

CNN 형태의 인공지능망 내부 연산 대체에 따른 성능 변화에 대한 연구

서지민, 이정우*
서울대학교

jimin.seo@cml.snu.ac.kr, *junglee@snu.ac.kr

A Study on the performance change from the substitution of operation in the convolutional neural network

Jimin Seo, Jung Woo Lee*
Seoul National University

요약

본 연구에서는 Convolutional neural network (CNN) 구조의 인공지능망에 대해 연산의 종류가 학습 후 최종 성능에 미치는 영향을 분석하기 위하여 인공지능망의 단일 연산을 다른 연산으로 변경하고 학습을 일정 Epoch 만큼 진행하였다. 이와 같이 진행한 후 각 경우의 성능을 계산하였으며, 해당 실험을 통해 연산의 선택에 따른 성능 변화를 예측 가능한지 분석하고자 하였다.

I. 서론

인공지능망 구조 탐색 알고리즘 (Neural Architecture Search, NAS)은 목표하는 데이터셋에 대해 성능이 최적화된 인공지능망을 여러 방법에 따라 찾는 기법이다. 이러한 연구들을 통해 목표하는 데이터셋에 대해 목표 성능 및 지연 시간 측면 등에서 우수한 성능을 내는 것을 가능하게 하나, 우수한 인공지능망 구조를 탐색하는 과정에서 매우 많은 시간을 필요로 하는 단점을 가지고 있다.

정수 최적화 기반의 최적 인공지능망 탐색 알고리즘은 이러한 NAS 기법 중 연구로서, 학습된 인공지능망을 최적화하여 제약조건을 충족하는 가장 최적의 인공지능망을 찾을 수 있도록 하였다. [1] 나아가, 해당 연구에서는 찾아낸 최적의 인공지능망 구조에서 특정 연산들이 선택되지 않은 이유에 대해서 분석하였다.

본 연구에서는 분류 태스크를 수행하는 CNN 구조의 인공지능망에 대해 해당 접근을 적용하는 경우를 가정하여 각 위치에서 다른 후보 연산으로 대체되었을 때 학습을 진행하고, 학습을 마친 상태에서 성능의 변화를 관찰하고자 한다. 이를 통해 연산의 종류에 따른 성능 변화가 최종적인 학습 후 성능에 미치는 영향을 분석하고자 한다.

II. 본론

기존의 NAS 연구에서 연산의 종류가 최종적으로 다른 성능을 나타내는 것은 이야기된 바이며, 기존 NAS 연구에서 언급된 것과 같이 다른 연산을 사용하는 인공지능망이 다른 성능이 나온다는 점에서 명백하다. NAS 연구에서는 이러한 점에서 인공지능망의 성능을 가장 높일 수 있는 적합한 연산들이 있음을 가정한다.

그러나 목표 작업에 대해서 적합한 연산들을 추정하는 것은 어려운 문제이다. 이 문제는 구조와 성능 사이의 관계가 명확하지 않기 때문에 나타나며, 구조에서 성능을 추정하는 것이 어려운 것이 주요한 원인이다.

이러한 관점에서 해결하기 위한 접근 중 하나는 연산 사이의 차이를 비교하는 것이며, 이를 위해 본 연구에서는 인공지능망 내부에서 유사한 연산이 사용되는 경우에 성능이 연속적으로 변화하는 경향을 파악하고자 한다.

이에 따른 실험 방법은 다음과 같다: [1]의 기법을 적용하여 각 후보 연산의 초기 가중치를 얻고, 전체 연산 중 일부 위치의 연산을 각기 다른 후보 연산으로 바꾸어 학습을 진행한다. 이후, 최종적인 Test 성능을 얻어 다른 후보 연산으로 대체한 경우와 비교한다. 실험에서 사용한 후보 데이터셋은 CIFAR10 [2]이며, 적용한 인공지능망은 WiderResNet40x2 [3] 이다.

연구에서 활용한 후보 연산들은 표 1에 작성하였다. 이때, 연산들 중 후보 3, 4, 5, 6, 7은 컨볼루션 연산과 BatchNorm 연산 등으로 구성되어있다는 측면에서 유사성을 가지고 있다. 그러나, 후보 4와 5에서 나타나는 것과 같이 연산의 종류에 따라 더 유사한 경우가 나타난다.

| 후보 연산 | 특성 및 변수 | 관련 연구 |
|-------|---|-------|
| 후보 1 | 기존 네트워크의 연산. (Baseline) | |
| 후보 2 | Identity 연산. 연산 없이 다음 연산으로 입력받은 연산 출력함. | [1] |
| 후보 3 | MobileNet 연산. Inverted Residual 구조. | [4] |
| 후보 4 | Convolution Block 연산. 간단한 Convolution 연산 및 BatchNorm 연산 수행. Expand는 없음. | [1] |
| 후보 5 | Convolution Block 연산. 간단한 | [1] |

| | | |
|------|---|-----|
| | Convolution 연산 및 BatchNorm 연산 수행. Expand 는 2 배. | |
| 후보 6 | ResNet 연산 블록. Expand 는 2 배. | [1] |
| 후보 7 | Bottleneck 구조 연산. Expand 는 2 배 | [1] |

표 1. 본 연구에서 활용한 후보 연산의 설명

네트워크의 성능은 아래의 표와 같이 나타난다. 후보 1 의 경우, 비교를 위해 3 번의 학습 과정에서 발생한 표준편차값을 추가로 작성하였다

| 위치 | 후보 1 | 후보 2 | 후보 3 | 후보 4 | 후보 5 | 후보 6 | 후보 7 |
|----|----------------|-------|-------|-------|-------|-------|-------|
| 1 | | | 92.88 | 94.66 | 94.76 | 93.84 | 93.16 |
| 7 | 96.19 ±0.16 | | 90.64 | 91.98 | 92.12 | 94.25 | 94.40 |
| 18 | | 96.13 | 96.47 | 96.47 | 96.45 | 96.47 | 96.50 |

표 2. 각 위치에서의 후보 연산 적용시 성능 [%]. 25 Epoch 동안 학습하였으며, Test 성능만을 기재.

추가적으로 50 Epoch 까지 학습을 진행한 경우의 성능은 표 3 과 같이 나타난다.

| 위치 | 후보 1 | 후보 2 | 후보 3 | 후보 4 | 후보 5 | 후보 6 | 후보 7 |
|----|----------------|-------|-------|-------|-------|-------|-------|
| 1 | | | 93.28 | 95.30 | 95.10 | 95.46 | 95.49 |
| 7 | 96.12 ±0.06 | | 93.08 | 93.96 | 95.03 | 95.03 | 94.63 |
| 18 | | 95.93 | 96.19 | 96.32 | 96.41 | 96.34 | 96.52 |

표 3. 각 위치에서의 후보 연산 적용시 성능 [%]. 50 Epoch 동안 학습하였으며, Test 성능만을 기재.

표 2 와 3 에서, 위치에 따른 성능 변화를 관찰 가능하다. 특히, 위치 18 에서 후보 2 를 제외하면 성능 향상이 나타난다. 위치 18 에서 후보 3 부터 7 은 후보 1 에 비해 더 많은 수의 매개변수를 사용하는데, 이로 인해 성능 향상이 발생하는 것으로 판단된다. 이에 비해 위치 1 과 7 은 성능 향상은 발생하지 않으나, 성능의 차이라 두드러지게 나타나는 것을 확인 가능하다. 또, 연산의 종류에 따른 성능 변화를 관찰할 수 있다. 위치 1 을 제외하면 후보 4 와 5 사이에서 표준편차 이하의 성능 차이가 나는 것을 확인할 수 있다. 이러한 연산 사이에서 성능 차이가 나지 않는다고 가정할 수 있다.

또한, 학습이 추가적으로 진행되면 위치 1 과 7 에서 변경된 경우 성능 차이가 감소하며, 위치 18 에서 변경된 경우에는 성능 차이가 약간 증가하는 것을 확인할 수 있다. 그러나, 단일 연산 후보에 대해 다른 위치에서의 학습 후의 성능을 관찰하면, 표 2 와 3 사이에서도 다른 경향성을 확인할 수 있다.

위의 관찰을 통해 연산의 변경에 따른 차이가 성능 사이에서 유사하게 나타날 수 있음을 파악할 수 있다. 그러나 성능에 영향을 주는 주요한 요인은 연산의 종류와 연산이 변경된 위치 모두 영향을 준다.

단, 연산 후보들이 변경된 경우에도 성능의 차이가 크지 않은 것은 전체 인공 신경망 중 하나의 연산만 교체하여 발생하는 현상일 수도 있으며, 초기값에 따른 최종 성능의 변동에 의해 나타나는 현상일 수 있다. 또는 연산 종류에 따른 매개변수의 수의 차이가 성능에 영향을 주었을 가능성이 있다.

III. 결론

본 연구에서는 인공신경망 내의 연산들을 다른 연산들로 대체하고, 이때 발생하는 성능의 차이에서

대체하고자 하는 연산의 명시적인 차이가 성능에 미치는 영향을 분석하고자 하였다.

유사한 연산을 사용하는 경우 최종 성능이 유사하게 나타나는 것을 확인할 수 있었으나, 연산의 종류 뿐만 아니라 위치 등의 요인도 영향을 주는 것을 확인할 수 있었으며, 이러한 요인들이 연속적으로 변화하는데 영향을 줄 수 있음을 확인할 수 있었다.

ACKNOWLEDGMENT

This work is in part supported by National R&D Program through the National Research Foundation of Korea (NRF, 2021M3F3A2A02037893), Institute of Information & communications Technology Planning & Evaluation (IITP, 2021-0-00106) grant funded by the Ministry of Science and ICT (MSIT), INMAC, and BK21 FOUR program.

참고 문헌

- [1] Molchanov, P., Hall, J., Yin, H., Kautz, J., Fusi, N., & Vahdat, A. (2022, October). LANA: latency aware network acceleration. In European Conference on Computer Vision (pp. 137-156). Cham: Springer Nature Switzerland.
- [2] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.
- [3] Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. arXiv preprint arXiv:1605.07146.
- [4] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).