KICSP 한국 융합신호처리학회

융합신호처리학회논문지 Vol. 22, No. 3 : 99-103 September. 2021

Deep Compression의 프루닝 문턱값 동적 조정

이여진[1], 박한훈[1*]

[1]부경대학교 전자공학과

Dynamic Adjustment of the Pruning Threshold in Deep Compression

Yeojin Lee[1], Hanhoon Park[*]

[1]Department of Electronic Engineering, Pukyong National University

요 약 최근 CNN(Convolutional Neural Network)이 다양한 컴퓨터 비전 분야에서 우수한 성능으로 널리 사용되고 있다. 그러나 CNN은 계산 집약적이고 많은 메모리가 요구되어 한정적인 하드웨어 자원을 가지는 모바일이나 IoT(Internet of Things) 기기에 적용하기 어렵다. 이런 한계를 해결하기 위해, 기존의 학습된 모델의 성능을 최대한 유지하며 네트워크의 크기를 줄이는 인공신경망 경량화 연구가 진행되고 있다. 본 논문은 신경망 압축 기술 중 하나인 프루닝(Pruning)의 문턱값을 동적으로 조정하는 CNN 압축 기법을 제안한다. 프루닝될 가중치를 결정하는 문턱값을 실험적, 경험적으로 정하는 기존의 기술과 달리 정확도의 저하를 방지하는 최적의 문턱값을 동적으로 찾을 수 있으며, 경량화된 신경망을 얻는 시간을 단축할 수 있다. 제안 기법의 성능 검증을 위해 MNIST 데이터 셋을 사용하여 LeNet을 훈련시켰으며, 정확도 손실 없이 약 1.3 ~ 3배의 시간을 단축하여 경량화된 LeNet을 얻을 수 있었다.

• 주제어 : CNN, 신경망 경량화, 가지치기, 문턱값 조정, LeNet

Abstract Recently, convolutional neural networks (CNNs) have been widely utilized due to their outstanding performance in various computer vision fields. However, due to their computational-intensive and high memory requirements, it is difficult to deploy CNNs on hardware platforms that have limited resources, such as mobile devices and IoT devices. To address these limitations, a neural network compression research is underway to reduce the size of neural networks while maintaining their performance. This paper proposes a CNN compression technique that dynamically adjusts the thresholds of pruning, one of the neural network compression techniques. Unlike the conventional pruning that experimentally or heuristically sets the thresholds that determine the weights to be pruned, the proposed technique can dynamically find the optimal thresholds that prevent accuracy degradation and output the light-weight neural network in less time. To validate the performance of the proposed technique, the LeNet was trained using the MNIST dataset and the light-weight LeNet could be automatically obtained 1.3 to 3 times faster without loss of accuracy.

• Key Words : CNN, Neural network compression, Pruning, Threshold adjustment, LeNet

Received 1 September 2021, Revised 29 September 2021, Accepted 30 September 2021

*** Corresponding Author** Hanhoon Park, Department of Electronic Engineering, Pukyong National University, 45, Yongso-ro, Nam-gu, Busan, Korea. E-mail: hanhoon.park@pknu.ac.kr

Ⅰ. 서론

인공신경망의 한 종류인 CNN(Convolutional Neural Network)은 최근 다양한 컴퓨터 비전 분야에서 우수한 성능으로 널리 사용되고 있다[1-2]. 일반적으로, CNN은 네트워크가 깊고 넓을수록 높은 정확도를 얻는 경향을 보인다. 정확도 향상을 위해, CNN의 크기와 파라미터 수는 점차 커졌고 이에 따라 요구되는 메모리와 연산량 또한 크게 증가하였다. 이로 인해 우수한 성능을 가진 대부분의 CNN은 한정적인 하드웨어 자원인 모바일 또는 IoT 기기에서 사용할 수 없다. 이런 한계를 해결하기 위해, 기존의 학습된 CNN 모델의 성능을 최대한 유지하며 네트워크의 크기를 줄이는 인공신경망 경량화 연구가 활발히 진행되고 있다. 경량화 연구는 크게 모델 압축, 지식 증류, 하드웨어 가속화, 모델 압축 자동 탐색으로 나눌 수 있으며, 모델 압축 기술은 가중치 프루닝(pruning), 양자화(quantization), 공유(sharing) 등의 방법으로 작은 값을 가지는 가중치(weight)를 없애고, 적은 비트 수로 가중치를 표현하고, 유사한 가중치를 동일한 값으로 표현함으로써 모델의 크기를 크게 줄일 수 있다[3].

본 논문에서는 모델 압축 기술 중 프루닝의 성능을 좌우하는 문턱값(threshold) 변수를 동적으로 조정하는 CNN 압축 기법을 제안한다.

Ⅱ. 신경망 프루닝 기법

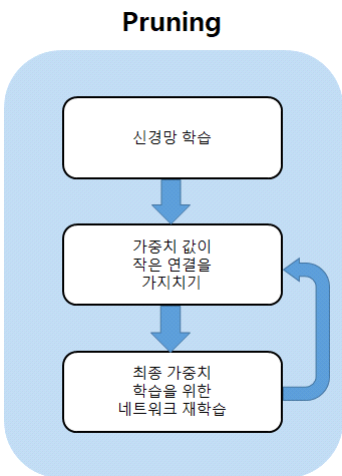


Fig. 1. Process flow of pruning

신경망 프루닝은 CNN 모델의 복잡함을 줄이고 과적합(over-fitting)을 막기 위한 유용한 기법이다. 그림 1과 같이, 먼저 일반적인 신경망 학습을 통해 가중치를 학습한다. 다음으로 값이 작은 가중치를 프루닝하는데, 이때 문턱값보다 낮은 값을 가지는 가중치를 제거한다(즉, 값을 0으로 변경). 마지막으로, 최종 가중치를 학습하기 위해 네트워크를 재학습한다[4]. 프루닝과 재학습을 반복함으로써, CNN 모델의 크기는 줄이면서 높은 정확도를 유지 또는 개선할 수 있다. 이 과정을 통해 AlexNet과 VGG-16 모델의 파라미터(가중치)를 9 ~ 13배 줄일 수 있다[5-6].

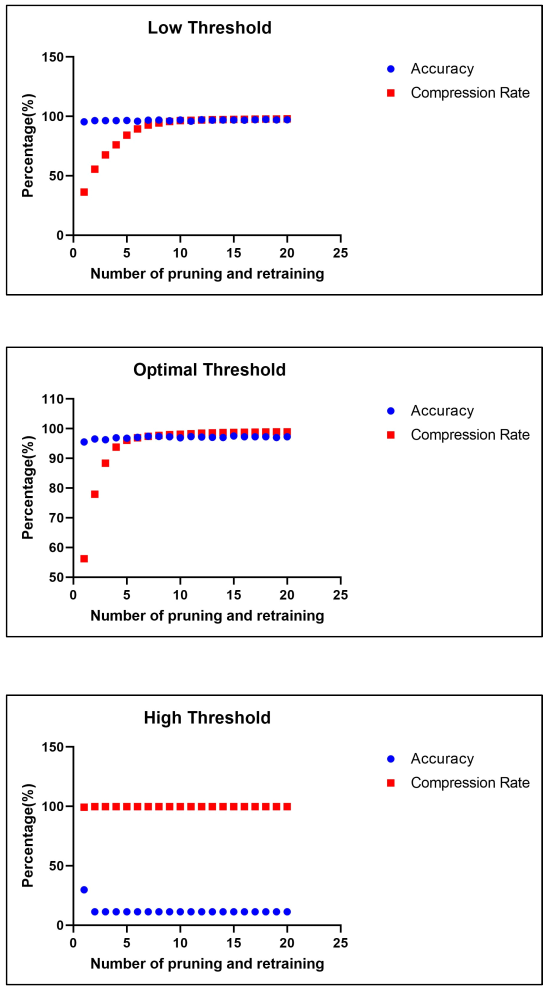


Fig. 2. Accuracy and compression rates of pruning with different threshold values

Ⅲ. 제안 기법

앞 절에서 설명한 신경망 프루닝 기법은 프루닝-재학습의 반복 과정에서 문턱값은 모델 압축률, 정확도 손실, 학습 시간 등에 큰 영향을 끼친다. 즉, 우수한 성능을 위해서는, 문턱값을 낮게 설정하여 정확도 손실을 최소화하는 것이 이상적이다. 그러나 그림 2와 같이 문턱값을 낮게 설정하면 원하는 압축률을 얻기까지의 프루닝-재학습 과정의 횟수가 커진다는 단점이 존재한다(즉, 원하는 압축 모델을 얻는 데 오랜 시간이 걸린다). 반대로 문턱값을 높게 설정하면, 압축률이 높은 모델을 빨리 얻을 수 있으나 정확도 손실이 커져 성능이 떨어진다. 따라서 높은 정확도와 압축률을 동시에 얻기 위해서 최적의 문턱값을 찾는 것이 중요하다. 그러나 기존의 프루닝 기법에서는 고정된 문턱값을 경험적으로 설정하기 때문에, 모델에 따라 원하는 압축 모델을 얻는 시간이 오래 걸리거나 정확도가 떨어진 압축 모델을 얻을 수도 있다. 본 논문에서는 이러한 문제를 해결하기 위하여 동적으로 최적의 문턱값을 찾아 모델을 압축하는 알고리즘을 제안한다.

제안 알고리즘은 알고리즘 1과 같다. s는 문턱값 조정 인자로서 문턱값(th)은 가중치의 표준편차와 s를 곱하여 얻어진다(기존 신경망 프루밍 알고리즘에서는 프루닝-재학습 과정 동안 s가 고정되었다). 기본적으로 s는 높은 값으로 초기화하여 시작한다. 이는 원하는 압축률을 가진 모델을 얻는데 필요한 프루닝-재학습 과정의 횟수를 최대한 줄이기 위함이다. 신경망 학습 후 프루닝-재학습 과정을 반복할 때, 모델의 정확도인 Q_{t-1}과 Q_t의 차이가 η보다 커지면, 프루닝을 과도하게 수행하여 재학습을 하더라도 정확도의 손실이 복원될 수 없기 때문에 프루닝 하기 전의 모델에 대해 d로 나눈 s를 이용하여(즉, s 값을 낮춰서) 다시 프루닝-재학습을 진행한다. 이처럼, 압축된 모델의 정확도 변화를 분석하여 문턱값을 결정하는 s를 동적으로 조정하면서 프루닝을 진행함으로써, 과도한 프루닝으로 인한 정확도 손실을 방지할 수 있다. 프루닝-재학습 과정 중 모델의 정확도(Q_t)와 압출률(C_t)이 원하는 정확도와 압축률에 도달하면 알고리즘은 종료된다. 결과적으로, 제안 알고리즘은 입력으로 원하는 압축 모델의 정확도와 압축률을 주면, 동적으로 문턱값을 조정하면서 프루닝-재학습 과정을 수행하여 빠른 시간 내에 원하는 압축 모델을 반환한다.

Algorithm 1. Dynamic adjustment of the pruning threshold

프루닝 문턱값 동적 조정 알고리즘

입력: M_0(초기 모델), $\overline{Q}$(기대 정확도), $\overline{C}$(기대 압축률)
출력: M_t(압축 모델)

$stdev()$: 가중치의 표준편차 계산
$cal\_accuracy()$: 모델의 정확도 계산
$calc\_compr\_rate()$: 모델의 압축률 계산

$t = 0,\ s_0 = 2$
while True:
  $th_t = s_t \times stdev(M_t)$
  $M_t = retrain(prune(M_{t-1}, th_t))$
  $Q_t = calc\_accuracy(M_t)$
  $C_t = calc\_compr\_rate(M_{t-1}, M_t)$
  If $Q_{t-1} - Q_t > \eta$:
    $M_t = M_{t-1}$
    $s_t = s_{t-1}/d$
  If $Q_t \geq \overline{Q}$ and $C_t \geq \overline{C}$:
    break
  $t = t + 1$

Ⅳ. 실험 결과

실험은 Intel Core i7-8700 3.2GHz CPU와 GeForce RTX 3600 그래픽 카드가 탑재된 PC에서 진행하였다. 모델 구현은 Python 3.6에서 PyTorch 딥러닝 라이브러리를 활용하였다.

제안하는 기법의 성능을 검증하기 위해 초기 CNN 모델인 LeNet[7]을 MNIST[8] 데이터 셋을 사용하여 학

습하였고, Github에 제공된 코드[9]를 사용하여 3개의 완전연결층(fully-connected layers)으로 구성된 LeNet (학습 후 정확도는 95.44%, 가중치 수는 266,610개)을 경량화했다. 문턱값을 경험적으로 설정하는 기존의 방식을 사용하는 Baseline 기법의 s는 0.25로 설정하였다. 제안된 문턱값 동적 조정 알고리즘의 제어 파라미터인 s_0, η, d는 각각 실험을 통해 가장 우수한 결과를 보이는 2, 5, 2로 각각 설정하여 실험을 진행하였다.

표 1과 2, 그림 3과 4는 기대 정확도와 압축률을 각각 95.0%, 96.0%로 설정하여 실험한 결과를 보여준다. 두 결과 모두 제안된 문턱값 동적 조정 기법을 사용하였을 때, 우수한 성능을 유지하며 압축 모델을 얻는데 걸리는 시간을 단축한 것을 확인할 수 있다. 기대 정확도와 압축률이 95%일 경우, 제안된 기법은 높은 s_0를 가지고 시작하기 때문에 한 번의 프루닝-재학습 과정을 통해 높은 압축률을 바로 달성할 수 있었으나, baseline 기법은 여러 번의 프루닝-재학습 과정을 필요로 했다. 기대 정확도와 압축률이 96%일 경우, 제안된 기법은 높은 s_0를 가지고 시작하기 때문에 빠르게 압축률을 달성하지만 정확도 저하가 동반되어 동적으로 s를 조정하면서 모델 복원(프루닝 하기 전 모델로 되돌림), 프루닝, 재학습 과정을 반복하면서 기대 정확도와 압축률을 만족하는 압축 모델을 반환했다. 그러나, 모델 복원 및 동일한 모델에 대한 프루닝-재학습 과정을 반복함에도 불구하고 baseline 기법보다는 프루닝-재학습 과정의 횟수가 적었다. 결과적으로 제안 기법은 압축 모델을 얻는 데 걸린 시간이 baseline 기법 대비 1.3 ~ 3배 단축되었다.

Table 1. Pruning results when setting the expected accuracy and compression rate to 95%

Algorithm	Accuracy	Compression rate	Elapsed time
Baseline (s0=0.25)	95.80 (st=s0)	96.30% (27.01x)	24m 9s
Ours (s0=2)	96.82 (st=2)	95.11% (20.46x)	8m 54s

Table 2. Pruning results when setting the expected accuracy and compression rate to 96%

Algorithm	Accuracy	Compression rate	Elapsed time
Baseline (s0=0.25)	96.92 (st=s0)	97.06% (34.7x)	27m 5s
Ours (s0=2)	96.52 (st=0.5)	97.66% (42.65x)	21m 27s

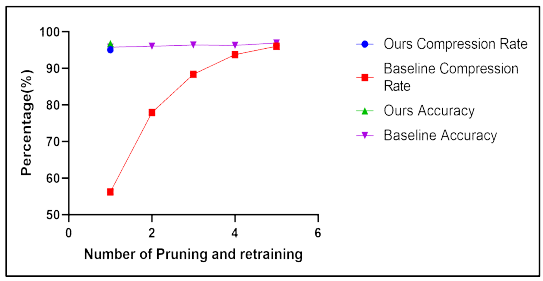


Fig. 3. Pruning results when setting the expected accuracy and compression rate to 95%

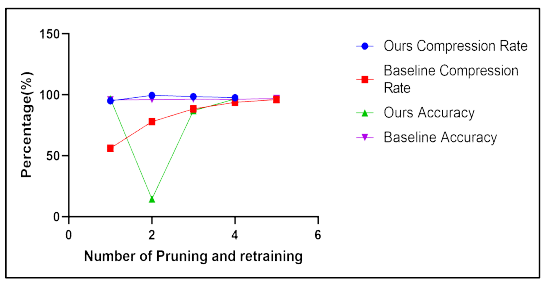


Fig. 4. Pruning results when setting the expected accuracy and compression rate to 96%

앞서 언급한 것처럼, 제안 기법은 세 개의 제어 파라미터를 가지는데, 표 3은 η 값에 따라 프루닝-재학습 과정의 횟수, s_t, 압축 모델을 얻는 데 걸린 시간의 변화를 보여준다. 이때, η를 제외한 파라미터들은 동일한 값으로 고정하였다. η가 커질수록 프루닝-재학습 과정의 횟수와 압축 모델을 얻는 데 걸린 시간은 줄어들고 s_t는 큰 값으로 유지되는 것을 확인할 수 있다. 그러나, η가 커질수록 프루닝에 의한 정확도 저하를 재학습을 통해 복원할 수 있는 가능성이 낮아지기 때문에 η를 5로 설정하였을 때, 속도 및 정확도 측면에서 가장 좋은 결과를 얻을 수 있었다.

Table 3. Results of the proposed algorithm with different η

η	1	3	5	7
Number of pruning+retraining	2	2	1	1
Final s_t	1.0	1.0	2	2
Elapsed time	39m 30s	33m 53s	23m 53s	22m 9s

Ⅴ. 결론

본 논문에서는 신경망 압축 기술 중 프루닝의 문턱값을 동적으로 조정하는 CNN 압축 기법을 제안하였다. 경험적으로 고정된 문턱값을 사용하는 기존 프루닝 기술과 달리 동적으로 최적의 문턱값을 찾아 프루닝-재학습 과정의 횟수를 줄임으로써 높은 정확도와 압축률을 가진 압축 모델을 얻는 시간을 최대 3배 정도 단축할 수 있음을 확인하였다.

본 논문에서는 제안 기법을 LeNet의 완전연결층에 적용한 결과만을 제시했기 때문에, 향후 다른 신경망이나 콘볼루션층에 적용하여 성능을 분석하는 연구가 필요하다.

ACKNOWLEDGMENTS

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2021R1F1A1045749).

REFERENCES

[1] H. Park, “A review of 3D object tracking methods using deep learning,” Journal of the Institute of Convergence Signal Processing, vol. 22, no. 1, pp. 30-37, 2021.

[2] R. Yamashita, M. Nishio, R. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” Insights into Imaging, vol. 9, pp. 611-629, 2018.

[3] Y. J. Lee, Y. H. Moon, J. Y. Park, and O. G. Min, “Recent R&D trends for lightweight deep learning,” Electronics and Telecommunications Trends, vol. 34, no. 2, pp. 40-50, 2019.

[4] S. Han, H. Mao, and W. J. Dally, “Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding,” Proc. of ICLR, 2016.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Communications of the ACM, vol. 60, no. 6, pp. 84-90, 2017.

[6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” Proc. of ICLR, 2015.

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[8] MNIST handwritten digit database, http://yann.lecun.com/exdb/mnist

[9] Deep-Compression-PyTorch,http://github.com/mightydeveloper/Deep-Compression-PyTorch.git

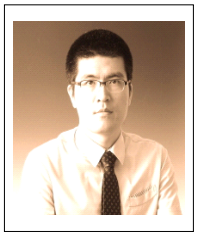
저자 소개

이 여 진 (Yeojin Lee)

2021년 3월~현재 : 부경대학교 전자공학과

관심분야 : 딥러닝 모델 경량화

박 한 훈 (Hanhoon Park)



2000년 2월 : 한양대학교 전자통신전파공학과(공학사)

2002년 2월 : 한양대학교 전자통신전파공학과(공학석사)

2007년 8월 : 한양대학교 전자통신전파공학과(공학박사)

2012년 3월~현재 : 부경대학교 전자공학과 교수

관심분야 : 증강현실, 인간컴퓨터상호작용, 컴퓨터비전/그래픽스, 딥러닝 응용