온디바이스 SoC에서의 Attention 연산 이기종 프로파일링 및 Dual-Stage Backend Selection 모델 전략

손현호, 윤수연*

국민대학교, *국민대학교

aa000098@kookmin.ac.kr, *1104py@kookmin.ac.kr

Profiling Heterogeneous Attention and Dual-Stage Backend Selection Strategy for On-Device SoCs

Hyun Ho Son, Soo Yeon Yoon*

Kookmin Univ., *Kookmin Univ.

요 약

본 논문은 모바일 이기종 SoC 환경에서 온디바이스 LLM의 추론 성능 최적화를 위해, Attention 연산을 QKV Projection과 Attention Core로 세분화하여 각 단계별 최적 백엔드(GPU/NPU)를 동적으로 선정하는 Dual-Stage Backend Selection 모델을 제안한다. 이를 위해 텐서 형상과 정밀도에 따른 머신러닝 기반의 백엔드 선택 모델을 설계하여 하이브리드 실행 전략을 수립하였다. Rockchip RK3588 기반 실험 결과, 제안 기법은 INT8 정밀도 기준 단일 GPU 대비 최대 1.85배의 속도 향상을 기록하였다. 이를 통해 연산 특성에 따른 세밀한 이기종 분산 처리가 모바일 환경의 제한된 자원 효율을 향상시킬 수 있음을 입증하였다.

Ⅰ. 서 론

CPU와 GPU, NPU가 하나의 칩에 탑재된 현대 모바일 SoC 환경에서는 연산 유형이나 텐서 규격 등에 따라 하드웨어별 성능 차이를 보인다. 따라서 온디바이스 LLM 추론 시, 고정된 하드웨어 백엔드보다는 상황에 맞춰 최적의 백엔드를 선별하는 유연한 전략이 필요하다[1].

기존 연구들은 대개 LLM 추론 최적화를 위해 GPU나 NPU 중 하나의 가속기만을 활용하거나[2][3] 대역폭 확보를 목적으로 텐서 분할을 통한 다중 백엔드 동시 실행을 시도해왔다. 하지만 단일 백엔드 방식은 NPU의 패딩 비효율이나 GPU의 실행 오버헤드 같은 구조적 한계를 상쇄하지 못한다는 맹점이 있다. 텐서 분할 방식 또한 구현 난이도가 높고 동기화 비용 예측이 난해해 모바일 환경에서 범용적인 적용에 제약이 따른다[1].

본 연구에서는 기존 방식 대신 Attention 연산을 구성하는 세부 단계의 연산 특성과 하드웨어 병목을 설명할 수 있는 파생 변수를 바탕으로 각 단계별 최적의 하드웨어를 순차적으로 점유하는 직관적이고 효율적인 백엔드 선택 기법을 제안하고자 한다.

Ⅱ. 본론

2.1 모바일 SoC 구조와 특징

본 연구의 타겟 시스템인 Rockchip RK3588은 엣지 컴퓨팅을 위한 고성능 SoC로, CPU, GPU, NPU를 모두 내장한 이기종 구조를 갖는다. GPU(Mali-G610)는 FP16 연산과 대규모 병렬 처리에 유리하나 작은 텐서 연산 시에 커널 실행 및 동기화 오버헤드를 가진다.[1] NPU는 INT8 연산과 행렬 곱과 같은 정형화된 연산에 높은 효율을 보이지만, 지원하지 않는 연산자나 비정형 텐서 처리 시 호스트 CPU의 개입 및 패딩 오버헤드가 발생할 수 있다[1][3].

2.2 LLM Attention 연산 구조

Self-Attention 연산을 특성에 따라 QKV projection과 Attention Core 두 부분으로 나눌 수 있다. QKV Projection은 입력 텐서에 가중치를 곱하는 전형적인 밀집 행렬 연산으로, 높은 연산 강도의 특성을 보인다. Attention Core는 QK^T 연산 후 Softmax와 AV 연산을 수행하는 과정으로, 비선형 연산이 포함되며 시퀀스 길이에 따라 메모리 접근 패턴이 달라지는 메모리 중심(Memory-bound) 특성을 보인다[4]. 이러한 특성 차이로 인해 전체 레이어를 단일 백엔드에 할당하는 것보다, 각 단계별로 최적의 가속기를 선택하는 이기종 분산 처리가 유리할 수 있다.

Ⅲ. 최적화 실험 설계

3.1 입력 특징 벡터 구성

입력 특징 벡터의 구성은 표 1과 같이 구성하였다.

표 1. 입력 특징 벡터(X)의 구성 요소 및 정의

변수명	기호	정의 및 계산식
Sequence Length	n	입력 시퀀스 길이
Hidden Dimension	d	어텐션 은닉 차원
Precision	P	데이터 정밀도 (0: FP16, 1: INT8)
Compute Operations	FLOPs	총 부동소수점 연산 횟수
Memory Traffic	MemBytes	입출력 데이터의 총 바이트 수
Arithmetic Intensity	Intensity	FLOPs / MemBytes (연산/메모리)
Aspect Ratio	AR	n / d (텐서 형상 비율)
Score Matrix Size	ScoreSize	n^2 (중간 어텐션 맵 크기)
Padding Overhead	Pad_n, Pad_d	각 차원별 64-byte 정렬 낭비 비율

3.2 Dual-Stage Backend Selection 모델 구조 및 정의

Attention 연산의 이기종성을 반영하기 위해 QKV Projection과

Attention Core 단계의 최적 백엔드($\hat{y}$)를 각각 독립적으로 예측하는 이중 예측기(Dual-Predictor) 구조를 제안한다.

제안하는 시스템은 단일 분류기가 아닌, 두 개의 독립된 머신러닝 모델(M_{qkv}, M_{attn})을 사용하여 각 연산 단계에 최적화된 백엔드를 결정한다. 우선, 밀집 행렬 연산(Dense GEMM)이 지배적인 QKV Projection 단계는 연산 패턴의 특성을 반영하여 아래 식 (3.2)와 같이 가장 적합한 백엔드를 예측한다. 여기서 X는 워크로드의 특징 벡터이며, M_{qkv}는 이를 바탕으로 해당 단계의 지연 시간을 최소화하는 하드웨어($\hat{y}_{qkv}$)를 출력한다.

$$\hat{y}_{qkv} = M_{qkv}(X) \in GPU, NPU \tag{3.2}$$

비선형 연산과 불규칙한 메모리 접근 패턴을 포함하는 Attention Core 단계는 식 (3.3)처럼 별도의 모델을 통해 독립적으로 백엔드가 결정된다.

$$\hat{y}_{attn} = M_{attn}(X) \in GPU, NPU \tag{3.3}$$

Ⅳ. 실험 결과

4.1 학습 알고리즘 선정

각 분류기는 백엔드 선택 문제를 주어진 입력에 따라 최적의 클래스(GPU 또는 NPU)를 할당하는 이진 분류 문제로 정의하였다. QKV Projection 단계는 Random Forest가 가장 좋은 성능을 보였으며, Attention Core 단계에서는 K-Neighbors Classifier가 가장 높은 정확도를 보여 이중 예측기(Dual-Predictor)의 탑재 모델로 선정하였다.

4.2 시퀀스 길이(n) 변화에 따른 Prefill 단계 성능 분석

실험 결과 Prefill 단계에서 n >= 64 구간에서는 GPU가 NPU 대비 낮은 지연시간을 기록하였으며, 제안하는 Hybrid 모델(초록색 실선) 또한 GPU와 동일한 성능 곡선을 그렸다. 반면 n=40 구간에서는 GPU의 지연시간이 급격히 상승한 것과 달리, Hybrid 모델은 NPU 수준의 낮은 지연시간을 유지하는 결과가 관측되었다. 이는 제안 모델이 시퀀스 길이에 따라 지연시간이 최소화되는 백엔드를 적절히 스위칭하고 있음을 보여준다.

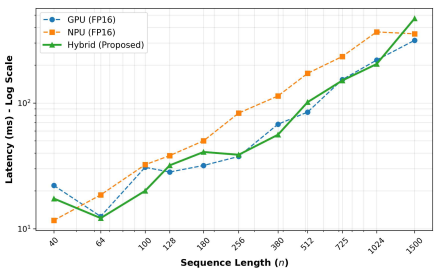

그림 3. INT8 정밀도에서의 Hybrid 전략 및 양자화 가속 효과.

4.3 Decode 단계에서의 정밀도별 Hybrid 전략 유효성 검증

Decode 단계(n=1)에서는 은닉 차원(d)과 정밀도(P)에 따른 Hybrid 기법의 속도 향상 비율을 측정하였다. 그림 2는 FP16 정밀도의 Decode 단계에서 은닉 계층 차원(d) 변화에 따른 각 백엔드 조합의 지연 시간을 비교한 결과이다. 실험 결과 d<=1024의 작은 차원 구간에서 NPU(QKV) -> GPU(Attn) 조합이 단일 NPU 대비 약 1.04배의 성능 향상을 보였다. 은닉 차원이 증가한 d=2048 구간에서는 GPU -> NPU 전략이 선택되었으나 시 전송 오버헤드가 연산 이득을 초과하여 성능이 소폭 하락(0.98배)하는 현상이 관측되었다.

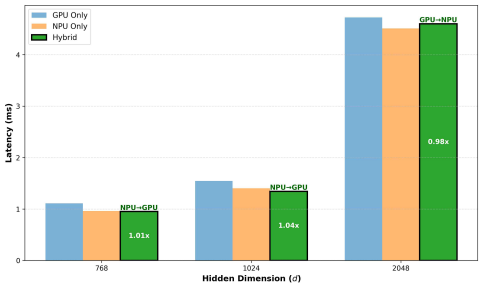

그림 2. FP16 정밀도에서의 Decode 단계 Hybrid 전략 효율성 분석

그림 3은 INT8 정밀도에서 Hybrid 실험 결과를 보여준다. d=4096 구간에서 GPU(QKV) -> NPU(Attn) 전략이 채택되어 GPU 단일 실행 대비 약 1.85배, NPU 단일 실행 대비 약 1.38배의 속도 향상 달성을 볼 수 있다. 반면 d=2048 구간에서는 Hybrid 조합 시 단일 GPU 대비 0.96배, 단일 NPU 대비 0.92배로 성능이 소폭 하락하는 현상이 관측되었다.

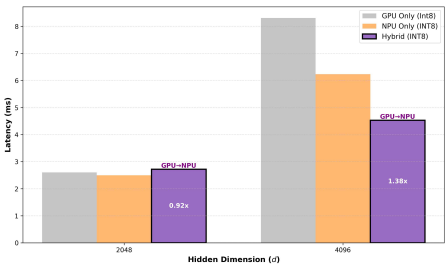

그림 3. INT8 정밀도에서의 Hybrid 전략 및 양자화 가속 효과.

Ⅴ. 결 론

본 연구는 모바일 이기종 SoC 환경(RK3588)에서 온디바이스 LLM의 추론 최적화를 위해 복잡한 텐서 분할 혹은 단일 백엔드 최적화 대신, Attention 연산의 하드웨어 병목을 정량적으로 모델링하여 각 단계별 최적 하드웨어를 순차적으로 점유함으로써, 구현의 용이성과 추론 효율성을 동시에 달성하는 직관적인 백엔드 선택 기법을 제안한다.

실험 결과, Prefill 단계의 짧은 시퀀스 처리 효율을 개선하고 Decode 단계에서 GPU 단독 실행 대비 최대 1.85배의 성능 가속 효과를 입증하였다. 이는 이기종 간 백엔드 전환 비용이 발생하더라도, NPU의 효율과 GPU의 유연성을 결합하는 전략이 전체 추론 성능에 유리함을 보여준다.

References

[1] L. Chen *et al*, "Characterizing Mobile SoC for Accelerating Heterogeneous LLM Inference," *Proceedings of the ACM SIGOPS 31st Symposium on Operating Systems Principles*, pp. 359, 2025. . DOI: 10.1145/3731569.3764808.

[2] MLC Contributors. *MLC-LLM*. Available: https://github.com/mlc-ai/mlc-llm.

[3] D. Xu *et al*, "Fast On-device LLM Inference with NPUs," *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, pp. 445, 2025. . DOI: 10.1145/3669940.3707239.

[4] T. Dao *et al*, "Flashattention: Fast and memory-efficient exact attention with io-awareness," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16344 - 16359, 2022. .