트랜스포머 모델을 위한 Skewing 기반 QK 가중치 가지치기 기법

조현래[0], 신동군
성균관대학교 전자전기컴퓨터공학과
hrae10324@gmail.com, dongkun@skku.edu

Skewing-based QK weight pruning technique for transformer models

Hyun-rae Jo[0], Dongkun Shin
Department of Electrical and Computer Engineering, Sungkyunkwan University

요 약

대규모 언어 모델들은 그 방대한 크기로 인해 심각한 메모리 부족 문제를 겪고 있다. 이에 가중치 가지치기나 토큰 가지치기를 통해 추론 과정에서의 메모리 사용량을 줄이는 것이 중요해지고 있다. 본 논문에서는 행렬의 차원에 편향을 주어 정보 손실을 최소화할 수 있는 Skewing을 활용하여 학습이 필요하지 않은 쿼리와 키 가중치 차원 및 키 캐시 압축 기법인 SKPrune을 제안한다. 우리는 OPT-2.7B 모델을 이용한 실험을 통해 약 35%의 차원을 제거하더라도 정확도가 유지되는 것을 확인할 수 있었고, 상용 토큰 가지치기 기법 대비 동일한 정확도에서 2.4배 이상의 메모리 감소량을 확인할 수 있었다.

1. 서 론

대규모 언어 모델(Large Language Model, LLM)은 다양한 산업 분야에서 뛰어난 성과를 보여주고 있다. 그러나 모델 크기의 지속적인 증가로 인해 대규모 연산과 메모리 용량을 갖춘 환경에서만 실행할 수 있는 제약이 생겼다. 이러한 문제를 해결하기 위해, 가중치 가지치기 기법으로 불필요한 가중치를 제거하거나, LLM의 연산 과정에서 발생하는 Key-Value 캐시를 압축하여 메모리 사용량을 줄이는 다양한 방법을 모색되고 있다.

최근 발표된 Key-Value 캐시 압축 기법인 Infinigen[1]은 Query와 Key 행렬의 차원에 편향을 주어 중요한 차원을 강조할 수 있는 Skewing 기법을 제안했다. 또한, 강조된 일부 중요 차원만 연산하여 빠르게 중요한 토큰을 식별하는 수 있음을 보였다.

본 논문에서는 Skewing 기법을 활용해 Query와 Key 행렬에 이상치 차원을 생성하고, 불필요한 차원을 제거하여 재학습을 하지 않고도 정확도를 유지할 수 있는 가지치기 기법 SKPrune을 제안한다. 가중치 차원의 제거를 통해 Query와 Key 행렬의 차원도 줄어들기 때문에 키 캐시도 동일한 비율로 압축되는 효과가 있다.

* 이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.IITP-2017-0-00914, 지능형 IoT 장치용 소프트웨어 프레임워크)

2. 관련연구

2.1. 어텐션 연산과 직교 행렬의 특징

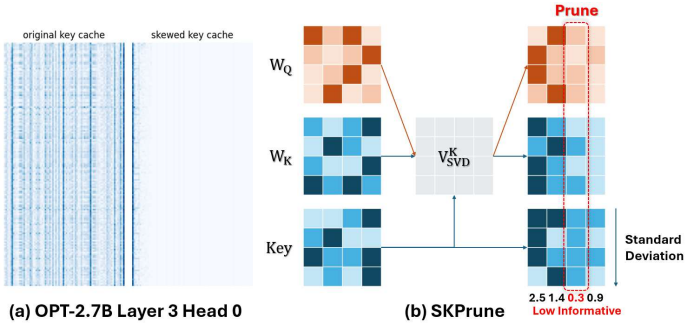
GPT[2]나 OPT[3]와 같은 트랜스포머 모델의 어텐션 연산은 다음과 같이 정의된다.

$$Q = x \cdot W_Q^T,\ K = x \cdot W_K^T,\ V = x \cdot W_V^T$$

$$A = Q \cdot K^T,\ S = \mathrm{Softmax}(A/\sqrt{d})$$

$$Out = S \cdot V \cdot W_O^T$$

어텐션 행렬 A는 Q와 K^T의 곱으로 정의되며, 소프트맥스 함수를 통해 최종 출력을 계산하게 된다.



[그림 1] (a) Skewing이 적용된 Key 행렬 (b) SKPrune

여기서 만약 Q와 K에 직교 행렬 D를 곱해준다면 다음과 같이 표현할 수 있으며,

$$\tilde{Q} = x \cdot W_Q^T \cdot D,\ \tilde{K} = x \cdot W_K^T \cdot D$$

이 경우 어텐션 행렬은 다음과 같이 계산된다.

$$A = \tilde{Q} \cdot \tilde{K}^T = x \cdot W_Q^T \cdot D \cdot D^T \cdot W_K \cdot x$$

$$= x \cdot W_Q^T \cdot W_K \cdot x = Q \cdot K^T$$

직교 행렬의 성질로 인해 $D^{-1} = D^T$이므로, D 를 곱해도 어텐션 연산에는 영향이 가지 않는다. 즉, 직교 행렬을 곱하는 것은 Q와 K의 형태를 변경할 수 있지만, 어텐션 연산에는 영향을 미치지 않음을 알 수 있다.

2.2. Skewing

Infinigen에서는 이 어텐션의 특성을 활용하여 Q와 K에 각각 직교 행렬 D 를 곱해 이상치 차원이 발생하도록 편향을 주었다. 여기서 직교 행렬 D 로는 Q 의 특이값 분해(Singular Value Decomposition, SVD)를 통해 얻은 V_{SVD}^Q를 사용한다. V_{SVD}^Q는 SVD의 정의에 따라 직교 행렬 조건에 부합하고 아래와 같은 특징을 가지게 된다:

$$Q = x \cdot W_Q^T = U_{SVD}^Q \cdot \Sigma_{SVD}^Q \cdot \left(V_{SVD}^Q\right)^T$$

$$\tilde{Q} = Q \cdot V_{SVD}^Q = U_{SVD}^Q \cdot \Sigma_{SVD}^Q$$

여기서 Σ_{SVD}^Q는 Q 행렬의 특이값을 나타내며, U_{SVD}^Q 의 기저벡터 방향으로 각 차원의 크기를 표현한다. V_{SVD}^Q를 곱해서 $\left(V_{SVD}^Q\right)^T$를 제거하면 특이값에 따른 이상치 차원이 발생한다.

이는 그림 1(a)에서 확인할 수 있다. 큰 값을 가지는 차원이 많았던 원본 Key 행렬과는 달리 Skewing을 적용하면 큰 값을 가지는 차원이 매우 적어지는 것을 볼 수 있다. 또한, 중요 차원을 제외한 나머지 차원의 값이 매우 작거나 거의 없어져서 불필요한 차원이 되는 것을 볼 수 있다.

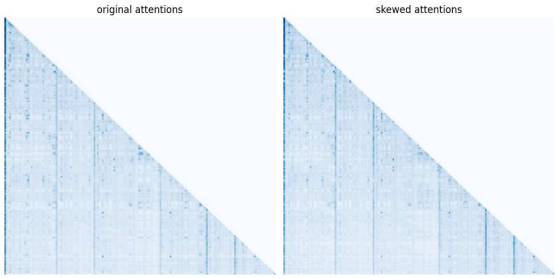
Infinigen은 이 특성을 활용해 Q와 K의 소량의 중요 차원만을 미리 계산하여 어텐션 연산에서 핵심적으로 사용될 토큰을 예측하고, 해당 토큰만을 사용하는 토큰 가지치기 기법을 제안했다. 가중치에 미리 직교 행렬을 곱해둔다면, Query와 Key에 Skewing이 적용된 것과 동일한 효과를 낼 수 있기 때문에 Prefill 과정에서 구한 직교 행렬을 가중치에 곱해두는 방식을 채택했다.

$$\tilde{Q} = x \cdot \left(W_Q^T \cdot V_{SVD}^Q\right),\ \tilde{K} = x \cdot \left(W_K^T \cdot V_{SVD}^Q\right)$$

3. 본 문

3.1. SKPrune

우리는 오프라인 단계에서 Calibration 문장을 사용하여 Skewing을 적용한 후, 그 영향으로 인해 불필요해진 차원들을 가지치기 할 것을 제안한다. 비중요



[그림 2] 원본 OPT-2.7B 모델의 어텐션 행렬과 SKPrun 적용 후의 어텐션 행렬

차원에 해당하는 가중치를 제거한다면 모델 자체의 크기를 줄일 수 있을 뿐만 아니라 Key 행렬의 크기도 작아지기 때문에 Key 캐시도 압축되는 효과를 누릴 수 있다.

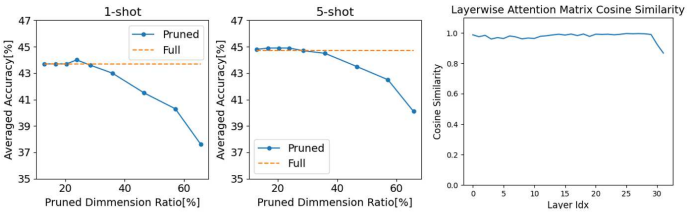
또한, 우리는 직교 행렬을 생성하기 위해 SVD가 적용되는 행렬을 기존의 쿼리 대신 키를 변경했다. 이는 쿼리에 비해 키에서 차원간의 특징이 더욱 뚜렷하게 나타나기 때문에 SVD를 적용했을 때에 편향이 더 잘 이루어지기 때문이다.

또한, 우리는 차원의 중요도를 평가하는 기준도 변경했다. Infinigen은 단순 합을 통한 평가 방법을 사용했다. Query가 (N, D)의 크기를 가질 때, N 방향의 합을 통해 각 차원의 중요도를 평가했다. 반면 우리는 차원의 표준편차 기준으로 중요도를 평가한다. N 방향의 표준편차가 높은 차원을 중요한 차원으로 결정한다. Query와 Key의 각 차원은 토큰이 나타내는 벡터의 한 축을 의미한다. 따라서 표준편차가 크다는 것은 해당 축에서 다양한 값들이 나타난다는 것을 의미하고, 이는 곧 그 차원에 포함된 정보가 풍부하다는 것을 의미한다. $Q \cdot K^T$ 에 이어지는 연산이 Softmax 이므로 차원의 절대적인 크기보다는 정보의 양으로 그 중요도를 판단하는 것이 바람직하다. 따라서 우리는 차원의 표준편차에 임계값을 두어 임계값 이하의 정보를 가지고 있는 차원은 제거한다. SKPrune의 전체적인 가지치기 기법은 그림 1(b)에서 확인할 수 있으며, Infinigen과 SKPrune의 차이점은 표 1을 통해 확인할 수 있다.

	Infinigen	SKPrune
Skewing	Prefill Phase	Offline
SVD	Query	Key
Metric	Magnitude	STD
Purpose	Token Pruning	Weight Pruning

[표 1] Infinigen과 SKPrune의 차이

Skewing의 영향으로 비중요 차원의 중요도가 더욱 떨어지고, 표준편차를 이용하여 정보의 양을 바탕으로 차원을 선택하기 때문에 가지치기 과정에서 최소한의 정보 손실만 발생한다. 따라서 가지치기 후의 어텐션 연산과 원본 모델의 어텐션 연산에는 큰 차이가 존재하지 않는다. 이는 그림 2에서



(a) 1-shot Task Accuracy (b) 5-shot Task Accuracy (c) Attention Matrix Similarity

[그림 3] (a), (b) OPT-2.7B 모델의 1-shot, 5-shot 상황에서의 평균 정확도 (c) SKPrune(35.9%)의 어텐션 유사도

확인할 수 있다. SKPrune으로 약 36%의 차원이 제거되었음에도 어텐션 연산의 결과가 매우 유사한 것을 확인할 수 있다.

4. 실 험

4.1. 실험환경

우리는 OPT-2.7B 모델을 사용하여 SKPrune의 성능을 측정했다. 1-shot과 5-shot 상황에서의 일반 추론 성능을 평가했으며 사용된 데이터셋은 OpenbookQA, PiQA, Arc-Challenge, Arc-Easy, MathQA이다.

4.2. 정확도

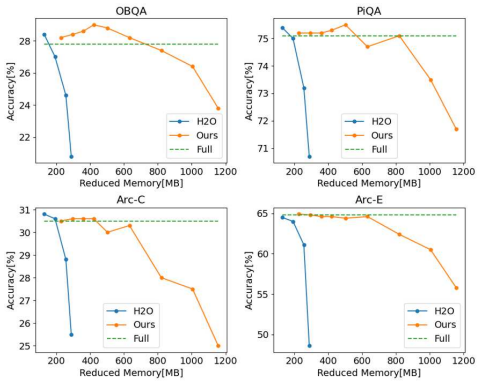
그림 3(a)는 1-shot 상황에서의 평균 정확도를, 그림 3(b)는 5-shot 상황에서의 평균 정확도를 가지치기 정도에 따라 나타낸다. 두 상황 모두에서 약 36% 까지 차원을 가지치기 하더라도 정확도가 유지되거나 심지어는 소폭 상승하는 것을 확인할 수 있다. 기존에는 모델이 문장을 이해하고 연산하기에 불필요한 정보가 많았으나, 가지치기를 통해 불필요한 정보가 제거되었기에 문장에 대한 이해도가 높아져서 정확도가 상승할 수 있었던 것으로 추측된다. 또한, 5-shot 상황에서는 가지치기의 비율이 높아지더라도 정확도가 상대적으로 잘 유지되는 경향이 있다. 이는 shot 수가 증가함에 따라 질문에 대한 힌트를 제공할 수 있는 단어가 많아지기 때문에, 개별 토큰의 정보가 부족하더라도 비교적 질문의 뜻을 정확히 이해하여 적절한 답변을 생성할 수 있기 때문이다.

4.3. 어텐션 유사도

그림 3(c)는 원본 모델과 35.9%의 SKPrune을 적용한 모델의 레이어별 어텐션 유사도를 나타낸 것이다. 대부분의 레이어에서 1에 가까운 유사도를 가지고 있다. 마지막 레이어가 비교적 낮은 유사도를 갖는 것은 Key의 채널간 특징이 비교적 약하기 때문이다.

4.3. 메모리 감소량

그림 4는 대표적인 토큰 가지치기 기법인 H2O[4]와 SKPrune의 메모리 감소량 대비 정확도를 비교한 것이다. 이는 일반 추론 5-shot 상황의 정확도이며, 평균 토큰



[그림 4] OPT-2.7B 모델의 SKPrune과 H2O에서의 메모리 감소량 및 정확도

수를 512로 설정하여 메모리 감소량을 계산했다. 이를 통해 SKPrune이 H2O 대비 동일한 정확도에서 훨씬 적은 메모리를 사용하는 것을 확인할 수 있었다.

5. 결 론

본 논문에서는 Skewing을 활용하여 학습 없이도 정확도를 유지할 수 있는 Query와 Key 가중치 차원 가지치기 기법 SKPrune을 제안한다. Skewing을 통해 가중치를 차원 가지치기에 더욱 적합하도록 편향할 수 있기 때문에 가지치기 이후에도 원본과 유사한 어텐션 연산이 가능하다.

또한, 기존의 Query의 크기 기반 중요도에서 Key의 표준편차 기반 중요도 평가로 변경함으로써 Skewing이 보다 효과적인 편향을 수행할 수 있도록 했고, 정보가 풍부한 차원을 선택하여 정보 손실을 최소화할 수 있었다.

실험 결과, 약 36% 가량의 차원을 가지치기 하더라도 모델의 정확도가 유지됨을 확인했으며, 상용 토큰 가지치기 기법과 비교하여 동일한 수준의 정확도에서 훨씬 큰 메모리 감소량을 달성할 수 있음을 보였다.

참 고 문 헌

[1] Lee, Wonbeom, et al. "{InfiniGen}: Efficient generative inference of large language models with dynamic {KV} cache management." 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24). 2024.

[2] Achiam, Josh, et al. "Gpt-4 technical report." arXiv preprint arXiv:2303.08774 (2023).

[3] Zhang, Susan, et al. "Opt: Open pre-trained transformer language models." arXiv preprint arXiv:2205.01068 (2022).

[4] Zhang, Zhenyu, et al. "H2o: Heavy-hitter oracle for efficient generative inference of large language models." Advances in Neural Information Processing Systems 36 (2024).