FLRot: 완전 학습 회전 기반 거대 언어 모델 양자화

강보현, 김찬훈, 박상기, 정기석*
한양대학교
{kbh2247, kch1103, skpark1101, kchung}@hanyang.ac.kr

FLRot: LLM Quantization with Fully Learned Rotation

Bohyeon Kang, Chan Hoon Kim, Sangki Park, Ki-Seok Chung*
Hanyang Univ, Seoul, Korea

요 약

거대 언어 모델 (LLM)의 엣지 디바이스 탑재를 위해서는 양자화 (Quantization)가 필수적이나, 이상치로 인한 성능 저하가 주요한 문제로 지적된다. 기존 연구인 SpinQuant는 가중치와 활성화 값의 회전을 통해 이를 완화하였으나, 추론 속도를 위해 일부 회전 행렬을 Hadamard 행렬로 제한하여 정확도 (perplexity) 향상에 한계를 보인다. 본 논문에서는 이러한 한계를 극복하기 위해 모든 회전 행렬을 학습 가능한 형태로 변환하면서 이에 따른 latency를 최소화하는 기법인 Fully Learned Rotation (FLRot)을 제안한다. 실험 결과, 제안하는 기법은 기존 기법 대비 0.43 정도의 PPL 개선 및 최대 4.3%의 zero-shot 추론 성능 향상을 보였으며, 커널 추론 지연 시간(latency)은 0.6x-1.27x 수준에서 유지됨을 확인하였다.

Ⅰ. 서 론

트랜스포머 기반 거대 언어 모델 (LLM)은 뛰어난 성능을 보이지만, 방대한 파라미터 수와 연산량으로 인해 모바일 및 엣지 디바이스에서의 구동에 제약이 따른다. 이를 해결하기 위해 4-bit 이하의 양자화 (Quantization) 기법이 활발히 연구되고 있으나, 활성화 값에 존재하는 이상치는 양자화 오차를 증폭시켜 모델 성능을 급격히 저하시킨다 [1].

기존 연구인 SpinQuant [2]는 회전 (Rotation) 행렬을 통해 이상치를 분산시키는 방식으로 이 문제를 효과적으로 완화하였다. SpinQuant는 일부 회전 행렬(R1, R2)은 학습을 통해 최적화하지만, 온라인으로 수행되는 R3와 R4는 고정된 아다마르 행렬 (Hadamard matrix)로 Fast Hadamard Transform (FHT) 연산을 수행한다. 그러나 이는 데이터 분포에 따른 정밀한 최적화를 저해하여 perplexity (PPL) 성능 향상의 한계를 야기한다.

따라서 본 논문에서는 모든 회전 행렬을 학습 가능한 파라미터로 변환하여 최적화하는 양자화 기법인 Fully Learned Rotation (FLRot)을 제안한다. 제안하는 방법의 효용성을 검증하기 위해 LLaMA-2 7B 모델을 대상으로 기존 양자화 방식과의 PPL을 측정하여 비교한다. 또한 RTX 3090 GPU 환경에서 FHT 커널과 커스텀 커널을 사용한 회전 행렬 연산 간의 지연 시간을 비교하여 모든 회전 행렬을 학습 가능한 파라미터로 확장함에 따른 실질적인 오버헤드가 미미한 수준임을 실험적으로 입증한다.

Ⅱ. 본론

2.1 회전 기반 양자화

회전 기반 양자화는 활성화 값에 회전 행렬을 곱하여 특정 차원에 집중된 이상치를 여러 차원으로 분산시키는 평탄화 기법이다. 그 중, 대표적 연구인 SpinQuant [2]는 회전 행렬을 그 특성에 따라 크게 병합 가능 회전과 온라인 회전으로 구분한다. 먼저 R1과 R2는 attention 블록 내부의 가중치에 흡수될 수 있는 병합 가능 회전 행렬이다. 이들은 직교성을 유지하는 Stiefel Manifold 상에서 Cayley SGD optimization [3]을 통해 학습되며, 추론 시에는 가중치 행렬에 미리 곱해지므로 추가적인 연산 오버헤드가 발생하지 않는다. R3와 R4는 4-bit 이하의 극단적인 양자화를 위해 도입된 온라인 회전 행렬로, 각각 KV Cache와 Feed-Forward Network (FFN) 블록 입력의 이상치를 제어하는 역할을 한다. 이들은 인접한 비선형 함수의 존재로 인해 가중치에 흡수될 수 없어 런타임에 직접 계산되며, SpinQuant는 연산 복잡도를 줄이기 위해 이를 학습 불가능한 아다마르 행렬로 고정하고 FHT 커널을 사용한다. 이는 $O(N^2)$의 연산 복잡도를 갖는 GEMM에 비해 FHT가 $O(N\log N)$의 연산만을 필요로 하기 때문이며, 특히 긴 시퀀스의 prefill이나 multi-batch 환경과 같은 연산 집약적 시나리오에서의 이론적인 처리량에 있어 우위를 보인다. 그러나 이는 온라인 회전 행렬을 고정된 아다마르 행렬로 제한함으로써 표현 자유도를 제한하는 결과를 초래한다.

2.2 완전 학습 회전

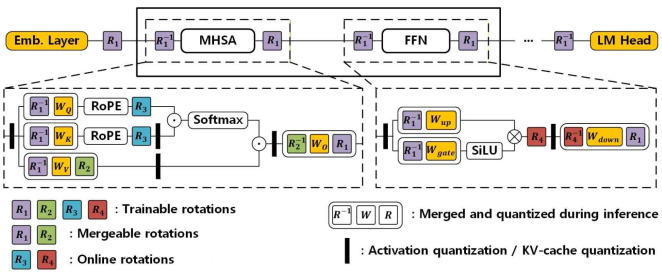

그림 1 FLRot 구조도

기존의 SpinQuant [2]는 FHT 연산을 위해 R3와 R4를 고정된 아다마르 행렬 또는 그 변형으로 제한하여, '학습을 통한 회전 최적화'라는 목적에도 불구하고 실제 학습 가능한 매개변수 공간은 병합 가능 행렬인 R1, R2에 국한되는 한계를 갖는다. 이러한 제약은 모델이 실제 활성화 값의 이상치 분포에 유연하게 대응하는 것을 방해하며, 결과적으로 4비트 이하의 저비트 양자화에서 좋은 성능을 보이지 못한다. 본 논문에서는 이러한 제약을 극복하기 위해 R3와 R4를 고정된 아다마르 행렬이 아닌, Stiefel

Manifold 상에서 정의된 학습 가능한 직교 행렬로 대체하는 완전 학습 회전 (Fully Learned Rotation, FLRot) 방식을 제안한다. 이를 통해 전체 회전 경로에 걸친 표현 자유도를 극대화함으로써, 4-bit 이하의 초저정밀도 양자화 과정에서 발생하는 이상치를 효과적으로 억제하였다.

이때 온라인 회전 행렬을 단순한 dense matrix 형태로 구현하는 경우 GEMM 연산은 $O(N^2)$의 연산 복잡도로 인해 연산 차원이 증가함에 따라 급격한 latency의 증가를 유발한다. Attention layer에 위치한 R3는 개별 헤드($d_{head} \times d_{head}$) 단위로 연산이 수행되기 때문에 이에 따른 latency 오버헤드가 미미한 반면, FFN에 위치하여 down-projection이 이루어지기 전에 수행되는 R4의 경우 intermediate dimension이 $m = 11{,}008$ (LLaMA-2 7B 기준)에 이르기 때문에 일반적인 GEMM 연산으로 구현할 경우 latency가 폭발적으로 증가하게 된다. 따라서 본 연구에서는 이들을 128×128 크기의 블록들로 구성된 블록 대각 직교 행렬로 정의하여 학습함으로써 추론 과정에서의 연산 효율성을 확보할 수 있도록 하였다. 이러한 구조적 제약은 최적화 시 표현 자유도를 일부 제한하지만, Triton 기반의 커스텀 커널 구현을 통해 zero-copy 방식으로 Tensor Core에서 연산이 가능하므로 기존 FHT 커널과 비교하여 추론 속도에 있어 실질적으로 미미한 오버헤드를 갖는다.

2.3 실험 환경 및 실험 결과

실험 환경: 제안하는 FLRot 기법의 성능을 검증하기 위해 LLaMA-2-7B 모델을 대상으로 실험을 수행하였다. 기존 기법과의 공정한 비교를 위해, SpinQuant [2] 공식 구현에서 사용된 것과 동일하게 WikiText-2 데이터셋에서 추출한 128개의 샘플을 회전 행렬 최적화에 사용하였다. 하드웨어 환경은 NVIDIA GeForce RTX 3090을 이용하였으며, 소프트웨어 환경은 PyTorch(Python 3.11) 기반으로 구축하였다. 블록 대각 행렬 곱셈을 수행하는 커스텀 커널은 Triton 3.1.0을 사용하여 구현 및 최적화하였다. 모델의 추론 성능을 평가하기 위해 PIQA, HellaSwag, ARC-Easy, ARC-Challenge, WinoGrande, LAMBADA (OpenAI) 총 6가지의 zero-shot 벤치마크 과제에 대한 성능을 측정하였으며, 최종적으로 기존 SpinQuant [2] 대비 PPL 변화량과 블록 대각 구조를 적용한 커널의 실행 속도를 정량적으로 비교 분석하였다.

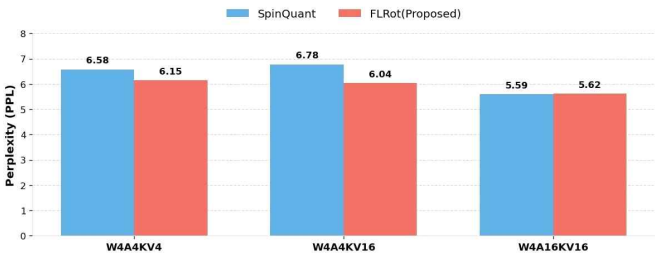

그림 2. 양자화 조건에 따른 PPL 비교

Table 1: Zero-shot performance comparison on LLaMA-2-7B

Configuration (W-A-KV)	Method	ARC-C (↑)	ARC-E (↑)	HellaS. (↑)	LAMB. (↑)	PIQA (↑)	WinoG. (↑)	Avg. (↑)
4-4-4	SpinQuant	0.3959	0.6679	0.7030	0.6652	0.7437	0.6314	0.6345
	FLRot (Ours)	**0.4241**	**0.7109**	**0.7287**	**0.7075**	**0.7666**	**0.6646**	**0.6671**
4-4-16	SpinQuant	0.3908	0.6263	0.7036	0.6652	0.7503	0.6101	0.6244
	FLRot (Ours)	**0.4206**	**0.6953**	**0.7311**	**0.7171**	**0.7671**	**0.6732**	**0.6674**
4-16-16	SpinQuant	0.4360	0.7243	0.7494	**0.7355**	0.7840	0.6701	0.6832
	FLRot (Ours)	**0.4420**	**0.7479**	**0.7511**	0.7273	0.7840	**0.6946**	**0.6912**

그림 3. Zero-shot 벤치마크 성능 비교

실험 결과: 실험 결과는 그림 2, 3, 4와 같다. PPL 분석 결과 FLRot은 W4A4KV4 설정에서 0.43의 PPL 감소를 기록하였으며, zero-shot 벤치마크 과제에서의 추론 성능은 W4A4KV4 및 W4A4KV16 설정에서 평균 정확도가 각각 3.26%p, 4.3%p 가량 향상되었다. 다만 활성화 값이 16-bit인 설정에서는 두 방식 간의 성능 차이가 미미한 수준으로 수렴하였다. 또한 실행 속도 비교를 위해 LLaMA-2-7B 모델의 FFN intermediate size (d=11,008) 조건에서 R4 연산의 실행 시간을 측정한 결과, PyTorch 기반의 단순 블록 대각 행렬 곱 구현(Torch BD)은 FHT 대비 높은 지연 시간을 보였으나 Triton 기반의 커스텀 커널(custom kernel BD)은 FHT와 대등하거나 우수한 수준의 추론 속도를 기록하였다. 측정 결과 $L < 128$ 구간에서는 커널 실행에 따른 오버헤드로 인해 FHT가 우세하였으나, $L \geq 512$ 구간에서는 Triton 기반 블록 대각 연산이 FHT 대비 약 1.1x-1.27x의 속도 향상을 기록하였다.

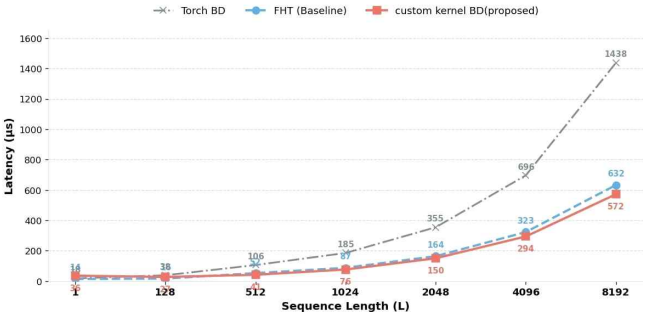

그림 4. 시퀀스 길이에 따른 커널별 실행 지연 시간(d=11,008)

Ⅲ. 결론

본 논문에서는 거대언어모델의 양자화 성능을 높이기 위해 기존 SpinQuant [2]의 구조적 제약을 극복한 FLRot 기법을 제안하였다. 온라인 회전에 사용되는 R3, R4 행렬을 고정된 아다마르 행렬로 제한하는 대신 학습 가능한 직교 행렬로 확장함으로써 활성화 값의 이상치를 효과적으로 제어하였다. LLaMA-2 7B 모델을 대상으로 실험한 결과 W4A4KV4 양자화 환경에서 PPL은 기존 대비 0.43가량 감소하였고, zero-shot 추론 성능에서 최대 4.3%의 성능 향상을 보이는 등 유의미한 개선을 보였다. 또한, 본 연구에서는 Triton 기반의 블록 대각 커널 최적화를 수행하여 실제 추론 시의 실행 속도는 시퀀스 길이에 따라 FHT 대비 0.6x-1.27x 수준을 유지하였고, 이를 통해 추론 과정에서 FLRot에 의한 추가적인 오버헤드는 제한적임을 확인하였다.

ACKNOWLEDGMENT

이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. RS-2024-00409492)

참 고 문 헌

[1] G. Xiao et al., "SmoothQuant: Accurate and efficient post-training quantization for large language models," ICML, 2023.

[2] Z. Liu et al., "SpinQuant: LLM quantization with learned rotations," ICLR, 2025.

[3] J. Li et al., "Efficient Riemannian optimization on the Stiefel manifold via the Cayley transform," ICLR, 2020.