

비트평면 영상을 이용한 이진 CNN 연산 알고리즘

최종호*

Binary CNN Operation Algorithm using Bit-plane Image

Jong-Ho Choi*

요약 본 논문에서는 이진영상과 이진커널을 사용하여 컨볼루션, 풀링, ReLU 연산을 수행하는 이진 CNN 연산 알고리즘을 제안한다. 256 그레이스케일 영상을 8개의 비트평면으로 분해하고, -1과 1로 구성되는 이진커널을 사용하는 방법이다. 이진영상과 이진커널의 컨볼루션 연산은 가산과 감산으로 수행한다. 논리적으로는 XNOR 연산과 비교기로 구성되는 이진연산 알고리즘이다. ReLU와 풀링 연산은 각각 XNOR와 OR 논리연산으로 수행한다. 본 논문에서 제안한 알고리즘의 유용성을 증명하기 위한 실험을 통해, CNN 연산을 이진 논리연산으로 변환하여 수행할 수 있음을 확인한다. 이진 CNN 알고리즘은 컴퓨팅 파워가 약한 시스템에서도 딥러닝을 구현할 수 있는 알고리즘으로 스마트 폰, 지능형 CCTV, IoT 시스템, 자율주행 자동차 등의 임베디드 시스템에서 다양하게 적용될 수 있는 시스템이다.

Abstract In this paper, we propose an algorithm to perform convolution, pooling, and ReLU operations in CNN using binary image and binary kernel. It decomposes 256 gray-scale images into 8 bit planes and uses a binary kernel consisting of -1 and 1. The convolution operation of binary image and binary kernel is performed by addition and subtraction. Logically, it is a binary operation algorithm using the XNOR and comparator. ReLU and pooling operations are performed by using XNOR and OR logic operations, respectively. Through the experiments to verify the usefulness of the proposed algorithm, We confirm that the CNN operation can be performed by converting it to binary logic operation. It is an algorithm that can implement deep running even in a system with weak computing power. It can be applied to a variety of embedded systems such as smart phones, intelligent CCTV, IoT system, and autonomous car.

Key Words : Bit-plane, Binary CNN, Computing Power, Embedded System, Binary kernel, XOR

1. 서론

최근 들어 컴퓨팅 파워가 급격하게 증가함에 따라 다수의 레이어로 구성되는 딥러닝이 영상인식의 분야에서 다양하게 응용되고 있다. CNN(Convolutional Neural Network)에서는 레이어의 수가 수십개 정도로 증가하고 있고, 특징추출 연산에 사용되는 커널 수도 레이어 당 다수가 사용된다[1,2,3]. 따라서 CNN은 주로 대용량 메모리와 연산속도가 빠른 CPU 및 GPU가 내장된 시스템에 구축되고 있다.

고속 대용량 서버에 구축하고 있는 CNN의 응

용분야는 금융, 회계, 의료 등으로 다양하고 그 활용도도 지속적으로 증가하고 있다. 이와는 달리, 최근에는 스마트 폰으로 대표되는 임베디드 시스템에 CNN을 구축함으로써 다양한 서비스를 앱을 통해 제공하고자 하는 수요가 급격하게 증가하고 있다. CCTV, IoT, ITS 등의 분야에서도 CNN에 대한 수요가 증가하고 있으나, 컴퓨팅 파워의 한계로 고성능 시스템을 구현할 수 없는 상황이다.

본 논문에서는 CNN에서 사용하고 있는 컨볼

This paper is the result of a study by the Kangnam University Intramural Research Support Project 2018.

*Corresponding Author : Department of IoT Electronic Engineering, Kangnam University(jhchoi@kangnam.ac.kr)

Received October 16, 2019

Revised November 17, 2019

Accepted November 20, 2019

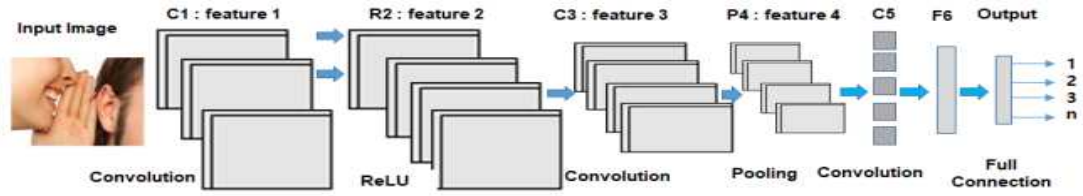


그림 1. CNN 구조
Fig. 1. CNN Architecture

루션, 풀링, ReLU 연산을 이진영상과 이진커널로 수행하는 알고리즘을 제안하였다. 이진영상과 이진커널의 컨볼루션 연산은 승산을 제외한 가산과 감산만으로 수행한다. 논리적으로는 XNOR 연산과 비교기로 구성되는 고속연산 알고리즘이다. 풀링과 ReLU 연산은 단순한 논리연산으로 수행한다. 256 그레이레벨 영상은 화소당 8비트가 할당되므로 8개의 비트평면 영상으로 분해된다. 8개의 비트평면 영상 모두를 CNN의 입력으로 사용할 수도 있고, 상위 비트평면 영상을 중심으로 축소시켜 선택함으로써 계층적으로 적용할 수도 있다. 응용분야에 따라 탄력적으로 적용함으로써 연산시간과 컴퓨팅 파워를 감소시킬 수 있다는 장점이 있다.

2. 이진 비트평면 영상

영상인식의 분야에서 딥러닝 시스템으로 널리 활용되고 있는 CNN의 일반적인 구조를 그림 1에 나타냈다[1]. 본 논문에서는 이진영상과 이진커널을 이용한 CNN 연산 알고리즘을 제안하였다. 따라서 영상인식의 분야에서 사용되는 컬러 및 그레이스케일 영상은 이진영상으로 변환되어야 한다. 컬러영상의 경우에는 그레이스케일 영상에 해당하는 R, G, B 영상으로 변환한 후, 각각의 영상을 다시 이진영상으로 변환하는 과정을 거쳐야 한다.

그레이스케일 영상을 이진영상으로 변환하는 기본적인 방법은 문턱치(threshold) 처리이다. 문턱치 처리 방식으로 처리된 이진영상은 그레이스케일 영상에서 특정 화소의 그레이레벨에 해당하는 정보는 모두 소실시키고, 단지 물체 영역과

배경 영역을 구분하는 방식이므로 CNN의 입력 영상으로 사용할 경우, 그 인식 결과의 정확도는 크게 제한된다.

그레이스케일 영상을 이진 영상으로 변환하는 두 번째 방법은 기준평면 영상으로 분해하는 방법이다[4,5]. 256그레이스케일 영상에서는 최대 255개의 이진 입력영상 패턴이 존재한다. 이는 처리시간의 측면에서 장점이 되지 않는 것으로 해석할 수도 있으나, 병렬로 처리가 가능한 경우에는 처리시간의 측면에서 매우 큰 장점이 될 수 있다[4,5]. 그러나 컴퓨팅 파워가 크지 않은 임베디드 시스템에 병렬처리 방식을 적용하는 것에는 비용과 칩설계의 측면에서 다양한 문제가 제기된다.

그레이스케일 영상을 이진 영상으로 변환하는 세 번째 방법은 비트평면 영상으로 분해하는 방법이다. 화소당 8비트의 영상을 8개의 비트평면 영상으로 분해하는 방법에서는 특정비트의 값이 0 또는 1인지의 여부에 따라 0 또는 1을 할당하는 방법이 주로 사용되고 있으나, 이진 CNN에서의 연산이므로 1과 -1을 할당하는 이진화를 수행한다[6]. 비트평면으로 변환한 영상을 이진화하는 방법은 다음과 같다.

n번째 비트평면 영상은 영상의 그레이레벨 값을 2진수로 변환한 후, n번째 비트가 1인 화소에는 1을, n번째 비트가 0인 화소에는 -1을 할당하는 방식이다.

문턱치 처리 방식에서는 1개의 이진영상이 생성되지만 대부분의 정보가 소실되는 문제가 있고, 기준평면 영상 방식에서는 이진영상이 256개로 너무 많이 생성되는 문제가 있다.

따라서 본 논문에서 제안한 이진 CNN 시스템의 구현을 위한 연산처리 알고리즘에서는 그레이스케일 영상을 비트평면 영상으로 분해하는 방법을 적용하였다.

3. 이진 CNN 연산 알고리즘

그림 1에 나타난 바와 같이 CNN은 컨볼루션 레이어(Convolutional Layer), 풀링 레이어(Pooling Layer), 풀리 커넥티드 레이어(Fully-Connected Layer)로 구분된다. 딥러닝 시스템 설계 과정에서 활성화함수 ReLU가 레이어 사이에 추가되기도 한다.

본 논문에서는 컨볼루션, 풀링, ReLU에 관한 연산을 화소당 1비트로 할당되는 이진영상과 웨이트(weight : 가중치)이 1과 -1로 구성되는 이진 커널을 사용하여 수행하는 것이다[6,7].

컬러영상은 R, G, B 성분에 해당하는 256 그레이레벨 영상으로 변환한다. 각각의 256 그레이레벨 영상은 8개의 이진 비트평면 영상으로 분해된다. 그림 2에 4 그레이레벨 영상을 2개의 이진 비트평면 영상으로 분해한 결과를 나타냈다.

특징추출 단계에서 이진영상으로 분해된 영상을 미리 정의된 커널로 컨볼루션 처리한다. 이진 연산을 위해서는 컨볼루션 과정에서 사용하는 커널의 웨이트도 1비트로 설정되어야 한다. 커널의 웨이트는 1 또는 -1($W \in \{+1, -1\}$)로 표현된다.

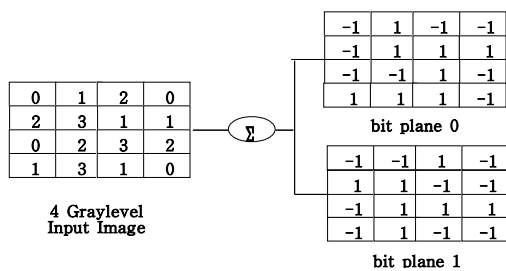


그림 2. 비트평면 영상 분해
Fig. 2. Bit-Plane image decomposition

3 x 3 커널의 경우, 512개의 경우의 수가 존재한다. 그림 3에 적용 가능한 3 x 3 커널 512

개 중 일부를 나타냈다.

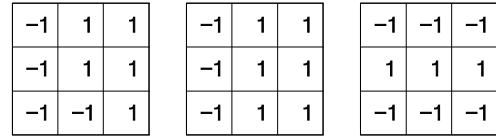


그림 3. 이진 3 x 3 커널
Fig. 3. Binary 3 x 3 Kernel

이진 비트평면 영상과 이진커널과의 컨볼루션은 $1 \times 1 = 1$, $1 \times (-1) = -1$, $(-1) \times 1 = -1$, $(-1) \times (-1) = 1$ 으로 구성되는 4개의 연산 결과의 합을 구하는 처리이다. 따라서 이 연산은 그림 4에 나타난 바와 같이 XNOR와 업다운 카운터로 대체할 수 있다.

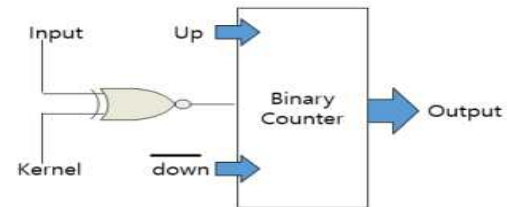


그림 4. 이진 컨볼루션 논리회로
Fig. 4. Binary convolution logic circuit

CNN에서 연산시간, 메모리 용량, 전력소모 등에 직접적으로 관계되는 연산은 컨볼루션이다. 컨볼루션 연산의 경우, 본 논문에서는 XNOR 게이트와 카운터를 이용하는 방법을 적용하였다. 그림 1에 나타난 CNN에서의 풀링과 ReLU 처리는 각각 최대치를 선택하고 음수를 구별하는 방법이므로 비교기를 사용하여 연산한다. 최대치를 선택하고 음수를 0으로 처리하는 연산은 가산 및 승산 연산이 요구되지 않으므로 별도의 이진 처리는 의미가 없다. Fully-Connected 레이어에서도 역시 가산과 승산 연산은 없다.

256 그레이레벨 영상에서는 8개의 비트평면 영상이 생성된다. CNN을 이용한 영상인식 분야에서 가장 의미 있는 영상은 MSB에 해당하는 비트평면 영상이다. 일반적으로 물체의 형상이 유지되는 비트평면 영상은 상위 4개 영상이다.

따라서 컴퓨팅 파워와 직접적으로 관계되는 분야에서는 인식 대상이 되는 영상에 따라 비트평면 영상 8개 모두를 사용하지 않을 수도 있다. 이는 계층적으로 CNN 시스템을 구성할 수 있다는 의미므로 매우 큰 장점으로 활용할 수 있다.

4. 실험

ILSVRC(ImageNet Large Scale Visual Recognition Challenge) 2017 데이터베이스의 테스트 영상을 대상으로 본 논문에서 제안한 CNN 고속연산 알고리즘의 유용성을 검증하였다. PC 환경에서 매트랩을 사용하여 실험을 수행하였다. 그림 5는 본 실험에서 사용한 테스트 영상의 일부이다.



그림 5. 테스트 영상(ILSVRC2017 #229, #653)
Fig. 5. Test image(ILSVRC2017 #229, #653)

컬러 입력영상으로부터 얻은 R, G, B 영상 각각은 이진화함으로써 비트평면 영상으로 변환한다. 그림 6에 각각 R,G,B 성분에 해당하는 영상을 나타냈다.

R,G,B 성분 영상으로 분해된 그레이스케일 영상은 다시 이진영상인 비트플랜 영상으로 변환한다. 그림 7에 테스트 영상에 해당하는 비트평면 영상을 나타냈다. 여기서 LSB 평면에 가까운 영상은 불규칙한 화소 배열이다. MSB 평면에 가까운 영상일수록 물체의 형상이 선명하다.

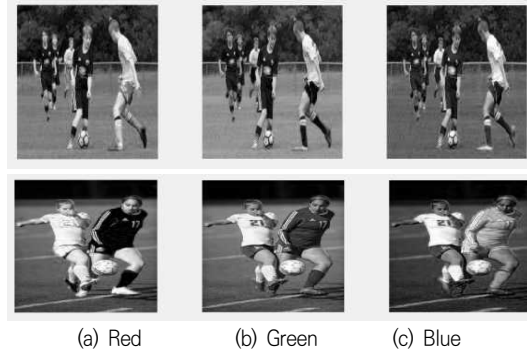


그림 6. R,G,B 컴포넌트 영상
Fig. 6. R,G,B component image

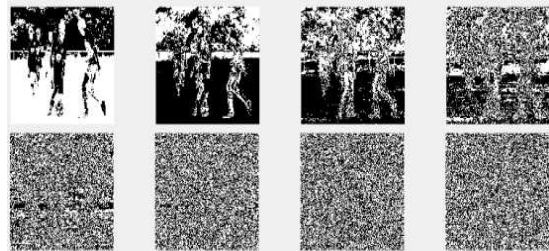
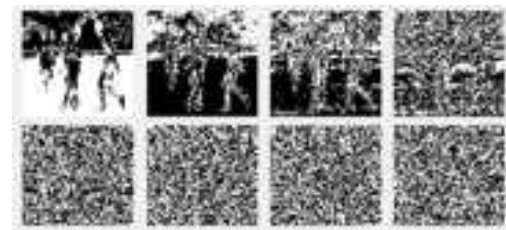


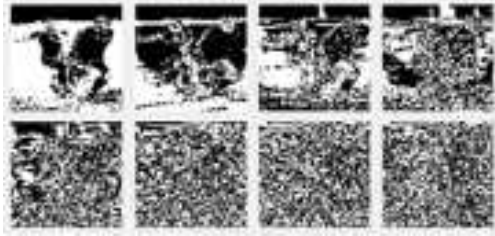
그림 7. #229 영상에서의 R성분 비트평면 영상(비트 평면 7~비트평면 0)
Fig. 7. R component bit-plane Image in #229(bit-plane 7-bit-plane 0)

#229 비트평면 영상 각각을 $k1=[-1 \ 1 \ -1 ; 1 \ 1 \ 1 ; -1 \ 1 \ -1]$ 로, 그리고 #653 비트평면 영상 각각을 $k2=[1 \ 1 \ 1 ; -1 \ -1 \ -1 ; 1 \ 1 \ 1]$ 커널로 컨볼루션한 결과를 그림 8에 나타냈다. XNOR 논리연산과 카운터만을 사용하는 이진 연산이다.

컨볼루션 연산결과는 부호에 따라 양수인 경우는 +1을 할당하고, 음수인 경우에는 0을 할당한다. 이 연산은 CNN에서 주로 사용되는 ReLU 처리이다.



(a) bit-plane 7~0 of IILSVRC2017 #229

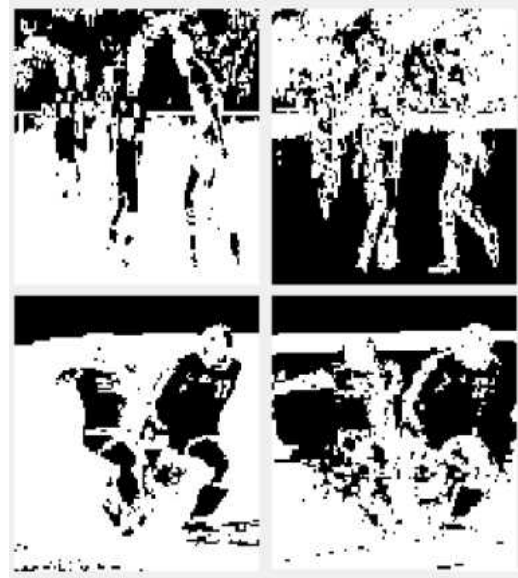


(b) bit-plane 7~0 of ILSVRC2017 #653

그림 8. 비트평면 영상의 이진 컨볼루션
Fig. 8. Binary convolution of bit-plane image

CNN은 다수의 컨볼루션 레이어로 설계되고, 컨볼루션 레이어 각각에서는 다수의 커널을 사용한다. 따라서 고성능 CPU와 GPU가 탑재되지 않은 스마트폰 등의 임베디드 시스템에서 CNN을 구현하기 위해서는 컨볼루션 연산을 축소하는 것이 필요하다. 그림 9에 나타난 바와 같이 LSB에 해당하는 하위 비트평면 영상의 일부분은 물체의 형상과는 무관하므로 MSB에 해당하는 상위 계층 중에서 일부만을 선택하여 사용하는 것이 가능하다. 이러한 특징은 이진 CNN을 계층적으로 구성할 수 있다는 측면에서 활용도가 매우 높은 부분이다.

CNN에서의 핵심 연산은 컨볼루션이다. 컨볼루션 외에 네트워크 설계과정에서 주로 사용되는 연산은 ReLU 연산과 서브샘플링을 수행하는 풀링 연산이다. 서브샘플링을 이용한 차원축소 과정에서 사용되는 연산은 max-pooling이다. 일반적으로 4개의 화소값 중에서 최대치 1개를 선택하는 연산이 CNN에서 널리 사용된다. 본 논문에서 제안한 방법은 이진영상과 이진커널을 이용하는 방식이므로 컨볼루션 결과에서 가장 큰 값은 1이 된다. 따라서 4개의 데이터 중 어느 것 1개라도 1인 경우에는 1이다. 4입력 OR게이트로 연산이 가능하다. 풀링 연산결과를 그림 9에 나타냈다. #229와 #653 영상에 해당하는 각각의 비트평면 영상 중에서 상위 MSB 평면을 대상으로 이진 풀링 연산을 수행한 결과이다.



(a) bit-plane 7 (b) bit-plane 6

그림 9. 상위 비트평면 영상의 이진 풀링
Fig. 9. Binary pooling of MSB bit-plane Image

4. 결 론

본 논문에서는 CNN에서 수행되는 컨볼루션, 풀링, ReLU 연산 등을 이진 연산으로 수행함으로써 처리시간을 축소시키는 알고리즘을 제안하였다. 이진 연산이 가능한 것은 이진 영상과 이진 커널을 사용하기 때문이다. 실험을 통하여 비트평면 영상과 이진 커널을 사용할 경우, XNOR, 카운터, OR 논리연산으로 CNN 연산을 수행할 수 있음을 확인하였다.

이진 영상과 이진 커널을 사용하는 것은 컴퓨팅 파워 및 연산시간에서는 큰 장점으로 작용하지만 고성능 CPU와 GPU를 요구하는 대형 CNN 분야에는 적용이 불가능하다.

이진 CNN 연산 알고리즘은 스마트폰 등과 같은 임베디드 시스템에 다양하게 적용할 수 있다. 향후, 연구주제는 스마트폰 및 지능형 CCTV에 적용 가능한 상황인지 CNN 시스템을 이진 시스템으로 개발하는 것이다.

REFERENCES

[1] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," *International Symposium in Circuits and Systems (ISCAS)*, IEEE, 2010.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information processing Systems 25*, NIPS, 2012.

[3] H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng., "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM, 2009.

[4] Jong-Ho Choi, "Algorithm of Morphological Multimode Binary Shape Decomposition," *Journal of KITE*, Vol. 36S, No. September, April, 1999.

[5] Jong-Ho Choi, "Parallel-Addition Convolution Algorithm in Grayscale Image," *Journal of KIIECT*, Vol. 10, No. 4, August, 2017.

[6] Rastegari, Mohammad, et al., "Xnor-net: Imagenet classification using binary convolutional neural networks." *European Conference on Computer Vision*. Springer, Cham, 2016.

[7] HUBARA, Itay, et al. "Binarized neural networks: Training neural networks with weights and activations constrained to+ 1 or-1." *arXiv preprint arXiv:1602.02830*, 2016.

저자약력

최 종 호(Jong-Ho Choi)

[중신회원]



- 1982년 2월 : 중앙대학교 전자공학과(공학사)
- 1984년 2월 : 중앙대학교 대학원 전자공학과(공학석사)
- 1987년 2월 : 중앙대학교 대학원 전자공학과(공학박사)
- 1990년 3월 ~ 현재 : 강남대학교 IoT전자공학과 교수

〈관심분야〉

영상처리, 컴퓨터시각, 딥러닝