컨볼루션 신경망의 병렬화 성능향상을 위한 OpenCL 기반 워크그룹 스케줄링 기법

박상수, 정기석*
한양대학교

po092000@hanyang.ac.kr, *kchung@hanyang.ac.kr

Highly Effective Work-group Scheduling Techniques for Convolution Neural Network using OpenCL

Park, Sang Soo and Chung, Ki Seok*
Hanyang University, Seoul, Korea

요 약

최근 OpenCL 과 같은 병렬 연산 프레임워크를 사용해 GPU 에서 컨볼루션 신경망 (CNN, Convolutional Neural Network)의 처리 성능을 높이는 연구가 활발히 이뤄지고 있다. 하지만 일반적인 컨볼루션 구조에 존재하는 불규칙적인 Input feature map 의 조합은 그룹 단위의 스레드 스케줄링을 통해 메모리 접근 오버헤드를 최소화하는 GPU 의 HW 구조에 적합하지 않다. 본 논문은 GPU 에서 OpenCL 워크그룹 (Work-group) 단위의 스케줄링 기법에 적합한 규칙적인 컨볼루션 신경망 연산 구조를 제안한다. 제안하는 구조를 글자 인식 벤치마크인 LeNet-5 에 적용한 결과 최대 37.26 배의 성능 향상과 26.4 배의 전력소모 개선 효과를 얻을 수 있었다.

Ⅰ. 서 론

컨볼루션 신경망은 인간의 뇌가 패턴을 인식하는 방법을 모사한 인공신경망의 한 종류로 최근 영상, 음성인식, 자연어 처리 등 다양한 분야에 적용되어 사용되고 있다 [1-3]. 컨볼루션 신경망은 행렬 곱을 이용하여 분류하고자 하는 이미지의 특징을 추출하고, 심층 신경망을 통해 추출된 특징 값을 분류하는 과정을 통해 높은 정확도의 분류 성능을 얻을 수 있다. 행렬 곱 기반의 컨볼루션 연산은 데이터 단위의 병렬처리에 적합한 구조를 가지고 있기 때문에, GPU (Graphic Processing Unit)를 활용할 경우 높은 성능 향상을 얻을 수 있다.

하지만 기존 컨볼루션 신경망의 구조는 메모리 접근 레이턴시를 감추기 위해 그룹 단위의 스레드를 활용하는 GPU 의 하드웨어 자원을 충분히 활용하기 어렵다. 따라서 본 논문은 더미 (dummy) 연산을 기반으로 한 새로운 컨볼루션 신경망 구조를 제안하고, 개방형 병렬 프로그래밍 프레임워크인 OpenCL [4]의 스레드 관리 인터페이스를 활용해 GPU 에서 병렬화 하였다.

Ⅱ. 본론

2.1 컨볼루션 신경망 구조

본 논문에서는 컨볼루션 신경망의 대표적인 모델인 LeNet-5[5]의 병렬화를 진행하였다. LeNet-5 는 광학 문자 인식 (OCR, Optical Character Reader)에 사용되는 인공신경망으로, 3 개의 컨볼루션 레이어 (Convolution layer), 2 개의 서브 샘플링 레이어 (Sub-sampling layer), 2 개의 풀리커넥티드 레이어 (Fully-connected layer)로 구성된다. LeNet-5 는 컨볼루션과 서브샘플링 레이어를 통해 이미지의 특징을 추출하고 분류를 한다. 추출된 특징은 풀리커넥티드 레이어를 거치면서 0~9 사이의 숫자 중 하나로 분류된다.

컨볼루션 레이어는 입력 특징 (Input feature map)과 컨볼루션 커널 (Convolution kernel), 바이어스 (Bias)를 통해 출력 특징 (Output feature map)을 계산한다. 컨볼루션 연산은 입력 특징과 컨볼루션 커널의 성분 별 곱셉으로 이뤄진다. 입력층과 컨볼루션 커널의 성분 별 곱셉을 계산하고 바이어스를 더한 후 활성함수를 적용하여 출력 특징을 계산한다.

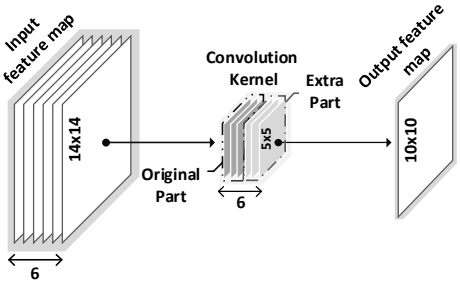
2.2 제안하는 더미 연산 기반 컨볼루션 구조

[표 1] LeNet-5 의 C2 출력 특징 생성 조합

입력층 인덱스 \ 출력층 인덱스	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

LeNet-5 의 컨볼루션 계층 2 (C2)는 출력 특징을 추출하기 위한 입력 특징과 컨볼루션 커널의 연결이 불규칙한 특성이 존재한다. 표 1 은 C2 의 출력층을 생성하기 위한 입력 특징과 컨볼루션 커널의 조합 방법을 나타낸다. 각 열은 특정 출력층을 생성하는데

필요한 입력 특징을 의미하며, 각 입력 특징은 5x5 크기의 가변적인 깊이를 갖는 컨볼루션 커널에 연결된다. 예를 들어, 표 1 의 14 번째 출력 특징은 깊이 4, 5x5 크기의 컨볼루션 커널을 사용하여 14x14 크기의 0, 2, 3, 5 번째 입력 특징을 필요로 한다. 이러한 불규칙한 특성은 GPU 가 워크 그룹에 동일한 수의 스레드를 할당하여 커널을 병렬로 실행하는 것을 어렵게 하기 때문에 GPU 의 성능을 최대한으로 사용하는 것이 어렵다



[그림 1] 제안하는 더미 연산을 사용한 C2

제안하는 방법의 핵심 아이디어는 더미 연산을 사용하는 것으로, 더미 연산은 연산의 결과로 항상 0 을 갖는다. 더미 연산을 삽입하면 출력 특징의 종류에 상관없이 연산 량을 동일하게 만들 수 있기 때문에 C2 는 동일한 갯수의 스레드를 갖는 워크 그룹과 함께 병렬화하는 것이 가능하며, 제안하는 방법은 그림 2 와 같다.

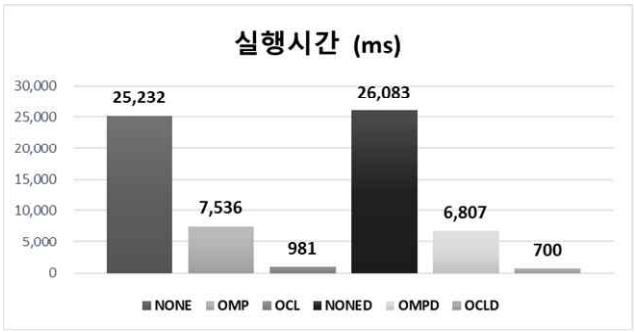
제안하는 방법은 표 1 의 해당하는 원래의 부분 (Original Part)과 추가적인 부분 (Extra Part)으로 나눠진다. 원래의 부분은 표 1 의 컨볼루션 커널에 해당하는 값을 가지며, 추가적인 부분은 모두 0 값을 갖는다. 예를 들어 0 번째 출력층을 구하기 위해서, 원래의 부분은 표 1 의 ‘ X’ 마크가 있는 3 개의 컨볼루션 커널을 포함하고 추가적인 부분은 ‘ X’ 가 없는 3 개의 컨볼루션 커널에 해당된다.

본 논문의 실험에서는 AMD APU 를 타켓 플랫폼으로 사용하였으며, APU 는 64 개의 워크 아이템 (Work-item)을 하나의 워크 그룹에 할당하기 때문에 워크 그룹의 크기가 64 의 배수여야 하며, 256 일 때 성능이 가장 좋은 것을 확인하였다.

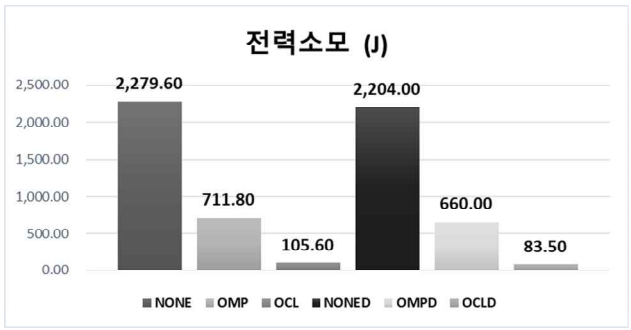
2.3 실험 결과 및 분석

본 논문에서는 AMD 社의 APU A10 에서 실험을 진행하였다. 동일한 반도체 다이에서 CPU 와 GPU 가 집적된 APU 는 CPU 가 PCIe (PCI Express) 버스를 통해 GPU 와 외부로 연결된 플랫폼보다 좋은 성능을 얻는 것이 가능하며, 본 논문의 실험에서는 APU A10 7870K 와 16GB DDR3 메모리를 사용하였다.

실행 시간을 비교하기 위해 표 1 의 기존 방법과 더미 연산을 사용하는 제안하는 방법에서 각각 3 종류의 병렬화 방법을 적용하였다. 먼저 기존의 방법에서는 C/C++ 코드 (NONE)와 OpenMP 를 이용하여 병렬화 한 코드 (OMP), 그리고 OpenCL 을 이용하여 병렬화 한 코드를 구현하였다 (OCL). 다음으로 더미 연산을 사용하는 제안하는 방법에서는 C/C++ 코드 (NONED), OpenMP 를 이용하여 병렬화 한 코드 (OMPD), 그리고 동일한 워크 그룹의 크기를 사용하는 OpenCL 코드 (OCLD)를 구현하였다. 그림 2 와 그림 3 은 각 코드의 실행시간과 전력소모를 나타낸다. 기존 방법에서의 OCL 대비 NONE 의 실행시간/전력소모는 25.72/21.89 배, 제안하는 방법에서의 OCLD 대비 NONED 의 실행시간/전력소모는 37.26/26.4 배 향상되었다.



[그림 2] 실행시간 비교



[그림 3] 전력소모 비교

Ⅲ. 결론

본논문에서는 CNN 의 컨볼루션 연산을 GPU 친화적으로 병렬화 하기 위해 더미 연산에 기반한 컨볼루션 계층 구조를 제안한다. 이를 통해 GPU 의 하드웨어 자원을 충분히 활용하는 OpenCL 워크 그룹 스케줄링 기법 적용이 가능하다. 제안하는 컨볼루션 연산 구조를 대표적은 글자 인식 벤치마크인 LeNet-5 에 적용한 결과, CPU 대비 최대 37.26 배의 실행 시간, 26.4 배 전력소모가 개선된 결과를 얻을 수 있었다.

ACKNOWLEDGMENT

이 논문은 2015 년도 정부 (교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업 임 (NRF-2015R1D1A1A09061079)

참 고 문 헌

[1] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[2] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[3] Carnimeo, Leonarda. "A CNN-based Vision System for Pattern Recognition in Mobile Robots." Proc. of the 15th IEEE European Conf. on Circuit Theory & Design, Espoo, Finland. 2001

[4] OpenCL, https://www.khronos.org/opencl/

[5] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.