주파수 영역에서의 군집화 기반 계층별 딥 뉴럴 네트워크 압축

*홍민수 *김성제 *정진우
한국전자기술연구원
*hms9110@keti.re.kr

Deep Neural Network compression based on clustering of per layer in frequency domain

*Hong, Minsoo *Kim, Sungjei *Jeong, Jinwoo
Korea Electronics Technology Institute

요약

최근 다양한 분야에서 딥 러닝 기반의 많은 연구가 진행되고 있으며 이에 따라 딥 러닝 모델의 경량화를 통해 제한된 메모리를 가진 하드웨어에 올릴 수 있는 경량화 된 딥 뉴럴 네트워크(DNN)를 개발하는 연구도 활발해졌다. 이에 본 논문은 주파수 영역에서의 군집화 기반 계층별 딥 뉴럴 네트워크 압축을 제안한다. 이산 코사인 변환, 양자화, 군집화, 적응적 엔트로피 코딩 과정을 각 모델의 계층에 순차적으로 적용하여 DNN이 차지하는 메모리를 줄인다. 제안한 알고리즘을 통해 VGG16을 손실률은 1% 미만의 손실에서 전체 가중치를 3.98%까지 압축, 약 25배가량 경량화 할 수 있었다.

1. 서론

최근 영상 및 음성 등 다양한 분야에서 딥 러닝을 이용한 연구가 진행되고 있다. 이러한 연구를 통해 높은 성능을 가진 DNN과 알고리즘들이 개발되었지만, 하드웨어 또는 제한된 메모리를 가진 장비에서 DNN을 탑재하기 위해 네트워크 경량화의 중요성 또한 화두에 오르고 있다.

네트워크를 경량화하고 최적의 파라미터를 찾기 위해 가지치기(Pruning)[1], 증류 기법 (Distillation)[2] 등 새로운 기술들이 연구되고 있고, 또한 기존의 영상, 비디오 코덱 등에서 사용되던 압축 기술을 적용하여 성능은 유사하면서 메모리는 적게 소비하는 모델들을 개발하고 있다.

본 논문에서는 DNN의 가중치 필터를 주파수 영역으로 변환하여 에너지 집중 현상을 만들고 양자화를 통해 희소성을 증가시킨다. 그 후, K 군집 알고리즘을 통해 유사한 필터끼리 군집화를 한 후 허프만 코딩을 계층별로 적용해 모델을 압축한다. 압축 대상이 되는 모델은 VGG16[3]을 사용하며 각각 적용된 기법에 따라 압축률과 손실률을 계산 및 분석한다.

2. 주파수 영역에서의 군집화 기법

2.1 이산 코사인 변환

이산 코사인 변환(Discrete Cosine Transform)은 오디오, 영상 등 신호처리에서 널리 사용되는 기법으로, 블록 단위 연산을 통해 계층별 가중치 필터를 주파수 영역으로 변환한다. 변환된 가중치 필터는 낮은 주파수 영역과 높은 주파수 영역으로 나누어지며 저주파 부분에 값이 모이는 에너지 집중 현상이 나타난다. 이후 양자화를 통해 고주파 영역의 값을 제거하여 압축할 정보량을 줄이고 희소성을 증가시킨다.

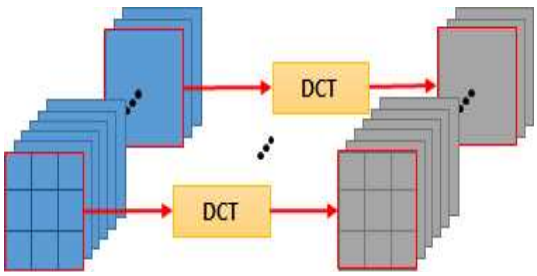

그림 1. 컨볼루션 계층에 대한 이산 코사인 변화 적용 방법
블록 크기는 컨볼루션 필터 크기와 같다.

가중치 필터에 이산 코사인 변환을 적용할 때에는 계층의 유형에 따라 변환 블록의 크기가 달라진다. 컨볼루션 계층(Convolutional layer)의 경우, 가중치 필터의 크기가 블록의 크기와 같고, 완전 연결 계층(Fully Connected layer)의 경우, 8 x 8 크기의 변환 블록을 사용한다.

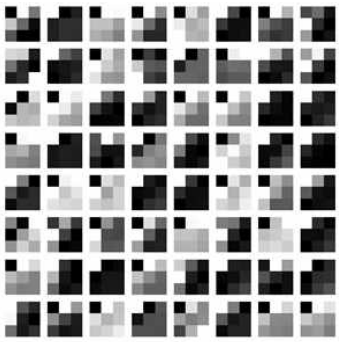

그림 2. 컨볼루션 계층의 이산 코사인 변환 결과.

2.2 양자화

부동소수점인 가중치를 정수 형태로 바꿔주기 위해 양자화(Quantization)를 적용한다. 또한, 이산 코사인 변화를 거친 가중치 필터의 높은 주파수 부분을 0으로 만들어 희소성(Sparsity)을 높이는 역할도 한다. 이를 통해 뒤에 이어질 엔트로피 코딩의 효율을 높일 수 있다.

본 논문에서는 양자화 방법으로 대칭 양자화 방법을 사용한다. 대칭 양자화 방법은 클리핑과 스케일링을 통해 가중치를 제시된 비트의 범위 내로 변환한다.

$$y_q = round(s * clip(y, -\alpha, \alpha))$$

$$clip(x) = \begin{cases} -\alpha, x \in (-\infty, -\alpha) \\ x, x \in [-\alpha, \alpha) \\ \alpha, x \in [-\alpha, \infty) \end{cases}$$

y 는 입력 가중치, α는 클리핑 계수, s는 양자화를 위한 스케일 계수이다. 32 비트 부동소수점 유형의 가중치를 s에 따라 스케일링을 한 후, 반올림을 통해 정수 유형의 가중치로 변환한다. 대칭 양자화 방법을 통해 기존의 32비트 보다 더 적은 비트를 사용함으로써 압축률을 높일 수 있다.

2.3 k-군집 알고리즘

이산 코사인 변환과 양자화를 거친 가중치 필터를 비슷한 형태를 가진 필터끼리 군집화하기 위해 컨볼루션 계층에 k-군집 알고리즘을 적용한다.

k-군집 알고리즘은 비슷한 가중치 필터를 k개의 군집으로 묶는 알고리즘으로, 가중치와 군집의 중심 값 간 유클리드 거리를 계산하여 가까운 군집에 가중치 필터를 배당하고, 군집의 무게중심을 군집의 중심값으로 재설정하며 군집화를 진행한다.

k-군집화의 수행방법은 그림 3과 같다. 각 컨볼루션 계층의 가중치는 4D이므로 연산을 빠르게 수행하기 위해 2D로 모양을 바꾼 후, 군집화를 수행한다. 군집화가 끝나면 k개의 군집과 각 가중치 필터가 어떤 군집에 속하는지에 대한 레이블 배열이 도출된다. 도출된 레이블 배열로 가중치 필터가 원래 어느 위치에 있었는지 알 수 있으므로, 압축된 가중치 필터를 복원할 때 사용한다.

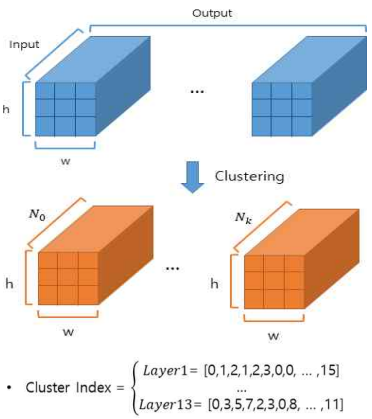

그림 3. 계층별 k-군집 알고리즘 수행방법. k는 군집의 개수, N_K는 k번째 군집에 속한 필터의 개수이며 계층별 Output의 길이만큼 레이블 배열이 생성된다.

2.4 허프만 코드

최종적으로 군집된 가중치 필터에 엔트로피 코딩 중 하나인 허프만 코드을 적용한다. 허프만 코드는 가중치의 빈도수를 기반으로 이진 트리를 만들어 각각의 가중치마다 다른 길이의 부호를 사용하는 무손실 압축 기법이다. 허프만 코딩을 거치고 나면 이진 형태의 비트스트림이 결과로 나오며 실제로 값을 의미하는 페이로드(Payload)부분과 각 가중치가 어떤 비트로 표현되는지에 대한 허프만 테이블(Tree)이 생성된다.

가중치의 범위가 좁을수록, 계층별 가중치의 분포가 치우쳐 있을수록 압축에 필요한 가중치의 개수가 줄어들기 때문에 허프만 코드의 압축률은 증가한다. 앞의 k-군집화를 통해 유사한 가중치 필터끼리 군집을 했기 때문에 군집 별 분포는 좁은 범위의 특정 분포로 치우쳐져 있게 되고 허프만 코드를 군집 단위 또는 군집의 픽셀 단위로 적용하여 최적화된 압축을 수행한다. 추가적으로 k-군집화를 통해 생성된 군집의 레이블 배열 또한 허프만 코드로 압축하여 레이블 배열이 차지하는 메모리를 감소시킨다.

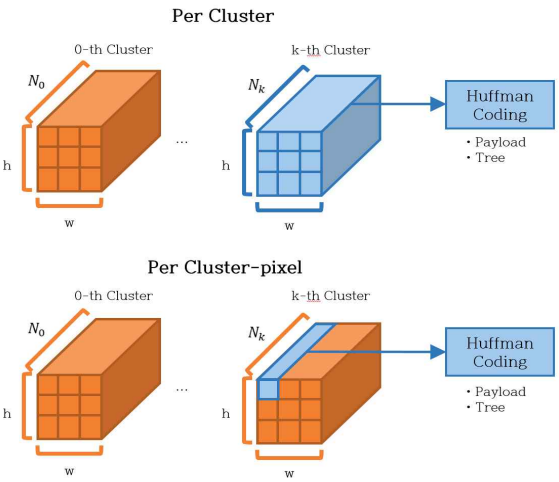

그림 4. 3 x 3 컨벌루션 계층의 군집별 허프만 코드 적용방법 예시

제안한 압축 기법의 전체 구조는 그림 5와 같다. 압축을 진행하는 인코더에서 순차적으로 DCT, 양자화, K-군집화, 엔트로피 코딩을 통해 가중치를 이진의 비트스트림으로 변환한다. 그리고 디코더에서 인코더의 순서를 역순으로 다시 진행하여 원래의 가중치를 복원해낸다.

본 논문에서 위의 인코더-디코더 구조를 통해 여러 경우의 실험을 진행하고 압축률과 복원 시 정확도를 측정하여 분석한다.

3. 실험 및 결과

그림 5의 인코더-디코더 구조를 통해 다양한 경우의 실험을 진행하고 압축률과 복원 시 정확도를 측정하여 분석한다. 윈도우 10, Geforce 2080Ti 기반 환경에서 Python 3.7.4 버전, pyTorch 1.3 버전을 이용하고 압축 대상이 되는 딥러닝 모델은 Top5 정확도 89.85%의 ImageNet[4]으로 학습된 VGG16을 사용했다. 실험에 걸린 시간은 정확도 측정까지 약 5분의 시간이 걸렸다. 양자화 비트는 8bit를 사용, 군집의 개수는 총 16개를 사용하여 4bit로 표현할 수 있게 했다.

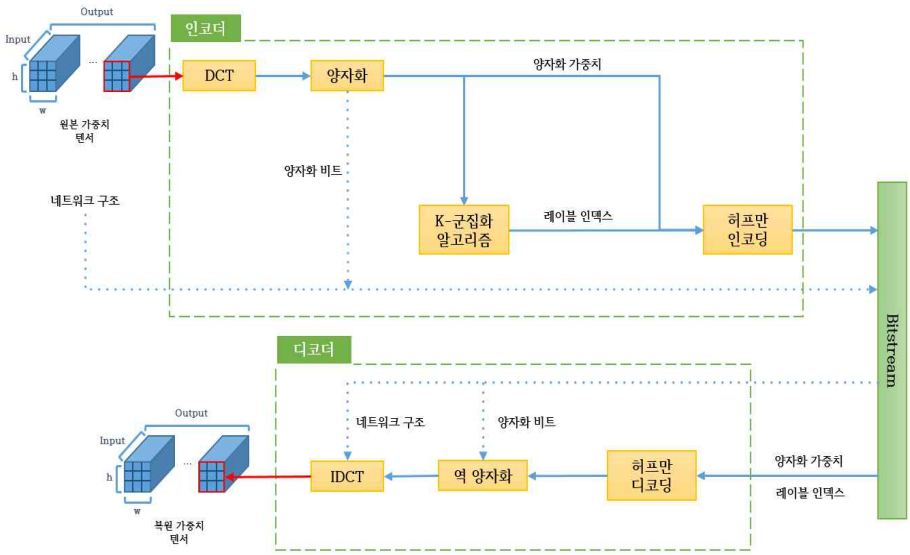

그림 5. 딥 뉴럴 네트워크 압축 전체 구조.

표 1은 VGG16에 대해 각 알고리즘 적용별 압축률(잔여 가중치) 및 정확도 결과이다. 압축률 계산은 아래의 수식과 같다.

$$Compression ratio = \frac{bitstream\ bytes}{original\ weight\ bytes}$$

이산 코사인 변환을 사용하지 않았을 때, 32 비트의 가중치를 8 bit로 양자화 하는 과정에서 약 2.6%의 정확도 손실이 일어난다. 그러나 이산 코사인 변환을 적용하면 저주파 에너지를 좀 더 보존할 수 있었기 때문에 1% 미만의 손실로 정확도 손실을 조금 더 막을 수 있었다.

엔트로피 코딩의 적용 예를 보면 단순히 계층별로 허프만 코드를 적용한 경우보다 K-군집화를 통해 가중치의 범위를 줄이고 특정 분포를 만들어줌으로써 군집별, 군집의 픽셀별로 허프만 코드를 적용했을 때 가중치를 조금 더 압축하는 결과를 나타냈다.

결과적으로 제안된 알고리즘을 통해 VGG16을 손실률은 1% 미만의 손실에서 3.98%까지 압축하여, 약 25배가량 경량화 할 수 있었다.

4. 결론

본 논문에서는 경량화된 딥 뉴럴 네트워크를 만들기 위해 주파수 영역 기반 압축 알고리즘을 제안하였다. VGG16의 각 계층의 가중치를 주파수 영역으로 변환하고 양자화를 적용한다. 그 후, K 군집 알고리즘을 통해 유사한 가중치끼리 군집화를 한 후 허프만 코딩을 계층별로 적용해 각 가중치의 분포에 맞게 최적화된 압축을 진행한다.

해당 압축 기법은 수행시간이 다른 기법에 비해 비교적 짧기 때문에 추가적으로 가지치기, 고도화된 엔트로피 코딩 기법과 함께 사용하면 압축률이 더 향상될 것으로 보인다. 또한, 계층의 유형에 맞게 최적화된 양자와 방법 또는 완전 연결 계층에 대한 최적화된 영역 변환을 적용하여 정확도의 손실도 줄일 수 있도록 향후 이에 대한 추가연구를 진행할 예정이다.

Layer		Q + SH	Q + K(16) + CH	Q + K(16) + CPH	D + Q + K(16) + CPH
Conv layer	1	23.4701%	33.4346%	37.9539%	29.3891%
	2	13.6214%	12.5332%	13.7315%	11.9322%
	3	12.1356%	11.9715%	12.0928%	11.2580%
	4	10.9243%	10.4646%	10.4372%	9.4478%
	5	9.7249%	9.0883%	8.6862%	8.1265%
	6	8.3497%	8.1422%	7.4788%	7.0401%
	7	8.3937%	8.1644%	7.5311%	7.0958%
	8	7.5595%	7.3234%	6.7069%	6.3572%
	9	6.4368%	6.7434%	6.0466%	5.7684%
	10	6.5307%	6.9106%	6.1802%	5.8955%
	11	7.0202%	7.1680%	6.4035%	6.1095%
	12	7.1507%	6.9604%	6.2002%	5.8433%
	13	6.9074%	6.3568%	5.8273%	5.5499%
FC layer	14	3.5262%	3.5262%	3.5262%	3.4180%
	15	4.8370%	4.8370%	4.8370%	4.8828%
	16	6.9945%	6.9945%	6.9945%	7.0731%
컨벌루션 계층 압축률		7.1845%	7.0499%	6.4349%	6.1018%
전체 압축률		4.1722%	4.1647%	4.0972%	3.9892%
복원 정확도		87.29%	87.29%	87.29%	89.01%
원본 정확도		89.85%			

표 1. 각 알고리즘 적용별 압축률(잔여 가중치) 및 정확도.
D : DCT, K : k-군집화, Q : 양자화, SH : 계층별 허프만 코드,
CH : 군집별 허프만 코드, CPH : 군집의 픽셀별 허프만 코드

Acknowledgement

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2019-0-01351, 활성화/커널데이터의 압축/복원을 통한 초저전력 모바일 딥러닝 반도체 기술 개발)

참 고 문 헌

[1] Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149.

[2] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." NIPS Deep Learning and

Representation Learning Workshop (2015)
[3] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
[4] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009.