DCT를 이용한 CNN 모델의 압축방법

김승환[1], 박은수[1], 굴람 무즈타바[2], 류은석[1]

성균관대학교[1], 가천대학교[2]

whitekomani@skku.edu, espark804@skku.edu, *mujtaba@gc.gachon.ac.kr, esryu@skku.edu

Compression Method for CNN Models Using DCT

Kim, SeungHwan Park, Eun-Soo Ghulam Mujtaba[2] Ryu, Eun-Seok

Sungkyunkwan Univ.[1] , Gachon Univ.[2]

요약

최근 이미지 인식을 위한 Convolutional Neural Network(CNN) 모델의 경량화에 관한 연구가 활발하게 이루어지고 있다. 그중 양자화는 모델을 구성하는 가중치의 크기를 낮추는 방법이다. 기존의 CNN 모델에서 가장 큰 비중을 하는 Fully Connected Layer(FCL)는 내부적으로 32 Bit의 실수 행렬로 표현된다. 본 논문에서는 미리 학습된 실수 가중치를 더 작은 비트의 정수 행렬로 양자화한다. 양자화된 행렬에 대해서 영상 압축 등에서 사용하는 Discrete Cosine Transform(DCT)을 통해 주파수 영역으로 변환한 후 고주파 영역을 생략하는 손실압축 방법을 제안한다. 실험을 통해 그 과정에서 손실에 따른 정확도의 변화를 나타낸다.

1. 서론

최근 딥러닝 모델에 관한 연구가 고도화되면서 경량화에 대한 수요가 증가하고 있다. GPU의 강력한 병렬처리를 바탕으로 많은 모델이 연구되었으며, 좋은 결과를 보여주었다. 그러나 모바일 기기 혹은 IOT장비처럼 연산성능이 적은 환경에서는 이러한 고도화된 모델의 사용이 제한된다. 모바일 기기 성능의 급격한 발전에도 불구하고 여전히 모바일 기기에서의 딥러닝 모델의 사용은 미미하다. 이를 해결하기 기존의 모델을 경량화하거나 연산 복잡도가 낮은 새로운 모델에 관한 연구가 활발하게 진행되고 있다.

모델 경량화는 좋은 성능을 보여준 기존의 모델을 변형하여 적은 연산량으로 동작하게 변형하는 기술을 말한다. 경량화는 크게 모델의 일부를 제거하는 Pruning[1]과 미리 학습된 큰 모델을 통해 작은 모델을 학습시키는 Knowledge-Distillation[2] 그리고 모델을 구성하는 가중치의 크기를 낮추는 Quantization이 있다. 본 논문은 그 중 Quantization 즉 양자화를 통해 경량화하는 방법을 제시한다.

Pruning은 모델에서 불필요한 부분을 잘라내는 경량화 기술이다[1]. 실제로 결과에 큰 영향을 주는 가중치는 한정되어 있다는 이론을 바탕으로 단순히 일정한 임계치보다 작은 값을 0으로 만드는 방법을 통해 50%의 가중치만으로 동일한 성능을 달성하였다. 또한, Pruning 후 재학습이라는 과정을 통해 기존의 10%의 가중치만 사용해도 성능저하가 없는 결과를 보여주었다. 이를 바탕으로 재학습하는 과정에 관한 연구가 활발하게 이루어진다. 최근에는 재학습 과정에서 처음 모델의 초깃값을 재활용하는 방법을 통해 기존의 모델의 20%의 가중치를 사용하지만, 학습속도와 정확도가 높은 모델을 만드는 방법이 발표되었다[3].

높은 성능의 모델은 일반적으로 많은 가중치가 필요하다. 특히 여러 개의 모델을 결합하여 사용하는 Ensemble 모델은 높은 정확도를 달성하지만 가중치가 매우 많은 모델이다. Geoffrey Hinton 가 발표한 [2]에서는 이러한 거대하지만, 성능이 좋은 모델을 통해 새로운 모델을 학습하는 Knowledge-Distillation의 개념을 제시했다. Knowledge-Distillation은 기존의 네트워크(Teacher Network)를 통해 더 작은 네트워크(Student Network)의 학습을 보조하는 방법이다. 학습 과정에서 Student Network는 Ground Truth과 예측값의 차이와 함께 Teacher Network의 예측값과의 차이를 Loss Function에 반영함으로써 Teacher Network를 모방하는 방향으로 수렴한다. 이를 방법을 통해 MNIST 데이터 세트에 대한 실험에서 3을 제외하고 학습했지만 98.6%의 정확도를 달성했다.

일반적으로 사용하는 딥러닝 모델의 가중치는 32 Bit 혹은 64 Bit 부동 소수점으로 구성되어 있다. 크기가 큰 가중치를 사용하면 학습 과정에서 미세한 차이가 손실되지 않는 효과가 있다. 또한, 고성능 기기에서는 최적화된 연산과 GPU의 가속을 통해 효과적인 연산이 가능했다. 하지만 GPU가 없는 모바일 기기나 IoT 장비에서는 이러한 모델을 이용하는 것이 불가능하다. 양자화(Quantization)는 모델을 구성하는 가중치를 더 작은 공간으로 표현하는 방법이다. 최근에 Google에서 발표한 [4]에서는 Linear Quantization을 통해 기존의 모델을 8 Bit 정수로 양자화하여 기존의 성능을 유지하면서 모델의 크기를 25% 수준으로 줄이는 방법을 제시했다.

대부분의 CNN 모델은 분류기로 Fully-Connected Layer (FCL)를 사용한다. FCL은 내부적으로 가중치(Weight)와 편향(Bias) 두 개의 행렬로 구성된다. (3×3), (5×5)의 Filter들로 구성되는 Convolution Layer에 비해 크기가 매우 큰 행렬로 모델의 대부분을 차지한다. 따라서 FCL의 크기를 크게 줄이면 모델 전체의 크기를 효과적으로 줄일 수 있다. 본 논문에서는 이미 학습된 모델에 대하여 Dense Layer만을 변형하여 그 크기를 줄이는 방법을 제시하고 손실압축에 따른 정확도의 변

화를 보인다.

2. 관련 연구

본 절에서는 DCT에 관해 설명하고 CNN 모델에 DCT를 적용한 연구들을 소개한다. 2.1에서는 DCT의 정의와 변형모델을 소개하며 양자화 방법을 설명한다. 2.2에서는 CNN 모델에서 입력 데이터에 DCT를 적용한 연구와 모델 자체에 DCT를 적용한 연구들을 각각 소개한다.

2.1 DCT(Discrete Cosine Transform)

DCT는 신호처리와 데이터 압축에 널리 사용되는 선형 변환이다. AVC(Advanced Video Coding), HEVC(High Efficiency Video Coding) 등의 영상 코덱과 JPEG 등의 이미지 압축기술에는 DCT가 포함되어 있다[5]. DCT는 고주파와 저주파 성분이 분해된 주파수 영역으로 변환하는 가역 선형 변환이다[6]. 수식의 정의는 다음과 같다. N개의 실수 $x_0, ..., x_{N-1}$는 DCT를 통해 N개의 실수 $X_0, ..., X_k, ..., X_{N-1}$ 로 변환된다.

$$X_k = \frac{1}{2}\left(x_0 + (-1)^k x_{N-1}\right) + \sum_{n=1}^{N-2} x_n \cos\left[\frac{\pi}{N-1} nk\right] \quad (1)$$

또한, DCT에는 수식 1의 정의를 수정한 몇 가지 변형이 존재한다. 그중 DCT-II는 비디오와 이미지 압축에 활용되는 변형이다[10].

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right] \quad (2)$$

수식 1과 수식 2에서 소개한 DCT는 1차원에 대한 변환으로 이미지와 같은 2차원 행렬에 대해 DCT를 적용할 때는 행과 열에 DCT를 각각 수행한다. 수식으로 표현하면 수식 3과 같다.

$$X_{k_1,k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1,n_2} \cos\left[\frac{\pi}{N_1}\left(n_1 + \frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N_2}\left(n_2 + \frac{1}{2}\right)k_2\right] \quad (3)$$

혹은 행과 열에 대한 DCT를 각각 행렬 C 와 C^T로 정의하여 변환된 행렬 $V = CXC^T$의 형태로 표현하기도 한다.

DCT는 선형 변환으로 역변환을 통해 원본의 행렬을 복구할 수 있다. DCT를 사용하는 압축 알고리즘은 DCT 계수를 양자화하여 데이터를 압축한다. 양자화된 행렬 Z를 계산하는 방법은 다음과 같다.

$$Z_{k_1,k_2} = round\left(X_{k_1,k_2} / Qstep\right) \quad (4)$$

양자화의 압축률과 손실은 $Qstep$에 의해 결정된다. $Qstep$을 결정하는 방법은 압축 모델에 따라 달라진다. H.264의 경우 52개의 QP(Quantization Parameter)에 따라 $Qstep$을 대응한 테이블을 이용하여 행렬 전체를 하나의 $Qstep$으로 나누는 스칼라 양자화를 실시한다[7]. JPEG의 경우 위치에 따른 $Qstep$을 양자화 행렬을 통해 정의한다[5].

높은 $Qstep$을 사용할수록 고주파 영역의 계수는 대부분 0이 된다. 이를 이용해 다양한 엔트로피 코딩 기법을 통하여 최종 크기를 압축할 수 있다.

2.2 CNN과 DCT

DCT를 활용하여 CNN의 성능을 개선하는 연구는 다양하게 진행되었다. [8]은 DCT 계수를 입력 데이터로 학습한 모델과 RGB 이미지를 통해 학습한 모델이 유사한 성능을 보인다고 밝혔다. 또한, 두 모델은 공통적인 부분에서 오답이 발생하는 것을 발견했다. 얕은 CNN 모델에서 Feature Map에 DCT를 적용하는 것이 학습속도가 더 빠르며 최종 정확도는 원본 모델에 비슷하거나 몇몇 경우에는 더 좋은 결과를 보인다[9]. [10]에서는 이미지 전체에 대해 DCT를 적용하여 추출한 feature와 RBF (Radial Basis Function Network)를 통해 얼굴을 감지를 수행했다. [11]은 작은 크기의 이미지를 학습하는 과정에서 Sparse Auto encoder를 사용하여 기존의 이미지로 학습하는 방법보다 빠르게 학습하는 방법을 제시했다.

DCT를 통해 CNN 네트워크를 압축하는 연구도 많이 발표되었다. Wang, Y는 [12]에서 CNN의 Convolution 필터들을 DCT를 통해 압축하는 CNNpack 모델을 제안했다. CNNpack은 DCT를 입력 데이터에만 사용하는 것이 아니라 필터 또한 DCT를 통해 주파수 계수로 변환한다. 변환된 필터는 중복을 제거하는 부호화 기술을 통해 압축한다. 또한, 압축된 필터를 통해 Feature Map을 추출하여 시간과 용량을 절약하는 방법을 제안한다.

3. 구현내용 및 실험

본 논문은 정수로 양자화된 FCL을 DCT를 통한 양자화 및 부호화를 통해 압축하는 방법을 제안한다. 3.1에서 본 논문에서 사용한 가중치를 양자화하는 방법을 소개하며 3.2에서는 본 논문에서 DCT를 활용하는 방법을 설명한다.

3.1 모델의 양자화

본 논문에서는 [13]에서 제시한 양자화 방법을 간소화하여 가중치를 양자화한다. 이 과정에서 32bit 실수로 구성된 가중치 행렬 W는 8 Bit 정수형으로 구성된 행렬 W_q 와 행렬의 배율 변수 S로 변환된다. 변환하는 과정은 다음과 같다.

$$S = \frac{\max(|W|)}{127} \quad (5)$$

$$W_q = round\left(\frac{W}{S}\right) \quad (6)$$

편향(Bias) 또한 수식 5와 수식 6을 통해 양자화된 편향 B_q와 S로 변환된다. 이 양자화 과정을 통해 가중치의 크기는 4배 줄어든다.

양자화된 모델에서 Feed Forward를 진행할 때 가중치는 양자화된 값B_q, W_q가 아닌 S를 곱한 값을 통해 계산한다. 본 논문에서는 FCL에 대해서만 양자화하기 때문에 양자화하지 않은 다른 계층과의 연산은 유지해야 한다. 이는 정수로 변환된 가중치에 상수를 곱하는 연산으로 가중치의 표현 가능한 가짓수는 8 Bit 정수와 같다.

3.2 양자화된 모델에 대한 DCT

양자화된 가중치는 8 Bit 정수형으로 구성되어 있다. 본 논문에서는 양자화된 행렬에 대해서 8×8 DCT를 수행하여 주파수 영역으로 변환한다. 그 후 수식 4에 따라 DCT 계수를 스칼라 양자화한다. *Qstep*은 H.264의 QP와 *Qstep*의 테이블에서 QP를 2부터 2씩 늘려가며 실험을 진행한다[7]. 적용한 QP는 다음과 같다.

QP	2	4	6	8	10	...	50
Qstep	0.8125	1	1.25	1.625	2	...	208

표 1. 실험에 적용한 QP와 *Qstep*

4. 실험 결과

실험에 사용한 모델은 VGG16으로 FC1, FC2 2개의 FCL을 가지고 있다. 실험에서 변형한 두 레이어는 다음과 같이 구성된다.[14].

	종류	모양	가중치 수
FC1	kernel	(25088,4096)	103M
	bias	(4096,1)	4096
FC2	kernel	(4096,4096)	16M
	bias	(4096,1)	4096

표 2 VGG16의 FCL

실험에 사용한 데이터 셋은 Imagenet의 검증 데이터로 1000가지 클래스의 이미지 총 50000개로 구성되어 있다[15].

4.1 정수 양자화의 결과

	Top-1 정확도	Top-5 정확도	크기
원본 모델	71.86%	90.20%	476MB
FC1	70.78%	89.78%	170MB
FC2	70.71%	89.81%	428MB
FC1, FC2	70.71%	89.78%	119MB

표 3 정수 양자화된 모델의 실험 결과

표 3은 원본 모델과 양자화된 모델의 정확도와 모델의 크기를 나타낸다. FC1만 양자화했을 때 크기는 2.7배 줄었지만, 정확도의 차이는 미미했다. 마찬가지로 FC2만 양자화한 경우는 FC1의 데이터가 그대로 유지되지만, 정확도에는 차이가 작았다. FC1과 FC2를 모두 양자화한 경우 했을 경우 모델의 크기는 4배 줄어들었지만, 정확도는 1% 정도밖에 차이 나지 않는다.

4.2 DCT 양자화 실험 결과

DCT를 통해 구한 주파수 영역에서 수식 4를 통해 양자화를 실시하면 *Qstep*에 따라 0의 비중이 늘어난다. 그림 1은 *QP*에 따라 0이 된 주파수 영역의 비율을 나타낸 그래프이다. 주파수 영역에서 중복된 값이 많이 나타날수록 엔트로피 코딩을 통한 압축효율은 증가한다.

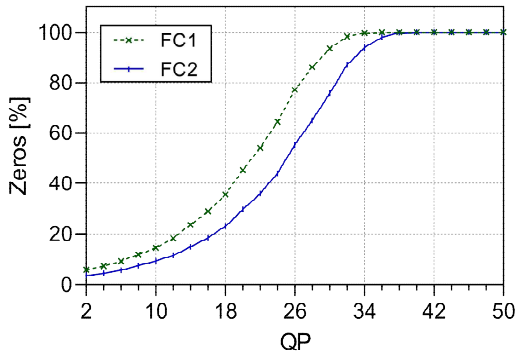

그림 1 *QP*에 따른 DCT 계수 중 0의 비율

그림 1에서 QP가 20이 되면서 급격하게 증가하다가 40에서는 0의 비율이 100%가 된다. 비율이 100%가 되면 모든 주파수 계수가 0이 된 것으로 역변환을 통한 구한 가중치 역시 모두 0이 된다. 따라서 QP가 40 이상이면 모델이 정상적으로 동작하지 않는다.

다음은 QP에 따른 정확도를 나타낸 그래프이다. QP의 범위는 그림 1에서 유의미한 증가가 시작되는 20부터 모든 가중치가 0이 되는 40까지의 정확도만을 살펴본다.

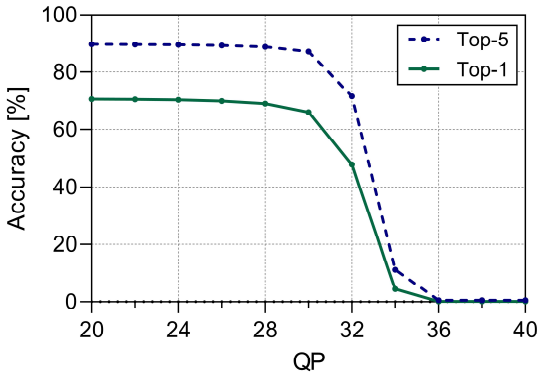

그림 2 *QP*에 따른 모델의 정확도

예를 들어 QP가 30인 경우 그림 1에서 0의 비율이 60%를 초과하지만, 그림 2에서 원본 모델과 비슷한 정확도를 달성했다. 반면 QP 32부터는 급격하게 정확도가 떨어지면서 34부터는 0에 가까운 정확도를 보인다.

5. 결론

본 논문은 양자화된 정수 가중치 행렬을 DCT를 통해 주파수 영역으로 변환하고 양자화하는 손실압축 방법을 제시한다. 실험 결과 DCT 계수의 대다수가 0이 되어도 정확도의 손실은 거의 없었다. 또한, 본 논문에서 제시하는 방법은 재학습 과정을 거치지 않으며, *Qstep*을 조절해 정확도와 크기를 조정할 수 있다. 추후 연구를 통해 다양한 모델에 적용하고 부호화를 통한 압축효율을 조사할 예정이다.

Acknowledgement

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT 연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2020-2017-0-01630)

참고문헌

[1] Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural network. In Advances in neural information processing systems (pp. 1135-1143).

[2] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

[3] Frankle, J., & Carbin, M. (2018). The lottery ticket hypothesis: Finding sparse, trainable neural networks. arXiv preprint arXiv:1803.03635.

[4] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., ... & Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2704-2713).

[5] G. K. Wallace, "The JPEG still picture compression standard," IEEE Transactions on Consumer Electronics, 1992, vol. 38, no. 1, pp. 18-34.

[6] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete Cosine Transform," IEEE Transactions on Computers, 1974, vol. C-23, no. 1, pp. 90-93.

[7] Wang, H., Kwong, S., & Kok, C. W. (2006). Efficient prediction algorithm of integer DCT coefficients for H. 264/AVC optimization. IEEE Transactions on Circuits and Systems for Video Technology, 16(4), 547-552.

[8] Ulicny, Matej, and Rozenn Dahyot. "On using cnn with dct based image data." Irish Machine Vision and Image Processing Conference 2017. Vol. 2.

[9] A. Ghosh and R. Chellappa. Deep feature extraction in the DCT domain. In 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 3536-3541.

[10] Er, M. J., Chen, W., & Wu, S. (2005). High-speed face recognition based on discrete cosine transform and RBF neural networks. IEEE Transactions on neural networks, 16(3), 679-691.

[11] Zou X., Xu X., Qing C., & Xing X. “High speed deep networks based on discrete cosine transformation.” 2014 IEEE International Conference on Image Processing (ICIP). IEEE, 2014.

[12] Wang, Y., Xu, C., You, S., Tao, D., & Xu, C. (2016). Cnnpack: Packing convolutional neural networks in the frequency domain. In Advances in neural information processing systems (pp. 253-261).

[13] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., ... & Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2704-2713).

[14] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[15] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." 2009 IEEE conference on computer vision and pattern recognition. IEEE, 2009.