동적 필터 프루닝 기법을 이용한 심층 신경망 압축

*조인천, **배성호
경희대학교
*dlscjs5362@gmail.com, **shbae@khu.ac.kr

Dynamic Filter Pruning for Compression of Deep Neural Network.

*InCheon Cho, **SungHo Bae
Kyung Hee University

요 약

최근 이미지 분류의 성능 향상을 위해 깊은 레이어와 넓은 채널을 가지는 모델들이 제안되어져 왔다. 높은 분류 정확도를 보이는 모델을 제안하는 것은 과한 컴퓨팅 파워와 계산시간을 요구한다. 본 논문에서는 이미지 분류 기법에서 사용되는 딥 뉴럴 네트워크 모델에 있어, 프루닝 방법을 통해 상대적으로 불필요한 가중치를 제거함과 동시에 분류 정확도 하락을 최소로 하는 동적 필터 프루닝 방법을 제시한다. 원샷 프루닝 기법, 정적 필터 프루닝 기법과 다르게 제거된 가중치에 대해서 소생 기회를 제공함으로써 더 좋은 성능을 보인다. 또한, 재학습이 필요하지 않기 때문에 빠른 계산 속도와 적은 컴퓨팅 파워를 보장한다. ResNet20 에서 CIFAR10 데이터셋에 대하여 실험한 결과 약 50%의 압축률에도 88.74%의 분류 정확도를 보였다.

1. 서론

콘볼루션 뉴럴 네트워크(Convolutional Neural Networks : CNNs)는 컴퓨터 비전 분야의 이미지 분류, 디텍션(detection), 세그멘테이션(segmentation) 분야에서 성공적인 업적을 달성했다.[8, 15] 하지만 CNNs 의 정확도를 높이기 위해서는 컴퓨팅 파워와 거대한 메모리 사용이 요구된다. 따라서 스마트폰, 웨어러블 디바이스와 같은 상대적으로 컴퓨팅 파워가 약한 기기에서는 딥러닝 모듈을 작동시키기에 어려움이 있다. 최근 정확도 성능은 유지하면서 FLOPs 값과 CNNs 의 사이즈를 줄이는 연구가 진행되어져 왔다. 그 중에서도 활발하게 연구가 진행된 내용은 필터 압축[16], 양자화[17], 네트워크 프루닝 기법[1, 2, 3, 4, 5, 9, 14]이다.

그 중에서도 네트워크 프루닝 기법은 다양한 측면에서 쉽게 적용될 수 있었다. 네트워크 프루닝 기법은 크게 가중치를 제거하여 네트워크를 프루닝하는 웨이트 프루닝 기법과 필터 자체를 제거하여 네트워크를 프루닝하는 필터 프루닝 기법으로 나뉜다. 웨이트 프루닝 기법은 가중치 자체의 크기를 기준으로 작은 값을 제거하거나[6], 가중치 크기 이외에도 기울기나 다른 요소들을 기준으로[2, 3, 9, 10] 가중치를 제거한다. 필터 프루닝 기법은 필터를 구성하고 있는 모든 가중치를 제거한다. 필터를 구성하고 있는 가중치의 합을 기준으로 낮은 필터를 제거하는 방법[7]과 피쳐맵과 같은 다른 기준으로 필터를 제거하는 방법이 있다.[4]

웨이트 프루닝 같은 경우에는 FLOPs 값을 줄이며 성능을 유지하는 데 큰 도움을 준다. 하지만 특화된 소프트웨어[11]와

-이 논문은 2020 년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2018R1C1B3008159)
- 본 논문은 과학기술정보통신부 및 정보통신산업진흥원의 '고성능 컴퓨팅 지원' 사업으로부터 지원받아 수행하였음

하드웨어가 존재하지 않는 이상 빠른 연산 속도를 보장할 수 없다. 실제로 CNNs 동작을 위해 사용하는 GPU 에서, 웨이트 프루닝은 연산 속도 차이가 없는 것을 확인할 수 있다. 반면에, 필터 프루닝 같은 경우에는 필터 자체를 제거하기 때문에 제약이 적고, 또한 행렬연산 수를 감소시키기 때문에 빠르다. 따라서 감소한 FLOPs 값 만큼 가속화가 이뤄진다. 본 논문은 CNNs 모델을 압축(가중치의 개수를 줄임)하면서 가속화(FLOPs 값을 줄임)를 위해 동적 필터 프루닝 방법을 제시한다.

본 논문의 구성은 다음과 같다. 2 절에서는 관련연구를 살펴본 다음, 3 절에서는 동적 필터 프루닝 기법에 대해서 설명한다. 4 절에서는 본 논문에서 제안하는 방법의 실험 성능을 제시하고, 5 절에서는 본 논문에 대한 결론을 맺고 6 절에서 추후 연구에 대한 방향을 제시한다.

본 논문의 기여는 다음과 같다.

- 모든 가중치들을 사용하면서 최초로 동적 필터 프루닝의 방법을 제안했다.
- 필터 프루닝 방법에서 Normalization term 을 적절히 사용하였고, 사용 이유에 대한 근거를 제시하였다.
- ResNet20 모델에 대하여 CIFAR10 데이터셋을 학습시키는 실험을 진행하였고, 약 50%의 압축률에도 88.74%의 분류 정확도를 보였다.

2. 관련 연구

웨이트 프루닝. 가중치 값을 기준으로 프루닝을 진행한다. 가중치 값을 제거하기 때문에 큰 압축률과 높은 분류 정확도 성능을 확보할 수 있다. 하지만 특별한 소프트웨어와 하드웨어가 존재하지 않으면 계산 속도에서의 향상을 확인하기 어렵다.

채널 프루닝. 가중치보다 큰 값인 필터를 구성하고 있는 채널 단위로 프루닝을 진행한다. 콘볼루션 연산의 커널 사이즈에 따라서 그 크기가 달라진다.

필터 프루닝. 가중치를 구성하고 있는 필터 전체를 제거하는 방법이다. 웨이트 프루닝과 다르게 실제 압축률이 빠른 계산 속도를 보장한다. Figure(3)을 참고하여 각각의 프루닝이 어떤 단위로 처리되는지를 확인할 수 있다.

학습 이후 프루닝. 모든 가중치가 온전히 존재하는 CNNs 모델을 학습시킨 후에 그 모델을 기준으로 프루닝을 진행하는 기법이다. 학습된 모델에 원하는 압축률을 기준으로 프루닝 기법을 적용한 후에 축소된 모델을 다시 낮은 학습률로 학습시킨다.

학습 중 프루닝. 모델이 학습을 진행하면서 원하는 압축률을 얻기 위해 점차적으로 가중치를 제거하는 방법이다. 크게 정적인 방법과 동적인 방법으로 나뉘는데 정적인 방법의 경우 제거된 가중치는 학습 끝까지 제거된 상태로 유지된다. 반면에 동적인 방법은 제거된 가중치에 다시 학습의 기회를 줌으로써 좀더 안정적인 학습을 가능하게 한다.

학습 전 프루닝. 모델을 학습하기 전에 미리 프루닝 기법을 적용하여 목표 압축률을 만족시키는 모델을 생성한다. 이 방법은 기울기[10], 기울기의 변화량[3]을 기준으로 가중치를 제거하는 방법이 대표적이다. 학습 전 프루닝 같은 경우에는 학습전에 가중치를 제거하는 방법이기 때문에 매우 도전적인 연구이다.

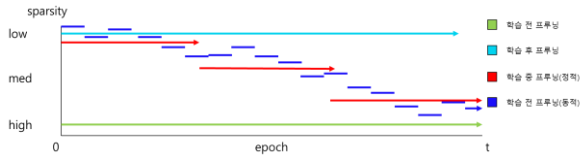

Figure 1. 프루닝 종류에 따른 학습 중 모델 내부의 Sparsity 변화량

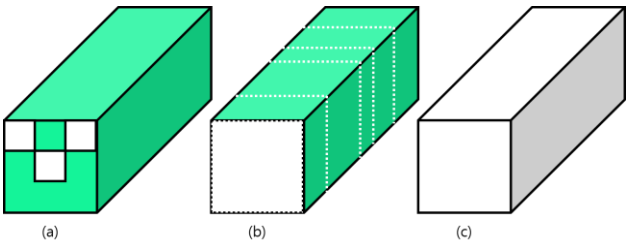

Figure 2. (a)는 웨이트 프루닝, (b)는 채널 프루닝, (c)는 필터 프루닝을 나타낸 그림이다. 가중치에서 잘라낼 크기 단위에 따라서 프루닝 방법이 달라진다.

3. 동적 필터 프루닝

(1) 동적 웨이트 프루닝

가중치의 절댓값을 기준으로 아래와 같은 식으로 가중치에 마스크를 형성한다. 생성된 마스크는 가중치에 곱해져 압축된 가중치를 제공한다.

$$m_t = \begin{cases} 1 & (|w_t| \geq threshold) \\ 0 & (|w_t| < threshold) \end{cases} \quad (1)$$

$$w_{t+1} = w_t - \gamma_t g(m_t \odot w_t) = w_t - \gamma_t g(\widetilde{w_t}) \quad (2)$$

동적 웨이트 프루닝은 Eq(1)를 기준으로 처리된다. g 는 gradient 를 가리키는 문자로 기울기를 말하고 lambda 는 학습률이다. t 는 epoch 를 나타낸다. 동적 웨이트 프루닝은 프루닝 방법이 적용되지 않은 모델과 프루닝 방법이 적용된 모델 2 개를 가지고 학습을 진행한다. 따라서 모든 가중치 값들을 사용할 수 있는

장점이 있다. 이미 제거된 값들을 임의의 랜덤한 값으로 추가하지 않아도 되는 편리함도 동시에 가지고 있다. 동적 웨이트 프루닝 같은 경우에는 좋은 성능을 보이지만 실제적으로 계산 속도 향상에 있어서는 한계를 보인다. 따라서 필터적으로 네트워크 프루닝을 진행함으로써 실제적으로 가속화된 압축 모델을 얻을 수 있다.

(2) 동적 필터 프루닝

웨이트 프루닝과 다르게 동적 필터 프루닝은 가중치가 이루고 있는 필터 전체를 고려해야한다. 기존의 동적 필터 프루닝[1]은 L2 norm 크기를 기반으로 필터 자체의 크기만을 고려하는 방법은 압축률이 커질 때에 Figure3 와 같은 결과를 가져오게 한다. 즉, 특정 레이어의 가중치들이 모두 제거되는 현상이 발생한다. 따라서 L1 norm 의 크기와 Min-Max Scaling Eq(4)을 통해 더 안정적으로 필터 프루닝 방법을 적용할 수 있다.

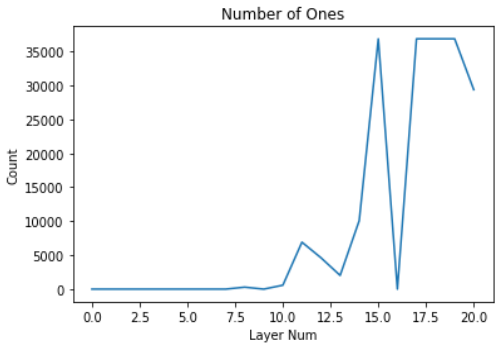

Figure 3. Normalization 과정이 없이 값 자체로 가중치를 지우면 특정 레이어의 모든 가중치가 0 으로 되어버리는 현상이 발생한다. 이러한 현상으로 인해 학습이 제대로 진행되지 않는 현상이 발생한다.

$$M_{i,j,t}\begin{cases}1 & (S(\|W_{i,j,t}\|) \geq threshold) \\ 0 & (S(\|W_{i,j,t}\|) \geq threshold)\end{cases} \quad (3)$$

i, j, t 는 각각 레이어 인덱스, 필터 인덱스, epoch 값을 나타낸다.

$$\|W\| = \sum_{k_3=0}^{n}\sum_{k_2=0}^{w}\sum_{k_1=0}^{h}|w_{k_1,k_2,k_3}|$$

편의상 i, j, t 는 생략하겠다. n, w, h 값은 순서대로 필터의 채널개수, 너비, 높이를 나타낸다.

$$S(x) = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (4)$$

$$W_{t+1} = W_t - \gamma_t g(M_t \odot W_t) = W_t - \gamma_t g(\widetilde{W}_t) \quad (5)$$

Eq(3)을 통해 얻어진 마스크를 가중치에 곱함으로써 압축된 모델을 얻을 수 있다. 동적 필터 프루닝은 Eq(5)을 기준으로 처리된다. 웨이트와 마스크의 차원은 $W_{i,j,t}, M_{i,j,t} \in \mathbb{R}^{in_channels \times k \times k}$로 표현된다.

4. 실험 결과

실험환경은 다음과 같다. 모델은 ResNet20 이용하였고, CIFAR10 의 데이터셋을 이용하여 분류 정확도 성능을 측정하였다. 학습 후 프루닝, 정적/동적 학습 중 프루닝의 성능을 비교하였다. 학습은 300 epoch 까지만 진행했고, 하이퍼 파라미터는 초기 학습률을 0.2 로 시작하고 150epoch, 225 epoch 에서 10 배씩 감소시켰다. Stochastic Gradient Descent 에 Neterov 모멘텀을 추가시켜서 학습시켰다.

Sparsity 의 값에 따른 정확도 성능도 측정해보았다. Table1. Figure4.의 결과를 통해서 알 수 있듯이, 동적 필터 프루닝이 정적 필터 프루닝 보다 적은 가중치로 높은 분류 정확도 성능을 보이고 있다.

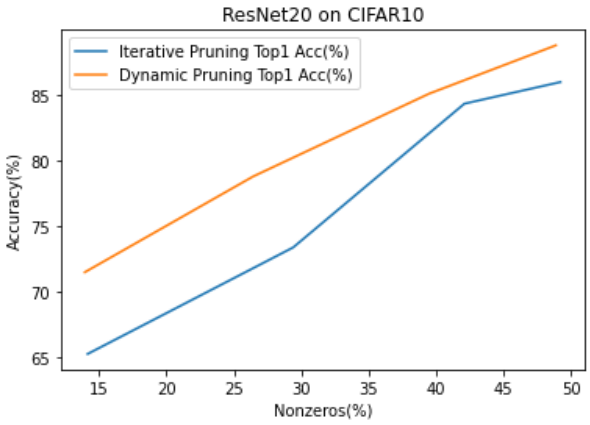

Figure 4. Sparsity 변화에 따른 정확도. Nonzeros(%)는 콘볼루션 연산을 기준으로 0 이 아닌 가중치의 비율을 나타낸다. Nonzeros(%)의 값이 작을수록 더 압축률이 큰 모델이다.

Table 1. Sparsity 변화에 따른 정확도. 모델 옆의 소수점은 target-sparsity 를 나타낸다.

	Top1 Acc	Nonzeros(%)
Baseline	91.25	0.00
Iterative Pruning0.5	85.94	49.23
Dynamic Pruning0.5	**88.74**	**48.89**
Iterative Pruning0.6	84.29	42.10
Dynamic Pruning0.6	**85.05**	**39.48**
Iterative Pruning0.7	73.33	29.40

Dynamic Pruning0.7	**78.74**	**26.42**
Iterative Pruning0.8	65.21	14.15
Dynamic Pruning0.8	**71.44**	**13.93**

5. 결론

본 논문은 기존의 동적 필터 프루닝에 대한 분석과 제거되는 가중치의 결과를 확인했다. 기존의 동적 필터 프루닝과 다르게 L1 norm, Min-Max Normalization 을 추가하였고, 그에 대한 근거를 적절히 제시하였다. 동적 필터 프루닝은 GPU 에서 계산 속도가 향상되는 장점을 가지고 있기 때문에 웨이트 프루닝보다 제약이 적어 실제적인 적용이 용이하다. 제안된 동적 필터 프루닝은 학습에 대해서 더 안정적이며, 적은 가중치를 가지고도 최적의 정확도를 가진 모델을 생성해낼 수 있다.

6. 추후 연구

필터 프루닝은 필터 정규화 방법에 따라 성능 결과가 상이할 것으로 예상된다. 필터의 중요도를 더 정확하게 파악할 수 있는 정규화 방법을 고안하는 연구를 진행한다면 더 안정적이고 효과적인 모델 압축이 가능할 것이다.

동적 필터 프루닝 학습이 진행될 때 동적 웨이트 프루닝과 다르게 몇몇 웨이트의 그래디언트가 사라지는 현상이 발생했다. 이를 보완하고자 마스크에 확률을 추가하여 필터의 소생기회를 주는 연구를 진행해볼 계획이다.

참 고 문 헌

[1] Tao Lin, Sebatian U. Stich, Luis Barba, Daniil Dmitriev and Martin Jaggi. Dynamic model pruning with feedback. In *ICLR - International Coference on Learning Representations*, 2020.

[2] Junjie LIU, Zhe XU, Runbin SHI, Ray C. C. Cheung and Hayden K.H. So. Dynamic sparse training: Find efficient sparse network from scratch with trainable masked layers. In *ICLR - International Coference on Learning Representations*, 2020.

[3] Chaoqi Wang, Guodong Zhang and Roger Grosse. Picking Winning Tickets Before Training by Preserving Gradient Flow. In *ICLR - International Coference on Learning Representations*, 2020.

[4] Mingbao Lin, Rongrong Ji, Yan Wang, Yichen Zhang, Baochang Zhang, Yonghong Tian and Ling Shao. HRank: Filter Pruning using High-Rank Feature Map. In *CVPR - Computer Vision and Pattern Recognition,* 2020.

[5] Shaopeng Guo, Yujie Wang, Quanquan Li and Junjie Yan. DMCP: Differentiable Markov Channel Pruning for Neural Networks. In *CVPR - Computer Vision and Pattern Recognition,* 2020.

[6] Song Han, Huizi Mao and William J. Dally. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In *ICLR - International Coference on Learning Representations*, 2016.

[7] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu and Yi Yang. Filter Pruning via Geometric Median for Deep Convolutional Neural Networks Acceleration. 2019.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. Deep residual learning for image recognition. 2015.

[9] Sejun Park, Jaeho Lee, Sangwoo Mo and Jinwoo Shin. Lookahead: A Far-Sighted Alternative of Magnitude-based Pruning. In *ICLR - International Coference on Learning Representations*, 2020.

[10] Namhoon Lee, Thalaiyasingam Ajanthan and Philip H. S. Torr. SNIP: Single-Shot Network Pruning Based on connection Sensitivity. In *ICLR - International Coference on Learning Representations*, 2019.

[11] Jongsoo Park, Sheng Li, Wei Wen, Ping Tak Peter Tang, Hai Li, Yiran Chen and Pradeep Dubey. Faster CNNs with Direct Sparse Convolutions and Guided Pruning. 2016.

[12] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A. Horowitz and William J. Dally. Eie: Efficient inference engine on compressed deep neural network. 2016.

[13] Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li and Ruosong Wang. Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks. 2019.

[14] Jonathan Frankle and Michael Carbin. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural

Networks. In *ICLR – International Coference on Learning Representations*, 2019.

[15] 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. 2017.

[16] Seyed Mehdi Ayyoubzadeh and Xiaolin Wu. Filter Bank Regularization of Convolutional Neural Networks. 2019.

[17] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam and Dmitry Kalenichenko. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. 2017.