Global Weight: 심층 신경망의 압축을 위한 네트워크 수준의 가중치 공유

*신은섭, **배성호
경희대학교
*kairos9603@khu.ac.kr, **shbae@khu.ac.kr

Global Weight: Network Level Weight Sharing for Compression of Deep Neural Network

*Eunseop Shin, **Sung-Ho Bae
Kyung Hee University

요 약

본 논문에서는 큰 크기의 심층 신경망을 압축하기위해 네트워크 수준의 가중치 공유방법인 Global Weight 패러다임을 최초로 제시한다. 기존의 가중치 공유방법은 계층별로 가중치를 공유하는 것이 대부분이었다. Global Weight 는 기존 방법과 달리 전체 네트워크에서 가중치를 공유하는 효율적인 방법이다. 우리는 Global Weight 를 사용하여 학습되는 새로운 컨볼루션 연산인 Global Weight Convolution(GWConv)연산과 GWConv 를 적용한 Global Weight Networks(GWNet)을 제안한다. CIFAR10 데이터셋에서 실험한 결과 2.18 배 압축에서 85.64%, 3.41 배 압축에서 85.46%의 정확도를 보였다. Global Weight 패러다임은 가중치 공유가 궁극적으로 풀고자 했던 중복되는 가중치를 최소화하는 획기적인 방법이며, 추후 심도 있는 연구가 수행될 수 있음을 시사한다.

1. 서론

지난 몇 년간의 연구로 심층 신경망이 고도로 발전되었고, 다양한 영역에서 사용되고 있다. 특히 컴퓨터 비전 분야에서는 컨볼루션 신경망(Convolutional Neural Networks, CNN)을 사용하여 놀라운 성능을 보여왔다[13, 16]. 그러나 대부분의 CNN 은 매개변수가 지나치게 많게 설계되어 있다. 즉, 매우 큰 크기의 매개변수를 포함하기 때문에 모바일 플랫폼과 같은 비교적 제한적인 저장공간, 제한적인 계산 성능을 보이는 장치에서 사용하기에 어려웠다.

이러한 문제를 해결하기 위한 다양한 네트워크 압축 방법이 제시되어 왔다. 가지치기(pruning)[1, 2, 3], 가중치 공유(weight sharing)와 양자화(quantization)[4, 5, 6], 가중치 예측(weight prediction)[7] 등이 그 방법들이며, 실질적으로 모델의 크기를 압축하는데 좋은 해결책이 되어 널리 사용되고 있다.

그 중 가중치 공유방법은 CNN 의 매개변수를 획기적으로 줄일 수 있다고 증명되었다. Deep Compression[4]과 Filter Pruning[1]의 결과를 보면 대부분의 CNN 에서 필터가 중복되어 사용되는 것을 알 수 있다. 마찬가지로 본 논문은 가중치 공유방법을 통해 효율적으로 모델을 압축하는 방법에 집중한다.

기존에도 가중치 공유를 이용하여 모델을 압축하는 연구[8, 9, 10]들이 있었다. 그러나 이 연구들은 모두 레이어 단위의 가중치 공유방법을 사용하였다. 즉, 레이어 안에서만 가중치가 공유되며, 레이어가 다르면 새로운 가중치를 학습해야 한다. 이런 방법은 네트워크 전체적으로 보았을 때 중복성이 여전히 남아있다고 할 수 있다. 따라서 이러한 중복성도 줄이는 것이 네트워크를 더 압축 가능하게 할 것이다.

이 논문에서는 가중치 공유가 네트워크 전체에서 이루어지는 Global Weight 라는 새로운 패러다임을 제시한다. 또한 이를 사용한 새로운 컨볼루션 연산인 Global Weight Convolution(GWConv)을 제안하고 이를 사용한 Global Weight Networks(GWNet)의 압출율과 성능을 실험을 통해 검증한다.

이 논문의 기여는 다음과 같다.

- 최초로 네트워크 수준의 가중치 공유 방법인 Global Weight 를 제안한다.
- Global Weight 를 사용한 효율적인 컨볼루션 연산인 Global Weight Convolution 과 이를 사용한 Global Weight Networks 를 제안한다.
- Global Weight Networks 의 적당한 학습방법을 설명하고 효율성과 성능을 실험으로 증명한다.

본 논문은 2020 년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2018R1C1B30008159)
본 논문은 과학기술정보통신부 및 정보통신산업진흥원의 '고성능 컴퓨팅 지원' 사업으로부터 지원받아 수행하였음

2. 관련 연구

단순히 한 번 정의된 레이어를 여러 번 반복해서 사용하는 네트워크는 이전에도 많이 존재하였다. 반면, 최근 논문들은 단순히 레이어를 반복하는 것이 아닌 한 레이어 안에서 가중치를 효율적으로 공유하는 구조를 새롭게 제안한다.

WSNet[8]에서는 Epitome 방법[10]에서 제안한 겹침 패턴 가중치 공유방식을 확장하여, 1D CNN 을 위한 압축 방법을 제안하였다. 그러나 이 방식은 1D CNN 을 위한 방법이어서 근사화를 거치지 않고는 우리가 일반적으로 사용하는 2D CNN 으로 직접적으로 확장할 수 없다. 이에 FSNet[9]은 2D CNN 에서 사용 가능한 1D 가중치 방식인 Filter Summary(FS)를 제안하였고, 이를 사용한 컨볼루션 연산을 빠르게 수행하기위해 FSFC(Filter Summary Fast Convolution) 방식도 소개했다. 그러나 이 방식은 필터를 재구성할 때 공유되는 위치가 고정적이어서 유연하게 가중치를 공유할 수 없다는 단점이 있다.

NES[9] 또한 Epitome[11]에서 영감을 받아 겹침 패턴 가중치 방법을 2D CNN 에 적용하였다. 이 논문에서는 각 레이어 안에서 공유되는 Epitome Kernel 이 존재하고, 원래의 커널을 재구성하기위해 Epitome Kernel 에서의 위치를 학습하는 Index Learner 를 두어 학습하였다. 그러나 이 방식은 학습시에 추가적인 Index Learner 를 두어야 한다는 단점이 있다.

3. 제안하는 방법

3. 1 Global Weight

기존의 논문들은 모두 레이어 별 가중치 공유방식을 사용하고 있었다. 그러나 이 방식은 레이어 간의 가중치 공유를 할 수 없어 효율성을 낮추게 된다. 따라서 우리는 전체 네트워크에서 공유가능한 가중치인 Global Weight 를 최초로 제안한다. Global Weight 는 전체 네트워크에서 공유가 가능하기 때문에 네트워크에서 사용되는 가중치의 중복성을 획기적으로 줄일 수 있고 그 만큼 네트워크의 매개변수도 효율적으로 압축할 수 있다.

3.2 Global Weight Convolution

Global Weight Convolution(GWConv)은 Global Weight 를 사용한 컨볼루션 연산이다. 이 연산은 Global Weight 로부터 컨볼루션 연산에 사용될 커널의 구성을 학습한다. 기존의 방법들은 Global Weight 로부터 커널을 구성하는데 위치가 고정적[9]이거나, 위치를 학습하는 추가적인 레이어[10]를 두어야 했지만, GWConv 는 동적으로 커널을 구성함에도 불구하고 추가적인 레이어 없이 매우 적은 매개변수만을 추가하여 학습한다. 또한 기존의 컨볼루션 연산을 바로 대체할 수 있는 장점이 있다.

GWConv 의 매개변수: GWConv 에서 사용되는 매개변수는 크게 두개로 볼 수 있다. 하나는 실제 커널의 가중치인 GW, 다른 하나는 커널을 구성하기 위해 사용되는 Selector 이다. GW 는 L의 크기를 갖는 임의의 3D 커널 집합이며, 커널 크기가 K 이고 길이가 L인 GW 의 크기는 $K\times K\times L$이 된다. 예를 들어 3×3 커널을 사용하고 길이가 512 인 경우 3×3×512 의 크기를 갖는다. GW 는 전체 네트워크에서 하나만 존재하며, 모든 레이어에서 공유되어 사용된다. Selector 는 각 레이어에서 개별적인 커널을 구성하기 위해 사용되는 매개변수로 채널 별로 어떤 위치의 커널을 사용할 지를 나타내는 즉, 위치를 학습하는 매개변수이다. GWConv 의 입력 채널을 C_{in}, 출력 채널을 C_{out}이라 했을 때, Selector 는 $C_{in}\times C_{out}$ 크기를 갖는다. 예를 들어 입력 채널이 64 출력 채널이 128 인 경우 64×128 의 크기를 갖는다. 다양한 GW 크기에 대해 대응하기 위해 Selector 는 0~1 사이의 값을 갖도록 학습된 후 GW 의 길이(L)를 곱해 사용된다.

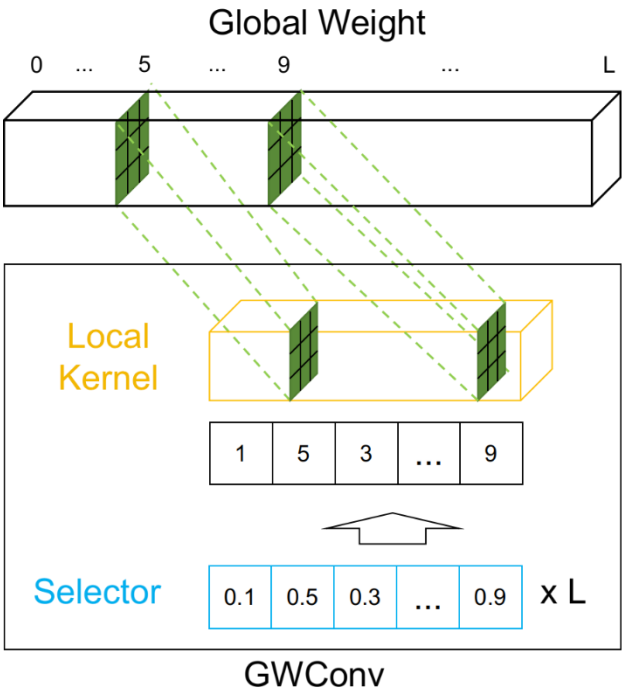

그림 1 GWConv 에서 Local Kernel 을 구성하는 방법. Local Kernel 은 Selector 와 Global Weight 를 사용하여 구성된다. Selector 는 각 채널에서 사용될 Global Weight 의 위치를 상대적으로 학습한다. 학습된 Selector 에 GW 의 크기 L 을 곱하고 반올림하여 선택해야하는 위치를 계산한다. 이를 이용해 GW 에 위치한 커널을 가져와 Local Kernel 을 구성한다.

Local Kernel 선택하는 법: 실제 컨볼루션 연산을 위해서는 컨볼루션 연산에 직접적으로 사용될 Local Kernel 을 생성해야 한다. Local Kernel 은 GW 와 Selector 를 이용해 구성된다. 그림 1 은 Local Kernel 을 구성하는 방법에 대해 설명한다. 먼저 Selector 의 값(0~1)에 GW 의 길이(L)를 곱하고 반올림하여 각 채널에서 사용될 커널의 위치를 결정한다. 이렇게 결정된 커널 위치를 기반으로 GW 에서 해당 위치의 커널을 가져와서 Local Kernel 을 구성한다. 이 과정은 곱셈이나 덧셈 계산을 하는 것이 아니기 때문에 실제 동작시에 매우 빠르게 수행된다.

Selector를 학습하는 방법: GWConv 에서 Selector 는 단순히 위치를 지정하는 연산에만 사용되기 때문에 기울기 계산이 되지 않아 학습할 수 없다. 우리의 목적은 Selector 가 적합한 위치를 찾는 것이기 때문에 Selector 를 학습해야 한다. 이를 위해 우리는 Selector 를 위한 기울기 추정 함수를 추가하여 Selector 가 학습될 수 있도록 하였다. Selector 의 기울기 추정은 [12]에서 score 를 학습하는 것과 같은 방식으로 학습하여 선택이 되지 않은 커널도 다음에 선택할 수 있도록 하였다.

GWConv 의 압축비: GWConv 의 압축비(Compression Ratio)는 GW 의 길이 L 에 달려있다. L 의 크기가 매우 크면 일반적인 컨볼루션을 사용한 네트워크보다 매개변수가 더 많아질 수도 있지만 적당하게 설정하면 매개변수가 충분히 작으면서 성능 하락을 적게 유지할 수 있다. N을 컨볼루션 레이어 수, K를 컨볼루션의 커널 크기, In 을 n 번째 컨볼루션의 입력 채널 수, On 을 출력 채널 수라고 했을 때, 일반적인 컨볼루션의 매개변수 크기는 수식(1)과 같다.

$$\sum_{n}^{N} KI_nO_n \qquad (1)$$

L 을 GW 의 길이로 표시하면, GWConv 의 파라미터 크기는 수식 (2)와 같다.

$$KL + \sum_{n}^{N} I_nO_n \qquad (2)$$

즉, 단일 컨볼루션 레이어의 매개변수는 K 배 만큼 줄어들고 전체 매개변수에 GW 의 크기인 KL만큼 추가된다. 이를 통해 압축비(CR, Compression Ratio)를 구하면 다음과 같다.

$$let) \quad \sum_{n}^{N} I_nO_n = H$$

$$CR = \frac{KH}{KL + H} = \frac{1}{\frac{L}{H} + \frac{1}{K}} \approx \frac{H}{L}$$

여기서 K 는 일반적으로 1, 9, 25 정도로 사용되어 L 과 H 보다 매우 작기 때문에 무시할 수 있다. 즉 GWConv 의 압축비는 기존 컨볼루션의 입출력 채널이 많을 수록, GW 의 길이가 작을수록 많이 압축된다.

3.3 Global Weight Networks

Global Weight Networks(GWNet)은 GW 패러다임을 적용한 네트워크이다. GWConv 모듈은 추가적인 연결이나 학습에서만 사용되는 매개변수가 없기 때문에 기존의 컨볼루션 연산을 바로 대체하여 사용이 가능하다. 따라서 CNN 모델에서 단순하게 기존의 컨볼루션 연산을 GWConv 로 변경한 것을 GWNet 이라고 할 수 있다.

Model	# Params	Top-1	CR
ResNet-20	272,474	92.02%	1
GWNet-10000	124,810	85.64%	2.18
GWNet-7500	102,310	85.73%	2.66
GWNet-5000	79,810	85.46%	3.41
GWNet-2500	57,310	84.36%	4.75
GWNet-1000	43,810	80.96%	6.22
GWNet-500	39,310	78.30%	6.93

표 1 CIFAR10 에서 GW 의 크기 변화에 따른 정확도 변화. GW 의 크기를 다양하게 변화시키며, 압축비(Compression Ratio, CR)와 정확도를 측정하였다.

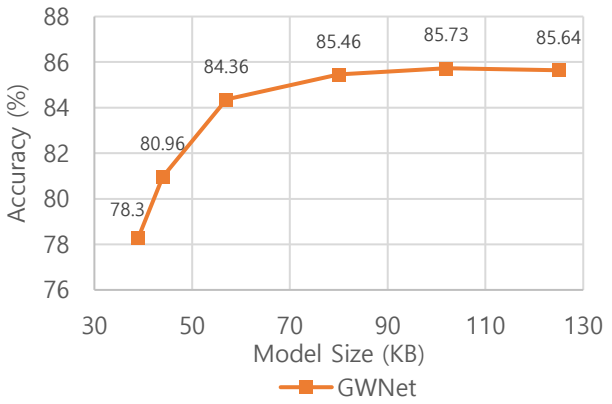

그림 2 CIFAR10 에서 실험한 GWNet 결과. Model Size 별 Accuracy 를 나타낸다. 모델 크기가 60K 이하로 떨어지면 정확도도 급격히 내려가는 것을 볼 수 있다.

4 실험 결과

GWConv 의 효율성과 압축비를 입증하기 위해 이미지 분류 실험을 수행하였다. 사용된 데이터셋은 CIFAR10 데이터셋으로 총 60000 장의 이미지가 있고 그 중 50000 장은 학습 데이터셋 10000 장은 검증 데이터셋으로 구성되어 있다. 이미지의 종류는 10 가지로 각각은 학습 데이터셋 5000 장, 검증 데이터셋 1000 장으로 되어있다[14].

Baseline 모델은 ResNet20[13]을 사용하였다. 모델 전체의 초기 학습율은 0.01 로 하였고 총 300 epochs 학습하였다. 300 epoch 중 50%, 75%에서 학습율을 0.1 씩 곱하면서 학습하였다. selector 의 학습율은 0.001 로 하였고, 30epoch 마다 0.1 씩 곱하면서 학습하였다. Weight Decay 는 전체 모델에 적용하였으며, Selector 에는 적용하지 않았다. Global Weight 는 Kaiming normalization[15]을 사용하여 초기화 하였고, Selector 는 0~1 사이의 랜덤 값으로 uniform distribution 을 이용해서 초기화 했다.

L 의 크기를 500~10000 까지 변화해 가며 실험을 수행하였고 결과는 표 1 에 있다. L 이 10000 일 때 약 2 배 정도 압축됨을 볼 수 있었고 L 을 줄임에 따라 GWNet-500 에서 약 7 배 압축되었다. 정확도는 압축비가 높아질수록 낮아지는 경향을 보였으며, GWNet-10000 의 경우 압축비 2.18 배, 정확도 85.64%로 baseline 인 ResNet-20 과 약 6.4% 정도 낮은 것을 확인할 수 있었다. 또한 정확도는 압축비가 약 2배에서 5배일 때 까지는 80% 중반을 유지하였으나, 그 이후에는 80%이하까지 떨어졌다.

5 결론

이 논문은 심층 신경망의 압축을 위한 가중치 공유방법의 하나인 Global Weight 패러다임을 최초로 제안하였다. Global Weight 는 네트워크 수준의 가중치 공유방법으로 기존의 레이어 수준 가중치 공유방법보다 중복성을 낮출 수 있어서 더 효율적으로 압축할 수 있다. 또한, Global Weight 를 적용한 새로운 컨볼루션 연산인 GWConv 와, 이를 적용한 GWNet 도 제안하였다. GWNet 은 CIFAR10 데이터셋에서 압축비에 따라 2.18 배 압축에서 85.64%, 3.41 배 압축에서 85.46%의 정확도를 보였다. 이는 각각 baseline 보다 6.38%, 6.56% 낮아진 수치이다.

6. 향후 연구

GWConv 의 GW 는 각 가중치 간의 직접적인 상관성이 없다. Epitome[11], WSNet[8], FSNet[9], NES[10]에서 사용된 방법인 겹친 패턴 가중치 공유방식을 적용하여, 가중치를 더 압축시킬 수 있도록 할 것이다. 또한 가중치 증강의 일종으로 GW 를 90, 180, 270 도 회전시키는 방식의 가중치 증강을 시도하여 약 4 배 더 압축할 수 있는 방식을 적용할 것이다. 또한 양자화 방법을 추가적으로 적용하여 정확도는 떨어지지 않으면서 압축비를 높이는 방법을 연구할 것이다.

참 고 문 헌

[1] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In IEEE International Conference on Computer Vision, ICCV, Venice, Italy, 2017.

[2] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[3] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017.

[4] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In Proceedings of the International Conference on Learning Representations (ICLR), 2016.

[5] Frederick Tung and Greg Mori. Clip-q: Deep network compression learning by in-parallel pruning-quantization. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[6] E. Park, J. Ahn, and S. Yoo. Weighted-entropy-based quantization for deep neural networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7197-7205, July 2017.

[7] Kang-Ho Lee, JoonHyun Jeong, Sung-Ho Bae. An Inter-Layer Weight Prediction and Quantization for Deep Neural Networks based on a Smoothly Varying Weight Hypothesis. arXiv preprint arXiv: 1907.06835, 2019.

[8] Jin, Xiaojie, Yingzhen Yang, Ning Xu, Jianchao Yang, Jiashi Feng and Shuicheng Yan. WSNet: Compact and Efficient Networks with Weight Sampling. In International Conference on Learning Representations Workshop(ICLR Workshop), 2018.

[9] Yang, Yingzhen, Nebojsa Jojic and Jun Huan. FSNet: Compression of Deep Convolutional Neural Networks by Filter Summary In International Conference on Learning Representations (ICLR), 2020.

[10] Daquan Zhou, Xiaojie Jin, Qibin Hou, Kaixin Wang, Jianchao Yang and Jiashi Feng. Neural Epitome Search for Architecture-Agnostic Network Compression. In International Conference on Learning Representations (ICLR), 2020.

[11] Nebojsa Jojic, Brendan J. Frey, and Anitha Kannan. Epitomic analysis of appearance and shape. In 9th IEEE International Conference on Computer Vision ICCV, Nice, France, pp. 34-43, 2003.

[12] V. Ramanujan, M. Wortsman, A. Kembhavi, A. Farhadi, and M. Rastegari. What's hidden in a randomly weighted neural network? arXiv preprint arXiv:1911.13299, 2019.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, June 2016.

[14] Krizhevsky, Alex, and Geoffrey Hinton. Learning multiple layers of features from tiny images. Master's thesis, University of Tront, 2009.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). IEEE Computer Society, USA, 1026-1034. 2015.

[16] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708). 2017.