - 1751, 2014

[8] Jason Brownlee “Deep Learning CNN for Fashion-MNIST Clothing Classification”, 2019