Edge 디바이스를 위한 NPU 기반 다중 FIR 필터 구현 및 성능 분석

임수환, 박진욱, 신동엽*

*한국전자기술연구원

ish0723@keti.re.kr, jinuk.park@keti.re.kr *dongyeob.shin@keti.re.kr

NPU-Based FIR Filter Implementation and Performance Evaluation on Edge Devices

Lim Su Hwan, Park Jin uk, Shin Dong Yeob*

*Korea Electronics Technology Institute

요 약

본 논문은 향후 대부분의 Edge 디바이스에 NPU(Neural Processing Unit)가 탑재될 것으로 예상되는 가운데, 이러한 환경에서 다중 FIR 필터를 저지연으로 처리할 수 있는 알고리즘을 제안한다. 최근 CNN과 LLM 기반 인공지능 모델의 상용화가 활발해지면서, On-Device AI를 위한 경량화 모델이 주목받고 있으나, 이는 성능 저하라는 trade-off 문제를 동반한다. 이에 대한 대안으로 많은 연구자들이 NPU를 활용하고 있으며, 실제 CNN과 LLM의 전체 연산량 중 80~90%가 GEMM 연산으로 구성되어 있다는 사실에 기반해, FIR 필터 연산을 GEMM 형태로 변환하여 NPU에서 실행할 수 있도록 구성하였다. 제안한 방식은 입력 데이터를 행렬로 변환하고 다중 FIR 필터 계수와의 곱셈을 GEMM으로 구성함으로써, 기존 CPU 기반 FIR 처리 방식보다 병렬성과 성능을 향상시킨다. 실험은 ZCU104 FPGA에서 구현된 커스텀 NPU와 CPU 간 연산시간을 비교하여 진행되었으며, 필터 개수와 차수가 클수록 NPU 기반 방식이 더 낮은 지연을 보임을 확인하였다. 반면, 필터의 개수나 차수가 작을 경우 NPU와의 통신 오버헤드로 인해 CPU 대비 성능 이점이 줄어드는 한계도 발견되었다. 결론적으로, 본 논문은 FIR 필터 처리를 GEMM 형태로 재구성하여 NPU에서 효율적으로 실행할 수 있는 구조를 제시하고, 이를 통해 고성능 필터링을 저지연 환경에서 가능하게 하는 방안을 실험적으로 검증하였다.

Ⅰ. 서 론

최근 Convolutional Neural Network(CNN)와 Large Language Model(LLM)을 중심으로 인공지능 연구가 활발히 진행되고 있으며, 이를 상용화하려는 다양한 시도가 이루어지고 있다. 이에 따라 온디바이스AI라고 하는 경량화된 인공지능 모델을 Edge 디바이스에 적용하여 제한된 연산과 네트워크 환경에서도 인공지능을 활용하려는 연구와 개발이 활발히 이루어지고 있다. 그러나 인공지능의 경량화는 필연적으로 성능 저하와의 trade-off 관계를 동반하여, 오히려 제한된 환경에서의 실용성을 저해하는 역설적인 문제를 야기한다. 이에 따라 연구계와 산업계는 인공지능 신경망 가속기인 NPU(Neural Processing Unit)를 활용하는 방안을 주요 해결책으로 채택하고 있다. 이러한 흐름에 따라 가까운 미래에는 대부분의 Edge 디바이스에 NPU가 기본적으로 탑재될 것으로 예상된다.

한편, 산업 현장 및 주변기기에는 다양한 센서와 출력 장치가 사용되며, 이들은 일상생활의 다양한 환경 변수로 인해 입력 및 출력 신호에 노이즈를 유발한다. 이러한 환경에서는 반응 속도가 중요한 만큼, 필터의 정확도보다는 연산 지연을 줄이기 위한 성능 저하의 타협이 요구된다.

본 논문에서는 향후 Edge 디바이스에 탑재될 NPU를 활용하여, 다중 FIR 필터를 효율적으로 처리할 수 있는 저지연 고성능 알고리즘을 제안한다.

Ⅱ. 본론

Ⅱ-1. NPU의 일반적인 기능

NPU는 현재 다양한 곳에서 개발하고 있으며, 그 틀은 각양각색이다. 이에 본 논문에서 핵심으로 활용할 gemm연산은 대부분의 NPU들이 기능하고 있다고 가정한다. NPU를 사용하는 주요 이유와 사용처는 CNN와 LLM이다. 이 두 신경망 네트워크는 cpu기준으로 gemm연산에 할당된 시간이 80-90 % 에 달한다.[1][2] 그러므로 대부분의 NPU 개발자들은 이 문제를 해결하고자 노력하고 있으며 본 논문의 가정은 합리적이다.

Ⅱ-2. 다중 FIR Fileter와 Gemm연산

NPU를 활용하기 위해 input 데이터를 다중 FIR Filter에 적용시키기는 연산을 gemm으로 변경하고자 한다. input의 배열을 I, FIR Filter를 F라고 할 때, 기본적인 input과 FIR Filter의 연산은 (식1)과 같다.

$$output[n] = \sum_{k=0}^{N} F[k] \cdot I[n-k] \cdots (식1)$$

여기에서, I와 F를 (식2)와 같이 표현한다면, (식3)이 성립한다.

$$I = (i_0 i_1 \cdots i_n), F = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix} \cdots (식2)$$

$$output[n] = I \times F \cdots (식3)$$

다중 필터의 경우 (식4)와 같이 표현한다면 (식5)와 같이 구성하여 다중필터를 한 번에 연산할 수 있다.

$$MultF = (F_0 F_1 \cdots F_n) \cdots (식4)$$

$$output[n] = I \times MltF \cdots (식5)$$

그리고 이 연산은 NPU를 활용하면 저전력, 고속도 연산으로 처리가 가능하다.

II-3. 실험

본 논문에서 제안하는 알고리즘을 구현하여 cpu에서 수행하는 일반적인 filter를 적용하는 연산과 비교한다. 활용된 NPU는 직접 제작한 NPU로 다른 NPU보다 성능이 낮을 것으로 기대된다. zcu104 FPGA 보드의 ubuntu20.04에서 동작하도록 구현되었다. 실험은 filter의 개수를 32로 고정하고 filter의 차수의 변화에 따른 시간 지연의 비교와 filter의 차수를 360으로 고정하고, filter의 개수를 변화시켜 시간 지연을 비교하였다. 그 결과는 각각 그림 1과 그림 2에서 확인할 수 있다.

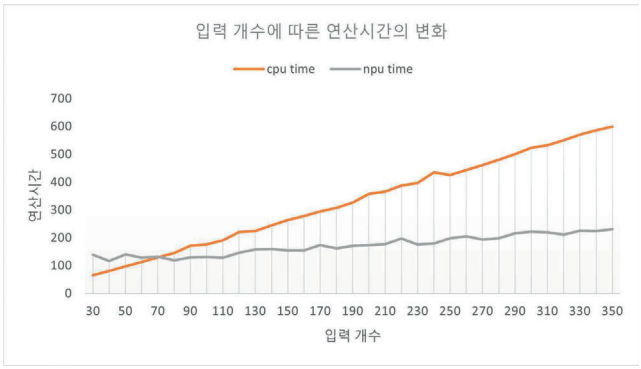

그림 1. filter 차수에 따른 시간 지연 비교

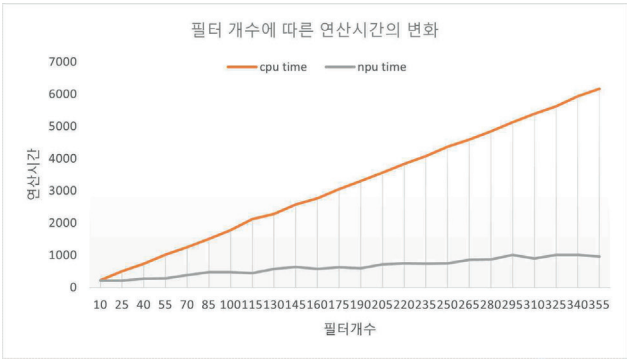

그림 2. filter 개수에 따른 시간 지연 비교

실험결과 filter의 개수와 차수가 높아질수록 본 논문에서 제안한 방식이 유리함을 알 수 있다.

Ⅲ. 결론

본 논문에서는 미래의 Edge 디바이스는 NPU를 포함하는 것이 일반적일 것이라 전망하며, 다중 FIR Filter를 적용할 수 있는 방법을 제안했다. 다중의 그리고 고차원의 filter를 저지연으로 사용함으로써 시간 자원 대비 높은 성능의 필터링을 사용할 수 있다.

심지어 본 논문에서 활용한 NPU는 커스텀하게 제작된 NPU로, 시중의 NPU들과 비교해본다면 낮은 성능일 것으로 예상된다. 그러므로 다른 NPU를 사용한다면 더 높은 성능을 기대할 수도 있다.

단, filter의 차수와 개수가 적은 경우, cpu와 npu사이의 통신 지연시간으로 인해 오히려 시간이 더 소요되는 것을 볼 수 있으므로 용도에 맞게 사용하는 것이 중요하다.

또한, NPU를 사용하는 환경에서는 시간 지연이 선형적으로 올라가는 것이 아니라, MAC의 구성에 영향을 받는다. 그러므로 NPU의 MAC 구성과 filter 및 filter의 차수를 적절히 고려하는 것이 좋다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부의 재원으로 정보통신기획평가원의 정보통신방송기술개발지원사업의 연구결과로 수행되었습니다(IITP-2020-0-01077, IoT 다중 인터페이스 기반의 데이터센싱, 엣지컴퓨팅 분석 및 데이터공유 지능형 반도체 기술 개발)

참 고 문 헌

[1] Xiang, Z., Wang, X., Ma, Z., Li, L., & Xu, C., "ConvBench: A Comprehensive Benchmark for 2D Convolution Primitive Evaluation," Proc. of IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 803 - 817, 2022.

[2] Zhang, X., He, B., Li, Y., & Shi, W., "Understanding the Performance Horizon of the Latest ML Workloads with NonGEMM Workloads," Proc. of IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 1 - 12, 2024.