향상된 비트 평면 분할을 통한 다중 학습 통합 신경망 구축

배준기, 배성호
경희대학교
2013104081@khu.ac.kr, shbae@gmail.com

Improved Adapting a Single Network to Multiple Tasks By Bit Plane Slicing and Dithering

Joon-ki Bae, Sung-ho Bae
KyungHee University

요 약

본 논문에서는 직전 연구였던 비트 평면 분할과 디더링을 통한 다중 학습 통합 신경망 구축에서의 한계점을 분석하고, 향상시킨 방법을 제시한다. 통합 신경망을 구축하는 방법에 대해 최근까지 시도되었던 방법들은 신경망을 구성하는 가중치(weight)나 층(layer)를 공유하거나 태스크 별로 구분하는 것들이 있다. 이와 같은 선상에서 본 연구는 더 작은 단위인 가중치의 비트 평면을 태스크 별로 할당하여 보다 효율적인 통합 신경망을 구축한다. 실험은 이미지 분류 문제에 대해 수행하였다. 대중적인 신경망 구조인 ResNet18 에 대해 적용한 결과 데이터셋 CIFAR10 과 CIFAR100 에서 이론적인 압축률 50%를 달성하면서 성능 저하가 거의 발견되지 않았다.

1. 서론

최근까지 이미지 분류, 물체 인식 등 컴퓨터 비전 분야에서 딥러닝이 보여준 성능은 가공할 만 하다. 신경망 구조 또한 이 사실에 기여한다. 대중적으로 많이 사용되는 신경망의 구조로는 [1], [2], [3] 등이 있다. 이들의 공통점으로는 더 좋은 성능을 내기 위해서 더 많은 층(layer)들로 구조를 구성했다는 것이며, 이에 추가적으로 [2]처럼 스킵 연결(skip connection)등의 특징을 별도로 갖는다. 하지만 신경망의 구조가 복잡해질수록 더 많은 하드웨어 자원이 요구되고, 연산의 양 또한 증가한다. 이러한 특성에 의해 하드웨어적 제약이 있는 환경에서 딥러닝 기술을 적용하기 힘들다. 앞선 문제를 해결하기 위한 신경망 압축 연구가 활발히 진행되고 있다. 본 연구는 그 중에서 통합 신경망(Universal Neural Network)에 관한 연구이다.

통합 신경망이란, 하나의 문제를 해결하기 위해 하나의 신경망을 특수하게 학습해왔던 것과 달리 하나의 통합된 신경망으로 여러 가지 문제를 해결하는 것이다. 이론적으로, 두 개의 문제를 하나의 신경망이 해결할 수 있다면, 절반 이상의 압축률을 달성할 수 있다.

통합 신경망에 관한 연구는 언제든지 데이터에 대한 접근이 가능하다는 점을 제외하면 연속 학습(continual learning)이라 불리는 연구 분야의 접근방법과 동일하다. 지금까지 제시된 기법들 중 특정 구간의 층을 공유하는 [4], 신경망 내의 가중치들을 특정 문제에만 관여하도록 만들어주는 [5], [6] 등이 있다. 본 논문의 선행 논문에서 특정 층이나 가중치보다 더 좁은 범위인 비트 단위에서 비트 평면 분할과 디더링 효과를 통해 통합 신경망을 구축하는 방법을 제시했다. 성능 보존에만 초점을 맞추었던 유사 연구들과는 달리 성능 저하와 신경망의 압축률을 동시에 고려하였다. 하지만 성능 저하를 피할 수 없었고,

본 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

제시되었던 실험 설정에서 23.4%의 압축률만을 보였다. 이의 한계점을 극복하기 위해 새로운 비트 평면 분할 방법을 제안한다. 그 결과 두 가지 문제에 대한 통합신경망이 성능 저하 없이 50%의 이론적인 압축률을 달성하였다.

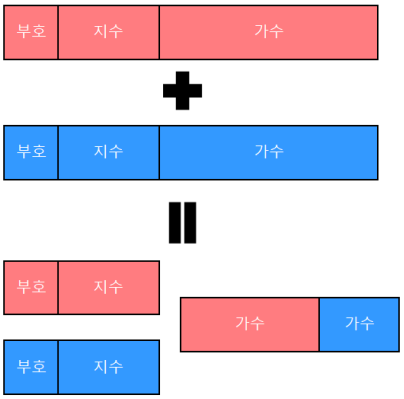

그림 1. 직전 연구의 비트 평면 분할 방식

2. 이전 연구

이전 연구에서는 비트 평면 분할과 디더링 효과를 기대할 수 있는 방법을 소개하였고, 이를 비트 융합이라 명명했다[7].

2.1 비트 평면 분할

비트 평면 분할은 영상처리 분야에서 쓰이는 양자화의 일종이다. 영상에서는 최상위 비트(MSB)가 영상의 정보를 가장 많이 담고 있으며, 반대로 최하위 비트(LSB)는 상대적으로 무시할 만한 정보량을 가진다. 이런 특성에 기반하여, 비트열의 하위 부분을 0 으로 치환하여 비트의 표현 범위를 줄여도 원본 영상에 비해 품질이 크게 떨어지지 않는다. 본 연구에서는 이러한 성질을 딥러닝 신경망의 가중치에 적용한다.

2.2 디더링

디더링은 양자화를 거친 데이터에 임의의 잡음을 더하여 양자화 오류를 줄이는 과정이다. 영상에 적용하는 경우, 매끄럽지 못한 계단 모양의 윤관선을 부드럽게 사실감을 높여주는 작용을 한다. 본 연구에서는 비트 평면 분할 과정에서 손실된 비트 부분을 작은 정밀도를 요구하는 태스크의 가중치로 채워 기존 문제의 성능을 보존하고자 한다.

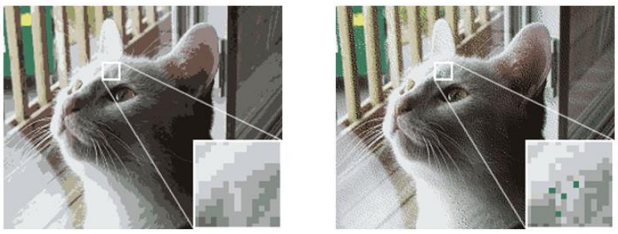

그림 2. 디더링 적용 전(좌)과 후(우)의 영상

2.3 비트 융합

본 연구에서는 실수 값을 가지는 가중치에 대하여 비트 융합을 실시하였으며, 부동 소수점 표준 IEEE 754 를 따른다. 두 개의 문제에 대해 비트 융합을 실시하는 과정을 설명한다. 문제 A 와 B 는 동일한 도메인의 문제이며, A 가 B 보다 더 높은 정밀도를 요구한다고 가정한다. 우선, 같은 구조의 신경망을 A 와 B 에 대하여 각각 학습시킨다. 그 다음, 학습된 두 개의 신경망의 가중치를 비트 평면 분할 한다. 마지막으로 A 에 대한 신경망 가중치의 절단된 비트 평면을 B 에 대한 신경망 가중치로 채워 디더링과 유사하게 처리한다.

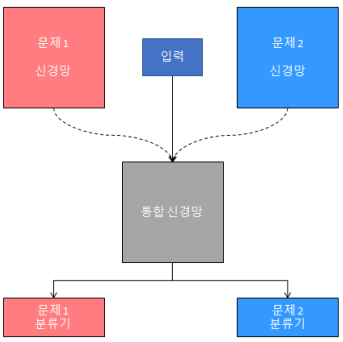

그림 3. 비트 융합을 통한 통합 신경망 구축

3. 관련 연구

직전 연구에서 제시된 비트 융합의 한계점을 극복하기 위해, 본 논문에서 새롭게 적용한 관련 연구들을 소개한다.

3.1 양자화

딥러닝 신경망에서 양자화란 넓은 비트 수를 사용하는 실수 자료 형을 더 적은 비트 수를 사용하는 정수로 근사 시키는

과정이다. 이 과정은 계산 복잡성, 메모리 소요, 연산 속도 등을 개선시키는 역할을 한다[8].

본 연구보다 먼저 기존의 방법을 개선하기 위해 32 비트 실수 형 가중치를 8 비트 정수 형 가중치로 양자화 한 후 비트 융합을 시도했다. 그 결과, 압축률은 대폭 상승하였으나, 성능 저하가 많이 발생했다. 첫 번째 원인으로는 비트 수가 적어질수록, 하위 비트의 중요도가 상승함을 들 수 있다. 그리고, 융합 전의 가중치가 작으면 작을수록 융합 후의 가중치가 임의의 잡음으로 역할 하지 않고 주된 값이 되는 현상이 발견되었다. 예를 들어, 융합 전의 가중치가 8 비트 정수에서 240 인 경우 하위 4 개의 비트가 융합된 후 255 의 값을 가진다면 변화하는 크기가 6.25% 밖에 되지 않지만, 기존 가중치가 1 인 경우에 융합 후 15 의 값을 가진다면, 변화한 크기가 1500%에 달한다. 즉, 가중치의 크기로 보았을 때, 기존 문제를 해결하기 위해 불필요하게 취급되었던 가중치가 깊게 관여하는 현상이 발생하게 된다. 이를 대체하여 [9]에서 제시된 블록 부동 소수점 양자화 전략을 고려했다.

3.2 블록 부동 소수점

[9]에서 제시된 효율적인 컨볼루션 기법 중 특징 맵 (feature map)에 적용되었던 양자화 기법을 소개한다. 특징 맵은 활성함수(ReLU)를 통과한 후를 가정하며, 부호 없는 정수로 취급한다. 32 비트 실수 형을 8 비트의 지수와 n 비트($n \leq 24$)의 가수로 분할하는 것이다. 즉 본래 24 비트였던 가수를 n 비트만을 사용하여 표현 범위를 줄인다.

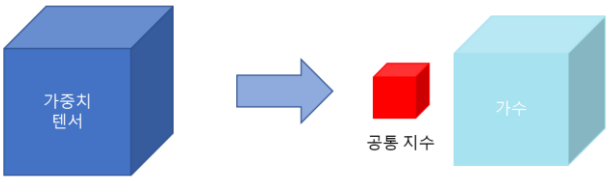

그림 4. 블록 부동 소수점의 적용

3.3 PackNet

본 연구와 유사한 연구인 연속 학습 기법 중 [5]가 있다. 이는 하나의 신경망이 여러 가지 문제를 해결할 수 있다는 점에서 통합 신경망과 유사하다. 하나의 기준이 되는 신경망을 학습한 후, 반복적인 가지치기[8]를 거쳐, 새로운 문제를 해결할 가중치를 확보하고 기존의 문제를 해결할 가중치를 고정한다. 다음 문제에 대한 학습을 진행할 때, 이전 단계에서 확보한 가중치만을 학습시키는 방식을 따른다. 본 연구도 이전 문제에 대해 학습한 신경망이 가지는 잉여(redundancy)를 비트 평면 분할로 해소하여 새로운 문제에 대한 가중치로 채워 넣는 기법이라는 점에서 유사하다.

4. 제안 방법

2.3 에서 보인 비트 융합과정에 3.2 기법이 적용된 것과 같다. 두 개의 문제(A, B)에 대하여 각각 학습한 신경망에 대해 블록 부동 소수점 기법을 적용한다. 이 때, 신경망의 가중치는 음수도 포함되기 때문에 부호를 위한 1 비트 텐서(Tensor)를 따로 저장한다. 기존에 가중치 별로 지수 비트를 저장했던 것과 달리 가중치 텐서 별로 공통된 지수 스칼라 한 개 만을 저장하여, 압축률을 높인다. 또한, 가수 부분의 비트 수를 줄이면서 양자화를 하기 때문에 압축률이 증가한다. 즉, A 문제에 대한 신경망의 가중치의 가수를 n 비트, B 문제에 대한 신경망의 가중치의 가수를 m 비트로 설정한다면, 기존의 32 비트씩 필요로 하여 64 비트의 공간이 필요했던 것이 m + n + 2(부호)만큼의 비트 수만이 필요하게 된다($m \leq 24$, $n \leq 24$). 추론 단계에서는 문제에 따라 가수 텐서에 쉬프트 연산이나 비트 마스킹 연산을 적용한 후 수행한다.

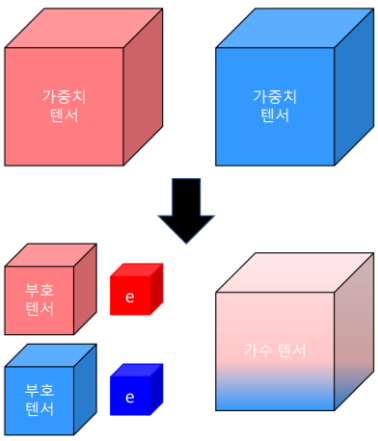

그림 5. 블록 부동 소수점을 적용한 비트 융합

5. 실험 및 분석

실험은 이미지 분류 문제에 대해서 수행한다. 기준 모델로는 대중적인 신경망 구조인 ResNet18 을 사용하고, 데이터셋은 CIFAR10 과 CIFAR100 을 사용한다. 문제에 따라 가수 부분에 몇

비트를 할당할 것인지를 선별하기 위해, 숨겨진 비트를 제외한 23 비트부터 1 비트까지 수행하였으며, 각각의 문제에 대해 성능 저하가 없는 선인 12 비트(CIFAR100), 11 비트(CIFAR10)을 선택하였고, 이에 따라 비트 융합을 실시하였다. 이론적으로 64 비트가 필요했던 것이 $(11 + 12 + 2)$ 비트가 필요하게 되었다. 지수는 공통된 하나의 실수 형 스칼라이기 때문에 상대적으로 작으므로 무시한 수치다.

	CIFAR100	CIFAR10	가중치 용량
ResNet18[2]	78.22	95.39	2α
분류기만 학습	78.22	90.6	α
Piggyback[6]	78.20	95.25	$\alpha + 2\beta$
비트 융합[7]	77.91	95.13	$\alpha + 2s + 2e$
향상된 비트 융합	78.13	95.18	$\alpha + 2s$

표 1. 실험 결과

실험 결과 기존 방법들에 비해 가중치 용량이 상대적으로 작지만, 비슷한 성능을 보이는 것을 알 수 있다. α는 기준 모델 의 컨볼루션 층의 가중치 용량, β는 [6]에서 문제 별 이진 마스크의 용량, s는 부호 비트의 용량, 그리고 e는 지수 비트의 용량이다. 본 실험에서 α는 $32 * 11.7 * 10^6$이고, β, s는 $1 * 11.7 * 10^6$, 그리고 e 는 $8 * 11.7 * 10^6$이다.

6. 결론

직전 연구에서 상대적으로 낮은 압축률과 성능 보존 능력을 보였던 것에 기초하여, 더 높은 압축률을 달성하고 성능 보존이 가능한 비트 융합 방법을 소개했다. 제시된 압축률을 달성하기 위해 기존에 가중치 별로 지수를 저장했던 방식과 달리 하나의 공통된 지수 스칼라를 추출하였고, 가수 부분에 블록 부동 소수점 방식을 적용하였다. 그 결과 가수 부분을 IEEE 754 표준과 같이 숨겨진 비트를 포함한 24 비트를 사용하더라도 이론적으로 50%의 압축률을 달성할 수 있음을 보였다. [10]에서 소개하는 혼합 정밀도(mixed precision)를 통해 더 높은 압축률을 달성하거나 [11]에서 제시된 잉여 비트 활용 방법과 함께 더 높은 성능 보존 능력을 달성할 수 있을 것으로 기대된다.

참고 문헌

[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[2] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[3] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[4] Li, Zhizhong, and Derek Hoiem. "Learning without forgetting." IEEE transactions on pattern analysis and machine intelligence 40.12 (2017): 2935-2947.

[5] Mallya, Arun, and Svetlana Lazebnik. "Packnet: Adding multiple tasks to a single network by iterative pruning." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[6] Mallya, Arun, Dillon Davis, and Svetlana Lazebnik. "Piggyback: Adapting a single network to multiple tasks by learning to mask weights." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[7] 배준기, and 배성호. "비트 평면 분할과 디더링을 통한 다중 학습 통합 신경망 구축." 한국정보과학회 학술발표논문집 (2019): 1498-1500.

[8] Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." arXiv preprint arXiv:1510.00149 (2015).

[9] Nascimento, Marcelo Gennari do, Roger Fawcett, and Victor Adrian Prisacariu. "DSConv: Efficient Convolution Operator." Proceedings of the IEEE International Conference on Computer Vision. 2019.

[10] Micikevicius, Paulius, et al. "Mixed precision training." arXiv preprint arXiv:1710.03740 (2017).

[11] "O-2A: Low Latency DNN Compression with Outlier-Aware Approximation". Design Automation Conference (DAC). 2020.