Vision Transformer 의 효율 향상을 위한 Sparse Attention 과 Token Pruning 의 동시 적용 연구

허지현, 노수민, 정기석*
한양대학교

jhheo@hanyang.ac.kr, smrho@hanyang.ac.kr, *kchung@hanyang.ac.kr

A Study on the Combined Application of Sparse Attention and Token Pruning for Enhancing the Efficiency of Vision Transformers

Ji Hyeon Heo, Soo Min Rho, Ki-Seok Chung*
Hanyang University, Seoul, Korea

요 약

Vision Transformer 의 효율적 연산을 목표로 한 기존 연구에서는 Attention 연산의 높은 연산 복잡도 문제를 완화하기 위해 Sparse Attention 기법을 널리 활용해 왔다. 하지만, Sparse Attention 은 이미지 처리 모델에서 높은 비중을 차지하는 Linear transform 연산량을 줄이는 데 한계를 보인다. 따라서, 본 논문에서는 Linear transform 의 연산량을 줄이는데 효과적인 token pruning 을 Sparse Attention 과 동시에 적용하는 방법을 제안한다. ImageNet 데이터셋 기반 DeiT 모델의 실험 결과, 제안된 방법은 기존 Sparse Attention 대비 정확도는 평균 0.2p%~0.9p% 감소를 보이며, end-to-end 성능은 평균 1.31 배 향상되는 결과를 보였다.

Ⅰ. 서 론

트랜스포머 (Transformer) 모델은 자연어 처리 영역뿐 아니라 이미지 분류와 같은 컴퓨터 비전 분야에서도 뛰어난 성능을 보이며 광범위하게 활용되고 있다. 특히, Vision Transformer (ViT) [1] 모델은 Attention 메커니즘을 통해 기존의 CNN 기반 모델 대비 향상된 정확도를 보인다. 그러나 Attention 연산은 입력 이미지 패치 개수의 제곱에 비례하는 연산 복잡도를 가지므로, 높은 연산 및 메모리 사용량이라는 문제점을 내포한다. 이러한 제약을 극복하기 위해 Sparse Attention 기법에 대한 연구가 활발히 진행되고 있으며, 이는 Attention 의 연산 복잡도를 크게 낮춘다. 그러나 트랜스포머 모델은 Attention 뿐만 아니라 다량의 Linear transform 연산을 가지기 때문에, Sparse Attention 을 적용하더라도 Linear transform 에서 높은 수준의 연산량이 유지된다는 한계점을 가진다. 더불어, 이미지 처리의 경우 자연어 처리와 달리, 입력 토큰의 개수가 고정되어 있으며, 이로 인해 Linear transform 연산이 차지하는 비중이 상대적으로 큰 특징이 있다. 따라서, 본 논문에서는 Attention 연산량 감소에 효과적인 Sparse Attention 과 더불어, Linear 연산의 부담을 완화하기 위한 token pruning 의 동시 적용하는 방법을 제안한다. 제안하는 방법의 효용성을 검증하기 위해 ImageNet 데이터셋에서 DeiT 모델을 대상으로 50~80% sparsity 레벨의 Sparse Attention 과 token pruning 을 함께 적용하는 방법의 정확도를 측정하고, 사이클 단위의 시뮬레이터를 통해 Sparse Attention 만 적용한 방법과의 성능을 비교하였다.

Ⅱ. 본론

2.1. Sparse Attention

그림 1 은 기존의 Sparse Attention 연구[2]에서 분석한 EdgeGPU 에서 DeiT [3]모델 추론의 연산량 (FLOPs)과 레이어별 처리 시간 (end-to-end latency)을 나타낸다. 이를 통해, Attention 연산이 연산량에 비해 긴 처리 시간을 유발하는 것을 확인할 수 있다. 하지만 입력 토큰의 개수가 고정적이라는 이미지 처리의 특성상 Attention 연산의 비중이 Linear transform 에 비해 극단적으로 높지는 않으며, 모델에 따라서 연산의 비중이 Attention 보다 Linear transform 에서 크게 나타나는 경우도 있다.

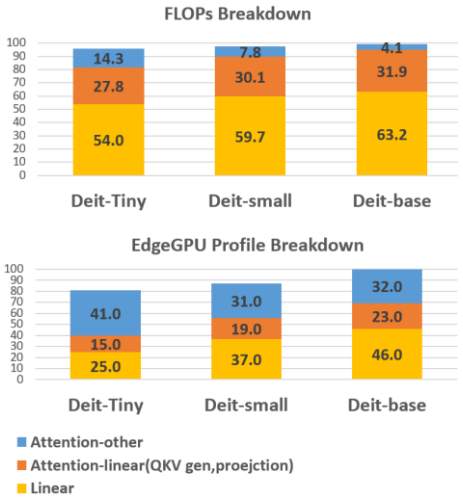

그림 1. EdgeGPU TX2 에서 DeiT 모델에 대한 FLOPs(위)와 latency(아래)의 breakdowns [2]

본 논문은 ViTCoD[2]에서 제안한 Static Sparse Attention 기법을 적용하여 Attention 의 연산량을 줄인다. [2]는 모델의 훈련 과정에서 attention score 를 이용해 attention map 의 마스크를 생성하고 이를 추론에서 사용하는 기법을 제안하였다. 훈련에서의 마스크 생성 과정은 다음과 같다. 먼저, 모든 데이터셋에 대한 attention map 의 평균값을 구하고, 각 쿼리 (attention map 의 각 행)에 대해 attention score 를

내림차순으로 정렬한 뒤, 총합이 임계값에 도달할 때까지의 attention score 를 선택하여 더한다. 선택된 attention score 에 해당하는 위치는 이진 마스크에서 1 로, 나머지는 0 으로 설정하여 마스크를 생성한다. 모델 추론 과정에서는 생성한 마스크에 따라 Attention 연산을 선택적으로 수행하는 것으로 Attention 의 연산량을 줄일 수 있다.

2.2. Token pruning

Sparse Attention 은 Attention 의 연산량을 줄이는 데 효과적이지만, 전체 연산에서 큰 비중을 차지하는 Linear transform 의 높은 연산량은 여전히 유지되는 한계가 있다. 이를 해결하기위해 본 논문에서는 Sparse Attention 에 token pruning 을 추가로 적용하여 전체적으로 연산량을 감소시키고자 한다. Linear transform 은 토큰 수에 비례하는 연산 복잡도를 가지므로 token pruning 을 통한 토큰 수 감소를 통해 큰 효율성 향상을 기대할 수 있다.

본 논문에서 제안한 token pruning 은 attention score 를 기반으로 토큰의 중요도 점수를 생성하고, 이를 평가하여 중요도가 낮은 토큰을 제거한다. 이 과정은 레이어마다 연쇄적으로 적용되며, 한 번 제거된 토큰은 이후 레이어에서 완전히 배제된다. 중요도 점수는 attention 값을 열 방향으로 쿼리 개수만큼 누적하여 산출한다. 모든 헤드에 대해 누적된 점수를 기준으로 사전 정의된 pruning ratio 만큼의 토큰이 제거된다. pruning ratio 는 시작 비율과 종료 비율을 설정하여 레이어 별로 조정되며, 지수 보간법을 통해 계산된다.

Token pruning 선행 연구 [4]에서 언급하는 바와 같이, 초기 레이어를 과도하게 pruning 할 경우, 입력 이미지의 기본 구조 정보가 손실되어 성능 저하를 초래할 수 있다. 따라서 본 논문에서는 초기 3 개의 레이어를 pruning 대상에서 제외하였다. 또한, pruning ratio 를 0.1~0.9 의 값으로 설정하고, 후반 레이어로 갈수록 더 높은 비율로 pruning 을 수행하여 이미지의 기본 구조 정보를 효과적으로 보존하고자 하였다.

2.3. 실험방법 및 결과

제안하는 방법의 성능 평가를 위해 ImageNet 데이터셋에서의 DeiT-Base/Small/Tiny 모델의 정확도를 측정하였다. 표 1 은 50~80% sparsity 레벨에서 기존의 Sparse Attention 만 적용한 것과 token pruning 을 함께 적용한 것을 비교하였다. 모델에 따라 평균적으로 0.2p%(Base), 0.5p%(Small), 0.9p%(Tiny)의 정확도 감소를 보였다.

표 1. Token pruning 유무에 따른 정확도 비교

Sparsity	Model	Sparse attention accuracy	Token pruning & Sparse attention accuracy
50%	DeiT-Tiny	72.1	71.0
60%		71.8	70.9
70%		71.5	70.6
80%		70.9	70.2
50%	DeiT-Small	79.9	79.5
60%		79.8	79.3
70%		79.5	79
80%		79.2	78.6
50%	DeiT-Base	81.8	81.8
60%		81.7	81.6
70%		81.5	81.3
80%		81.3	80.9

또한, 사이클 단위의 시뮬레이터를 구현하여 end-to-end 성능을 측정하였다. 그림 2 는 Sparse Attention 가속기 [3]와 본 논문에서 제안한 token pruning 이 추가 적용된 가속기의 성능을 비교한 결과를 보여준다. 각 모델을 대상으로 50~80% sparsity 레벨에 대해 Sparse Attention 만 적용한 가속기와 비교하였을 때, 평균적으로 1.31 배의 성능 향상을 보였다.

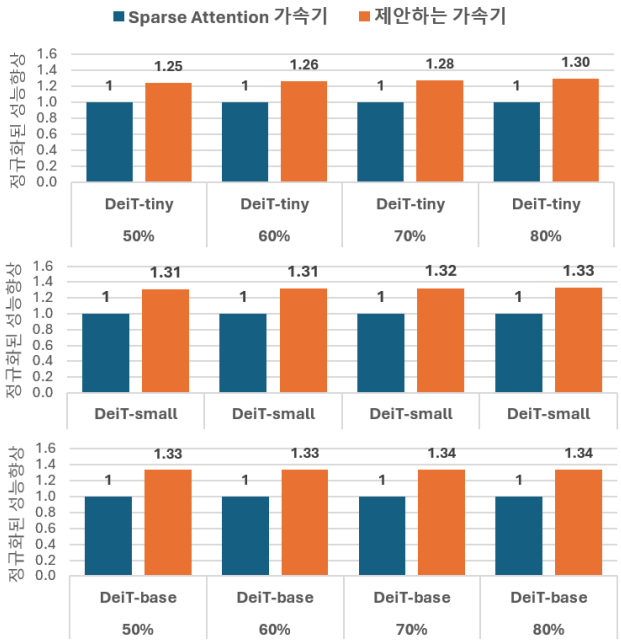

그림 2. End-to-end 모델의 가속 성능 평가 결과

Ⅲ. 결론

본 논문에서는 Sparse Attention 적용 시 발생하는 한계점을 해결하기위해, token pruning 과 Sparse Attention 의 동시 적용한 방법을 제안하였다. 실험 결과 Sparse Attention 만 적용했을 때 대비 평균 0.2~0.9p%의 정확도 하락에서, 평균적으로 1.31 배의 end-to-end 성능 향상을 관찰할 수 있었다.

ACKNOWLEDGMENT

본 논문은 2020 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No. 2020-0-01304, 모바일 자가 학습 가능 재귀 뉴럴 네트워크 프로세서 기술 개발).

참 고 문 헌

[1] Dosovitskiy, Alexey. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).

[2] You, Haoran, et al. "Vitcod: Vision transformer acceleration via dedicated algorithm and accelerator co-design." 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA). IEEE, 2023.

[3] Touvron, Hugo, et al. "Training data-efficient image transformers & distillation through Attention." International conference on machine learning. PMLR, 2021.

[4] Wang, Hanrui, Zhekai Zhang, and Song Han. "Spatten: Efficient sparse attention architecture with cascade token and head pruning." 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA). IEEE, 2021.