범용 Transformer 모델의 처리 속도 벤치마크: BERT, Longformer, BigBird 비교 연구

조진용, 김동균, 조부승
한국과학기술정보연구원
jiny92@kisti.re.kr

Performance Benchmarking of Transformer Models: BERT, Longformer, and BigBird

Jinyong Jo, Dongkyun Kim, Buseung Cho
Korea Institute of Science and Technology Information

요 약

Transformer의 self-attention 메커니즘은 $O(n^2)$ 복잡도로 인해 긴 시퀀스 처리에 제약이 있다. 대표적인 dense attention 모델인 BERT는 계산 복잡도로 인해 최대 512 토큰만 처리 가능하여 긴 문서 분석이나 시계열 데이터의 장기 패턴 탐지가 불가능하다. 긴 시퀀스 처리를 위해 sparse attention 모델이 제안되었으나, 성능 분석은 이론적 복잡도나 downstream task의 정확도 평가에 집중되어 있고 실제 학습 및 추론 속도에 대한 체계적 벤치마크는 부족한 실정이다. 본 연구는 BERT, Longformer, BigBird 등 3개 범용 Transformer 모델의 처리 성능을 다양한 토큰 길이와 배치 크기에서 측정하고 각 모델의 성능 특성을 분석하였다. 실험 결과, 512 토큰 이하의 시퀀스에서는 BERT와 BigBird가 높은 처리 성능을 보였으며, 512 토큰을 초과하는 긴 시퀀스의 경우 추론은 BigBird가, 학습에서는 Longformer가 효율적임을 확인하였다.

1. 서론

Transformer 아키텍처[1]의 등장은 자연어 처리 분야에 혁신을 가져왔으나, self-attention 메커니즘의 $O(n^2)$ 복잡도는 긴 시퀀스 처리에 제약으로 작용한다. 입력 시퀀스의 모든 토큰 쌍에 대한 attention score 계산이 복잡도의 원인이며, 시퀀스 길이가 증가할수록 메모리 사용량과 처리 시간이 급격히 증가하는 문제가 발생한다. 이로 인해, BERT[2] 등 초기 Transformer 모델들은 대부분 512 토큰으로 최대 시퀀스 길이가 제한되어 있다.

계산 복잡도 문제는 대량의 데이터를 실시간으로 분석해야 하는 보안 분야에서 특히 장벽으로 작용한다. Transformer 기반의 이상 탐지 연구가 활발히 진행되고 있으나[3], 512 토큰 제한은 시간 범위에 걸친 공격 패턴의 분석을 어렵게 한다. 효과적인 보안 위협 탐지를 위해서는 개별 이벤트의 의미와 함께 시간적 상호 연관성을 분석해야 하기 때문이다.

긴 시퀀스 처리의 계산 복잡도 문제를 해결하기 위해 Longformer[4], BigBird[5] 등 sparse attention 기반 모델들이 제안되었다. Sparse attention 모델은 $O(n^2)$인 dense attention 모델의 복잡도를 $O(n)$ 또는 $O(n\log n)$로 개선하였으며 512 토큰 이상의 긴 시퀀스도 처리할 수 있다. Sparse attention 모델의 성능 분석은 주로 이론적 복잡도나 downstream task의 정확도 평가에 집중되어 있으며, 학습 및 추론 속도에 대한 벤치마크는 부족한 실정이다.

본 연구는 범용 Transformer 기반 모델인 BERT, Longformer, BigBird의 처리 속도를 토큰 길이와 배치 크기 변화에 따라 측정하고 결과를 비교 분석한다. 이를 통해 sparse attention 모델의 실제 처리 속도를 dense attention 모델과 비교하고 각 모델의 성능 특성을 파악한다. 벤치마크 결과는 긴 시퀀스 처리를 필요로 하는 Transformer 응용 분야에서 모델 선택의 기준을 제시하고, sparse attention 메커니즘의 효율성에 대한 실험적 근거를 제공한다.

2. 배경 이론

Transformer 기반 모델들은 attention 메커니즘에 따라 dense attention과 sparse attention으로 구분된다. Dense attention은 Transformer 초기 모델의 attention 방식으로, 입력 시퀀스의 모든 토큰이 완

<표 1> Comparison of transformer base models (Reformer: average complexity)

Model	Attention	Complexity	Position encoding	Params.	Hidden size	Layers	Heads	Max tokens
BERT[2]	Dense self	$O(n^2)$	Learned	110M	768	12	12	512
Lonformer[4]	Window, global	$O(n)$	Learned	149M	768	12	12	4,096
BigBird[5]	Random, window, global	$O(n)$	Learned	131M	768	12	12	4,096
Reformer[6]	LSH	$O(n \log n)$	Axial	38M~ 150M+	512~1024	2~12	8~16	65,536+
Linformer[7]	Low-rank projection	$O(n)$	Learned	~110M	768	12	12	∞
Performer[8]	FAVOR+	$O(n)$	Sinusoidal	~110M	768	12	12	∞

전 연결 방식으로 상호 비교된다. BERT가 dense attention의 대표적 모델이며, 시퀀스 길이 n에 대해 $O(n^2)$의 복잡도를 갖는다. Attention 계산 시 Query (Q)와 Key (K) 행렬의 곱인 유사도 QK^T의 크기가 n×n이 되기 때문으로, 긴 시퀀스 처리 시 메모리와 계산 비용이 급격히 증가하는 문제가 있다.

Sparse attention 메커니즘은 attention 연결을 선택적으로 제한하거나 행렬 차원을 축소하여 계산 복잡도를 줄이는 방식이다. 표 1은 dense attention과 sparse attention을 대표하는 Transformer 모델들의 특성을 보여준다. Longformer는 sliding window attention과 global attention을 결합한 모델로, 대부분의 토큰이 윈도우 w 범위의 인접 토큰들과 연결되고, 소수의 global 토큰만 전역 연결되어 O(n)의 선형 복잡도를 갖는다.

BigBird는 sliding window, global tokens, random attention의 세 가지 패턴을 조합한 모델로, 특히 708 또는 1024 토큰 이하에서는 full attention을, 그 이상에서는 block sparse attention을 적용하는 특징이 있다. 블록 단위로 인접 블록들 및 무작위 블록과 연결하며, 선택된 블록 내부는 완전 연결된다. BigBird 메커니즘은 짧은 시퀀스에서 성능 저하를 방지하고 긴 시퀀스에서 O(n)의 계산 효율성을 제공한다.

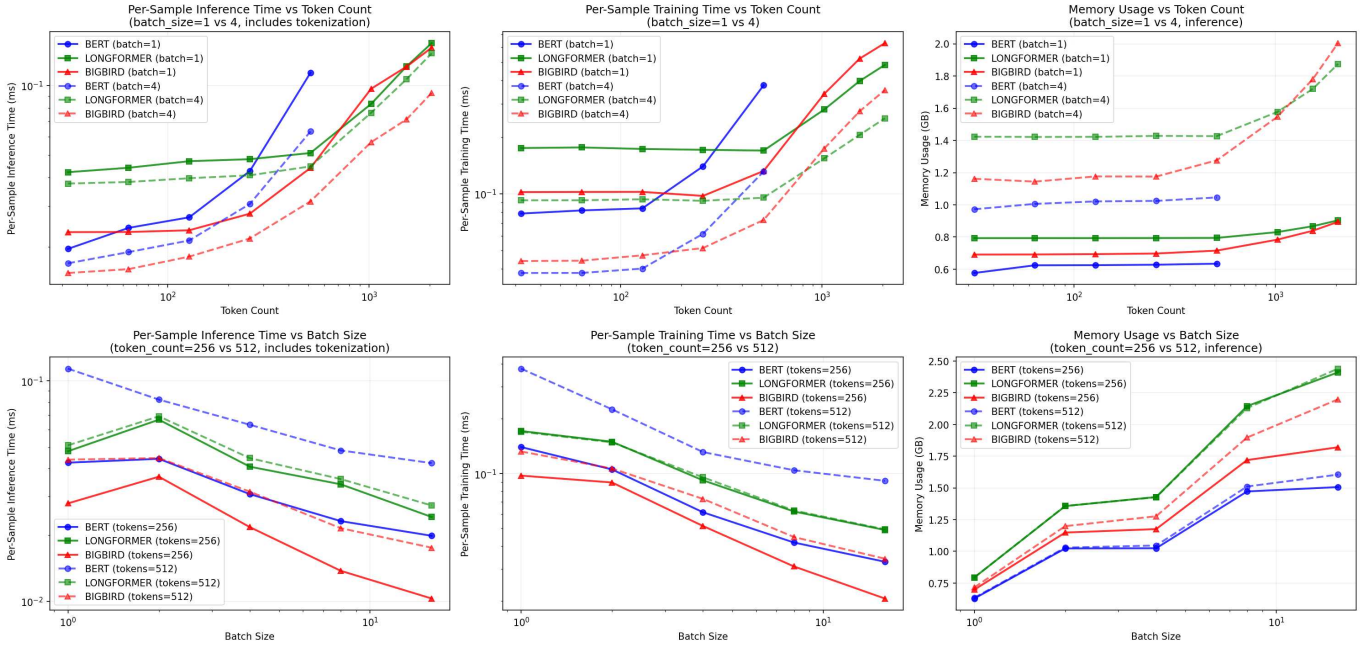
BigBird와 Longformer 외에도 다양한 sparse attention 모델들이 존재한다. Reformer는 Locality-Sensitive Hashing (LSH)을 활용하여 벡터 공간에서 유사한 토큰들을 같은 버킷으로 그룹화하고, 버킷 내 토큰들끼리만 attention을 계산하여 복잡도를 낮춘다. Linformer는 attention 행렬이 본질적으로 저차원 구조를 갖는다는 가정 하에 K와 V (Value)를 저차원으로 투영하여 O(n) 복잡도를 구현한다. Performer는 FAVOR+ 알고리즘을 사용하여 지수 연산이 필요한 유사도 계산 (Kernel) 함수를 선형 연산으로 근사하고, 행렬 곱셈 순서를 변경하여 O(n) 복잡도를 확보한다.

위치 인코딩 방식도 모델별로 상이하다. 위치 인코딩은 attention이 입력 순서에 무관하게 작동하는 문제를 보완하기 위해 각 토큰에 위치 정보를 추가하는 메커니즘이다. BERT, Longformer, BigBird는 각 토큰 위치에 학습 가능한 가중치 벡터를 할당하는 learned positional encoding을 사용한다. Reformer는 메모리 효율성을 높이기 위해 토큰의 위치를 2차원으로 분해하는 axial positional encoding을 채택하며, Performer는 삼각함수 기반의 sinusoidal encoding으로 추가 가중치 없이 위치를 표현한다.

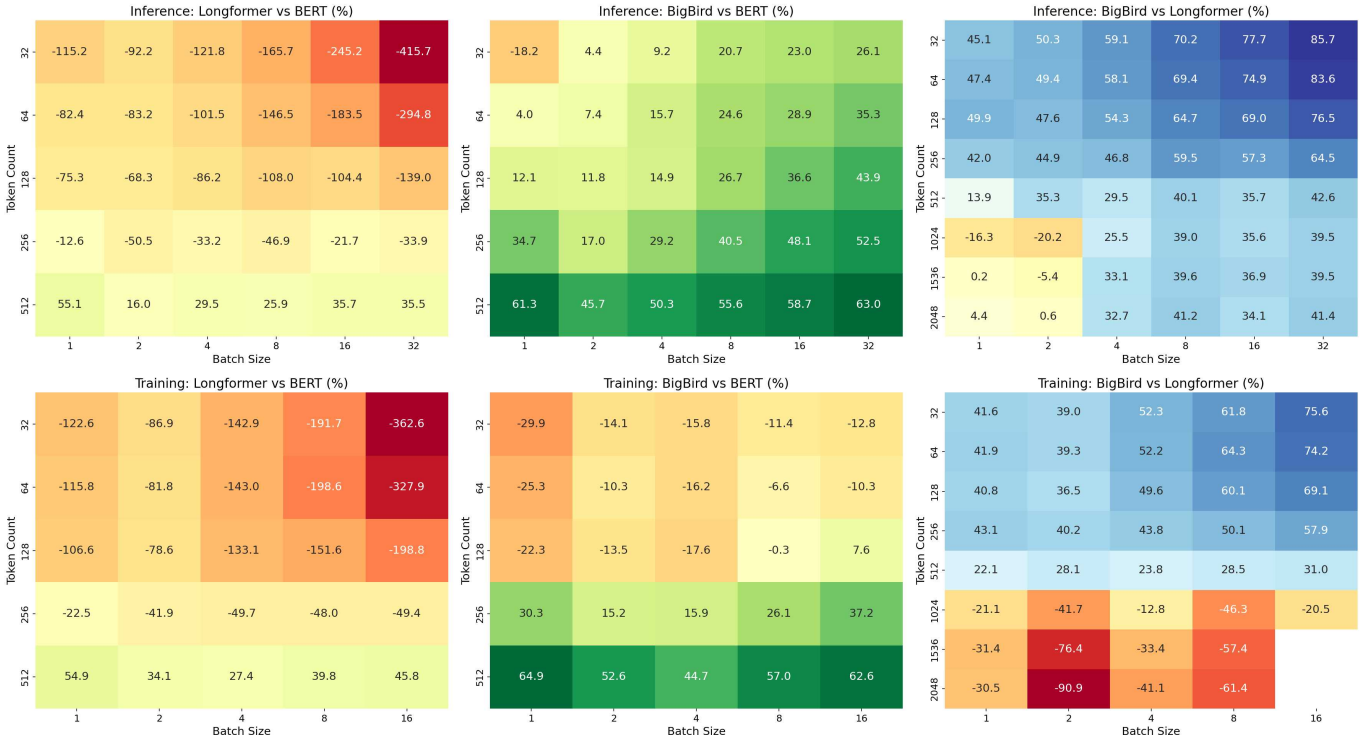
3. 실험 및 분석

실험은 Intel Xeon E5-2637 v4 @ 3.50GHz 듀얼 프로세서 (16코어)와 132GB RAM을 탑재한 서버 시스템에서 수행되었다. GPU 가속을 위해 NVIDIA GeForce GTX 1080 Ti (11GB) 3개를 사용하였으며, PyTorch 2.5.1 (CUDA 12.2)를 이용해 실험 코드가 구현되었다. 본 연구에서는 표준 PyTorch tensor를 출력하여 직접적인 성능 비교가 가능한 BERT, Longformer, BigBird로 성능 비교를 제한한다.

모든 모델은 HuggingFace Transformers 라이브러리 (v4.36.0)에서 제공하는 사전 학습된 가중치를 사용하였다. 라이브러리 및 구현 시기에 따른 최적화 차이가 벤치마킹 결과에 영향을 미칠 수 있다. 실험은 토큰 길이를 32부터 4096까지, 배치 크기를 1부터 16까지 변화시키며 각 조건에서 10회 반복 측정하여 평균값을 산출하였다. 측정된 시간은 토큰화, GPU 메모리 전송, 모델 추론, 동기화 등 전체 처리 과정의 end-to-end 시간이므로, 멀티 GPU 병렬 처



<그림 1> Training and inference speed comparison of transformer models.



<그림 2> Performance gains in inference and training time (%).

리에 따른 통신 오버헤드를 포함한다.

그림 1은 토큰 길이와 배치 크기 변화에 따른 세 모델의 학습 및 추론 시간, 메모리 사용량을 나타낸다. 학습 시간 측면에서 BERT는 256 토큰 이하의 짧은 시퀀스와 작은 배치 크기에서 Longformer와 BigBird와 비교해 우수한 성능을 보였다. 추론 시간에서는 BigBird가 64 토큰 이상 구간에서 배치 크기와 무관하게 가장 빠른 처리 속도를 기록하였다. Longformer는 입력을 512의 배수로 패딩하는 구현 특성으로 인해 512 토큰 이하에서 큰 오버헤드가 발생하였다. 예를 들어, 32 토큰 입력이 512 토큰으로 패딩되어 16배의 불필요한 연산이 추가된다. 패딩 오버헤드는 짧은 시퀀스에서 Longformer의 성능을 BERT 대비 최대 2.15배 저하시키는 주요 요인으로

작용하였다.

실험에서 BigBird는 요구되는 최소 수인 708 토큰을 임계값으로 하여 dense attention과 sparse attention을 전환하도록 구성하였다. 즉, 708 토큰 이하에서는 BERT와 유사한 dense attention 방식으로 동작하고 그 이상에서는 block sparse attention으로 작동한다. 256 토큰부터 BigBird가 dense attention을 사용함에도 BERT보다 빠른 성능을 보였다. 이는 BigBrid의 최적화된 구현이나 효율적인 메모리 접근 등에 기인한 것으로 추정된다.

메모리 사용량 측면에서 BERT는 512 토큰 이하에서 가장 효율적인 모델로 확인되었다. 배치 크기와 무관하게 BERT가 Longformer와 BigBird 대비 일관되게 낮은 메모리 사용량을 기록하였다. BigBird는 512 토큰을 초과하면서 메모리 사용량이 급격히 증가하여, 배치 크기 8과 토큰 길이 2,048 조건에서 학습 시 메모리 부족(Out-Of-Memory, OOM)이 발생하였다. Longformer 역시 배치 크기 16과 토큰 길이 2,048 조건에서 OOM이 관찰되어 실험 환경과 같은 저사양 GPU에서 해당 모델의 활용에 제약이 있음을 확인하였다.

그림 2는 BERT, Longformer, BigBird 간 상대적 성능 비교를 나타낸다. Longformer는 512 토큰 미만에서는 모든 배치 크기에서 BERT보다 느렸으나, 512 토큰에서는 BERT를 능가하는 처리 속도를 보였다. 이는 Longformer의 패딩 오버헤드 비중이 시퀀스 길이 증가에 따라 감소하기 때문이다. BERT의 토큰 제한으로 인해, 512 토큰을 초과하는 구간에서는 Longformer와 BigBird 간 비교만 가능하다. 추론 처리 속도에서 BigBird가 일관되게 우수했으나, 1024 토큰 이상의 학습에서 Longformer가 BigBird보다 빠른 처리 속도를 보였다. 이는 긴 시퀀스 학습에서 Longformer의 sliding window attention이 BigBird의 block sparse attention보다 효율적임을 의미한다.

4. 결론

본 연구는 범용 Transformer 모델의 특성을 살펴보고 처리 속도를 비교 분석하였다. Longformer는 패딩 오버헤드로 인해 512 토큰 미만에서 BERT 대비 평균 2.15배 느린 성능을 보였다. BigBird는 추론에서 BERT보다 빠른 속도를 보였으며, 학습에서는 256 토큰부터 BERT를 능가했다. 512 토큰을 초과하는 긴 시퀀스에서 추론은 BigBird가, 학습은 Longformer가 각각 우수한 성능을 보였다. 본 연구는 처리 속도와 메모리 효율성을 평가하였으며, 모델의 정확도는 측정하지 않았다. 향후, sparse attention 모델의 정확도와 처리 속도 간 trade-off를 연구할 예정이다.

※ 본 연구는 한국과학기술정보연구원의 지원(K25L5M1C1-01)으로 수행되었습니다.

참고문헌

[1] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, vol. 30, 2017.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. 2019 Conf. North American Chapter Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, Jun. 2019, pp. 4171-4186.

[3] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, "Deep learning for anomaly detection in log data: A survey," Machine Learning with Applications, vol. 12, pp. 100470, 2023.

[4] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The Long-Document Transformer," arXiv preprint arXiv:2004.05150, 2020.

[5] M. Zaheer et al., "Big Bird: Transformers for Longer Sequences," in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 17283-17297.

[6] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The Efficient Transformer," in Proc. 8th Int. Conf. Learning Representations (ICLR), Addis Ababa, Ethiopia, Apr. 2020.

[7] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-Attention with Linear Complexity," arXiv preprint arXiv:2006.04768, 2020.

[8] K. Choromanski et al., "Rethinking Attention with Performers," in Proc. 9th Int. Conf. Learning Representations (ICLR), virtual event, May 2021.