자원 제약 환경에서 SIMD 를 활용한 신경망 연산 가속

정세현 [1], 강지원 [1], 이윤서 [1], 구본욱 [1], 황정민 [1], 오현영 [2*]
[1] 가천대학교 AI · 소프트웨어학부 학부생
[2] 가천대학교 AI · 소프트웨어학부 교수

{jjsshh3116, gjwon123, lyssva345, chris3565, hjmin0406, hyoh}@ gachon.ac.kr

Accelerating Neural Network Inference using SIMD in Resource-Constrained Environments

Se-Hyeon Jeong, Gi-won Kang, Yun-Seo Lee, Bon-Wook Gu, Jeong-Min Hwang, Hyunyoung Oh
Dept. of AI · Software, Gachon University

요 약

본 연구는 자원 제약적 임베디드 시스템에서 신경망 연산의 효율성을 극대화하기 위해 SIMD(Single Instruction Multiple Data) 기술을 활용한 최적화 기법을 제안한다. 기존 연구들이 주로 합성곱 연산에 집중된 것과 달리, 본 연구는 신경망의 전체 연산 구간에 SIMD 최적화를 적용하고, 범용 DNN 프레임워크인 Darknet 을 기반으로 다양한 모델에 적용 가능한 방법론을 적용하였다. Raspberry Pi 3B+를 테스트베드로 활용하여 다양한 CNN 모델에 대한 성능 평가를 수행하였으며, 최대 55.2%의 성능 향상을 달성하였다. 또한, SIMD 레지스터 활용도와 연산 속도 간의 상관관계를 분석하여 최적의 구현 전략을 도출하였다.

1. 서론 및 배경

최근 임베디드 시스템과 IoT 디바이스에서의 AI 애플리케이션 수요가 급증함에 따라, 제한된 컴퓨팅 자원 환경에서 효율적인 신경망 연산 구현의 중요성이 부각되고 있다. SIMD(Single Instruction Multiple Data)는 하나의 명령어로 여러 데이터 요소를 동시에 처리할 수 있는 병렬 처리 방식으로, 딥러닝 연산의 가속화에 널리 적용되고 있다. 기존 연구들은 주로 SIMD 기술을 활용하여 합성곱 연산을 최적화하는 데 집중하였으나[1,2], 본 연구는 SIMD 최적화 기법을 신경망의 전체 연산 구간에 적용하고, 범용 DNN 프레임워크인 Darknet[4]을 기반으로 다양한 모델에 적용 가능한 최적화 방법론을 제안한다. 특히, Raspberry Pi 3B+를 테스트베드로 활용하여, 실제 임베디드 환경에서의 성능 향상을 정량적으로 분석하였다. Raspberry Pi 3B+는 쿼드 코어 ARM Cortex-A53 프로세서와 1GB RAM 을 탑재한 저전력, 저비용 싱글보드 컴퓨터로, Armv8-A 아키텍처(AArch64)의 NEON SIMD engine 을 지원한다[3]. 이러한 하드웨어 특성을 고려하여, 본 연구에서는 32 개의 128-bit NEON SIMD 레지스터를 활용한 최적화 기법을 개발하고, SIMD 구현 전략에 따른 성능 변화를 심층적으로 조사하였다.

2. 실험 및 평가

실험은 Raspberry Pi 3B+ (Quad-core ARM Cortex-A53, 1GB RAM)에서 수행되었으며, Raspbian OS 와 GCC 8.3.0 컴파일러를 사용하였다. 벤치마크 모델로는 LeNet, VGG-7, CIFAR, Darknet, AlexNet 을 선정하였고, LeNet 모델에는 MNIST 데이터셋을, 나머지 모델에는 CIFAR-10 데이터셋을 사용하였다.

3.1 전체 모델 성능 평가

(표 1) SIMD 적용 전과 후 성능 비교

모델	Darknet(s)	Darknet_simd(s)	향상률(%)
LeNet	0.008810	0.007428	15.7
VGG-7	0.058325	0.026111	55.2
CIFAR	3.100413	2.162913	30.2
Darknet	1.762604	1.307395	25.8
AlexNet	3.632001	2.538073	30.1

SIMD 최적화를 적용한 Darknet_simd 가 모든 모델에서 성능 향상을 보였다. 특히 VGG-7 모델에서 55.2%의 가장 큰 향상을 보였다. 이는 VGG-7 모델의 구조적 특성, 특히 연속적인 3x3 합성곱 연산이 SIMD 연산과 잘 부합하기 때문으로 분석된다. 반면, LeNet 모델의 경우 상대적으로 낮은 15.7%의 향상률을 보였는데, 이는 LeNet 의 단순한 구조와 작은 입력 크기로 인해 SIMD 연산의 이점을 충분히 활용하지 못했기 때문으로 추정된다. 이어지는 실험에서 성능 향상폭이 가장 큰 VGG-7 의 경우에 심층 분석을 위해 계층별 break-down 및 레지스터 활용 개수에 따른

* 교신저자

성능 차이를 분석하였다.

3.2 VGG-7 에 대한 계층별 성능 분석

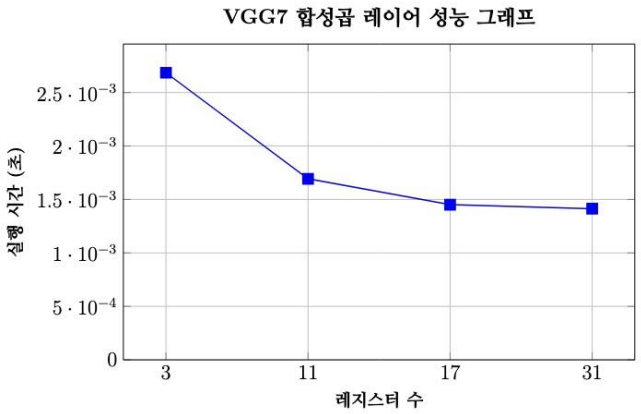
(표 2) 네트워크 계층별 성능 비교 (VGG-7)

Layer	Darknet(s)	Darknet_simd(s)	향상률(%)
Conv	0.014449	0.003559	75.4
Max-pool	0.000157	0.000082	47.8
F.C.	0.000105	0.000105	0.0

심층 분석을 위해 연산 계층별로 성능 향상률을 비교했다. VGG-7 모델을 기준으로 각 계층별 성능을 비교하였으며, <표 2>에는 성능 차이가 큰 계층들을 발췌하여 제시하였다. 합성곱 계층에서 75.4%의 가장 큰 성능 향상을 관찰하였다. 이는 합성곱 연산의 병렬 처리 가능성이 높고, 데이터 지역성(data locality)이 우수하여 SIMD 연산의 이점을 극대화할 수 있기 때문으로 분석된다. 최대 풀링 계층에서도 47.8%의 상당한 성능 향상을 보였다. 이는 풀링 연산의 단순성과 규칙적인 메모리 접근 패턴이 SIMD 연산과 잘 부합하기 때문으로 분석된다.

반면, 완전 연결 계층에서는 성능 향상이 거의 관찰되지 않았다. VGG-7 외의 모델에서도 F.C.의 성능 향상폭이 가장 낮았다. 이는 NEON SIMD 를 사용하기 위해 float32x4_t 같은 데이터 타입을 사용하는데, F.C. 계층의 경우에는 대규모 가중치(weight) 행렬로 인해 SIMD 레지스터에 로드하고 저장하는 과정이 다수 추가되면서 캐시 미스로 인한 메모리 접근 비용이나 정렬 비용이 발생하는 것으로 판단된다.

3.3 SIMD 레지스터 활용도 분석



(그림 1) 레지스터 개수에 따른 연산 속도 상관관계.

VGG-7 모델의 합성곱 연산을 SIMD 레지스터의 개수를 달리하여 구현하고 성능을 비교하였다. 레지스터 개수가 증가함에 따라 연산 속도도 함께 증가하는 경향을 보였다. 그러나 일정 수준 이상으로 레지스터 개수를 늘리면 성능 차이가 없거나 오히려 약간 느려지는 현상이 관찰되었다. 이는 과도한 레지스터 사용은 레지스터 스필링(spilling)을 유발하여 메모리 접근 비용을 증가시키기 때문으로 판단된다.

3. 한계점 및 향후 연구

본 연구에서 제안한 SIMD 기반 신경망 연산 가속화 기법은 모델 구조에 따른 성능 향상의 불균형성, 완전 연결 계층에서의 제한적 성능 향상, SIMD 레지스터 활용도 증가에 따른 성능 향상의 포화 현상 등의 한계점을 보였다. 향후 연구에서는 다양한 모델 구조에 대한 최적화 전략 개발, 희소 행렬 압축 기법과 블록 기반 행렬 곱셈 등을 통한 완전 연결 계층 최적화 등을 진행할 계획이다. 또한, Raspberry Pi 3B+의 GPGPU 로 활용 가능한 Videocore 를 사용한 가속 방법에 대해서도 연구를 진행할 예정이다.

4. 결론

본 연구에서는 자원 제약적 환경에서 SIMD 를 활용한 신경망 연산 가속화 기법을 제안하고, 다양한 CNN 모델에 대해 그 효과성을 검증하였으며, 최대 55.2%의 성능 향상을 달성하였다. 특히 합성곱 연산과 최대 풀링 연산에서 탁월한 성능 향상을 보였으나, 완전 연결 계층에서의 제한적 성능 향상 등 몇 가지 한계점도 확인되었다. 이러한 결과는 임베디드 시스템에서의 AI 애플리케이션 구현에 있어 SIMD 최적화가 중요한 역할을 할 수 있음을 시사하며, 향후 다양한 모델 구조와 Videocore 에 대한 추가 연구를 통해 제안된 기법의 일반화 가능성을 검증하고 임베디드 AI 시스템의 성능과 효율성을 더욱 향상시킬 수 있을 것으로 기대된다.

사사문구

이 논문은 2024 년도 정부(산업통상자원부)의 재원으로 한국산업기술기획평가원의 지원(No. RS-2024-00406121, 자동차보안취약점기반위협분석시스템개발(R&D))과 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(No. RS-2022-00166529)을 받고 과기정통부 정보통신기획평가원의 정보보호핵심원천기술개발사업(No. RS-2024-00337414)으로 수행한 결과임.

참고문헌

[1] Al Jbaar et al., "SIMD Implementation of Deep CNNs for MYOPIA Detection on A Single-Board Computer System," Eastern-European Journal of Enterprise Technologies, 2023

[2] Lee, Sung-Jin et al., "Efficient SIMD Implementation for Accelerating Convolutional Neural Network," Proceedings of the 4th International Conference on Communication and Information Processing, 2018

[3] ARM. (n.d.). Programmer's guide for ARMv8-A. ARM Community. https://community.arm.com/arm-community-blogs/b/architectures-and-processors-blog/posts/programmer-s-guide-for-armv8-a (Accessed on September 22, 2024)

[4] Reddie, P. (n.d.). Darknet: Open Source Neural Networks in C. PJ Reddie. https://pjreddie.com/darknet/ (Accessed on September 22, 2024)