대규모 NLP 모델에서 병렬화 기법의 성능 분석

이재영 [1], 남재현 [2]
[1] 단국대학교 컴퓨터공학과 학부생
[2] 단국대학교 컴퓨터공학과 교수

32193424@dankook.ac.kr, namjh@dankook.ac.kr

The Performance Analysis of Parallelization Techniques in Large-Scale NLP Models

-

Jaeyoung Lee[1], Jaehyun Nam[2]
[1]Dept. of Computer Engineering, Dankook University (Undergraduate Student)
[2]Dept. of Computer Engineering, Dankook University (Professor)

요 약

최근 자연어 처리(Natural Language Processing , NLP) 모델의 규모가 급격히 증가함에 따라, 하드웨어가 처리해야 할 파라미터 수가 기하급수적으로 늘어나고 있다. 이로 인해 대규모 모델의 효율적인 학습을 위해 다양한 병렬화 및 분산 학습 기법이 도입되었다. 이러한 기법은 데이터셋의 크기와 모델의 구조에 따라 학습 성능에 상당한 차이를 보인다. 본 연구에서는 IMDB (Internet Movie Database) 감정 분류 작업을 대상으로, Bert 와 ALBERT 모델을 활용하여 두 가지 병렬화 방식이 학습 시간에 미치는 영향을 분석하였다. 이를 통해 모델의 크기에 따라 최적의 병렬화 기법을 선택하는 것이 성능 향상에 필수적임을 밝히고, 분산 학습 환경에서 자원의 효율적 활용 및 성능 최적화를 위한 지침을 제시하고자 한다.

1. 서론

자연어 처리(Natural Language Processing, NLP) 는 인간의 언어를 이해하고 생성하는 인공지능의 핵심 분야로, 텍스트 분류, 기계 번역 등 다양한 응용 분야에서 중요한 역할을 하고 있다. 최근 GPU 와 같은 고성능 하드웨어와 네트워크 기술의 발전으로 인해, 더 많은 수의 파라미터를 가진 대규모 자연어 처리 모델을 훈련할 수 있는 환경이 마련되었다. 그러나 데이터의 크기와 모델의 복잡도가 증가함에 따라, 단일 장치에서 이러한 모델을 효율적으로 학습시키는 데 한계가 발생한다 [1].

이러한 문제를 해결하기 위해, 대규모 NLP 모델의 학습 속도를 가속화하고 자원 활용을 최적화하는 분산 학습 기술이 주목받고 있다. 분산 학습은 여러 장치를 활용하여 병렬로 학습을 진행함으로써, 처리 속도를 향상시킬 수 있는 방법이다. 하지만, 분산 학습에서 적용되는 병렬화 기법에 따라 학습 성능과 자원의 효율성이 크게 달라질 수 있다 [2].

본 연구에서는 IMDB(Internet Movie Database) 감정 분류 작업을 대상으로 BERT 와 ALBERT 두 모델을 사용하여 병렬화 방식이 모델 학습에 미치는 영향을 분석한다. 특히 데이터셋의 크기와 모델 복잡성에 따라 병렬화 기법이 학습 시간에 미치는 영향을 평가하여, 각 상황에 가장 적합한 병렬화 기법을 도출하고자 한다. 이를 통해, 대규모 NLP 모델의 분산 학습 환경에서 자원 효율성과 성능 최적화를 위한 지침을 제시하고자 한다.

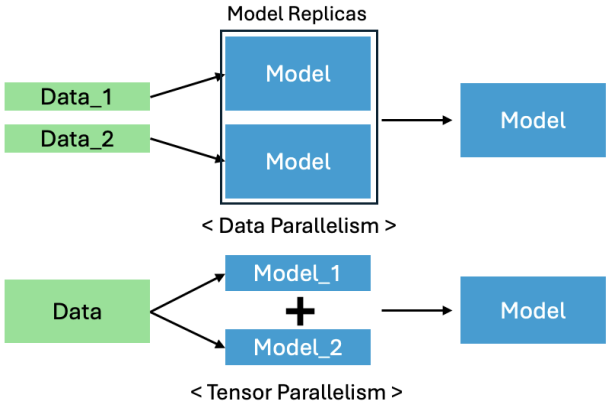
2. Data Parallelism 과 Tensor Parallelism

분산 학습은 대규모 데이터와 복잡한 모델을 여러 컴퓨팅 노드에 분산시켜 병렬로 학습시키는 기법이다. 데이터와 연산을 나눠 처리함으로써 학습 속도를 높이고 대규모 모델을 처리하기 위한 메모리 용량 확장을 위해 분산 학습을 사용한다. 본 연구에서는 NLP(Natural Language Processing) 모델에서 학습 속도에 직접적으로 영향을 끼치는 Data, Tensor Parallelism 을 사용하여 비교 분석한다.

Data Parallelism 은 동일한 모델을 여러 장치에서 서로 다른 데이터 샘플을 사용하여 병렬로 학습한다. 배치를 미니 배치로 나누고 여러 장치에 분배하여 학습을 진행하고 배치마다 장치에서 계산된 Weight Parameter 을 다양한 동기화 기법을 통해 모든 장치의 Weight Parameter 업데이트를 진행한다. 동기화는 각 에

폭 또는 미니 배치마다 이루어지며 통신 비용은 상대적으로 적지만 메모리 요구사항이 각 장치에 동일하게 발생하므로 모델 크기 제한이 존재한다.

Tensor Parallelism 은 크기가 큰 Weight Matrix 를 Row 를 기준으로 여러 장치로 나누어 연산을 한 후 그 결과값을 합치는 방법으로 모델을 학습시킨다. 각 장치는 서로 다른 텐서의 부분에 대해 연산을 진행하므로 단순히 결과를 합치기만 하면 되지만 각 Layer 마다 동기화가 이루어져야 하기 때문에 Data Parallelism 에 비해 동기화를 위한 통신 빈도가 높다. 이는 통신 효율성이 성능에 큰 영향을 미치는 것을 의미한다.



(그림 1) Data Parallelism 과 Tensor Parallelism

3. Bert Model 에서의 Data, Tensor Parallelism 비교

3.1. 실험 환경

본 실험은 2 대의 온프레미스 서버에서 각각 2 개의 Nvidia TITAN RTX 와 4 개의 Nvidia GeForce RTX 3090 을 사용하여 수행되었다. 소프트웨어 환경은 Linux 20.04, Kubernetes v1.29.8, 그리고 Kubeflow v1.9 로 구성되었다. Data Parallelism 에서는 PyTorchJob 을 활용하여 각 파드에 1 개의 GPU 를 할당하고, 6 개의 미니 배치로 나누어 학습을 진행하였다. Tensor Parallelism 은 DeepSpeed 를 사용하여 병렬화 설정을 6 으로 지정하였고, 6 개의 GPU 에서 분산 학습이 이루어지도록 구성하였다.

3.2. 사용 모델 및 데이터 셋

본 실험에서는 BERT 모델을 활용하여 Data Parallelism 과 Tensor Parallelism 의 성능을 비교 분석하였다. 모델의 크기에 따른 성능 변화를 확인하기 위해, 108M 개의 파라미터를 가진 BERT(base)와 12M 개의 파라미터를 가진 ALBERT(base) 모델을 사용하였다. 실험에 사용된 데이터셋은 50,000 개의 영화 리뷰로 구성된 IMDB(Internet Movie Database) 데이터셋이다.

3.3. 성능 평가 기준

분산 학습의 주요 목적은 모델 학습 속도의 향상과 자원 활용의 극대화에 있다 [3]. 본 실험에서는 모델 크기에 따른 학습 속도의 변화를 평가하기 위해, 성능 평가 기준으로 학습 시간만을 측정하였다. 모델 학습 시간은 Kubeflow Pipeline 의 Python 기반 컴포넌트에서 학습 시간을 기록하여 측정되었다.

3.4. 실험 결과

<표 1> Bert, ALBERT 의 Data, Tensor Parallelism 결과

Training Time(s)	Bert	ALBERT
Data Parallelism	556	264
Tensor Parallelism	392	200

실험 결과, BERT 와 ALBERT 모델 모두에서 Tensor Parallelism 이 Data Parallelism 보다 더 우수한 학습 속도를 보였다. 특히, 파라미터 수가 많은 BERT 모델의 경우, Data Parallelism 을 적용했을 때보다 Tensor Parallelism 을 적용했을 때 학습 속도가 더욱 크게 향상되었다.

4. 결론 및 향후 과제

본 연구에서는 IMDB 감정 분류 작업을 대상으로 BERT 와 ALBERT 모델을 사용하여 Data Parallelism 과 Tensor Parallelism 의 학습 속도 차이를 분석한 결과, 두 모델 모두에서 Tensor Parallelism 이 더 우수한 성능을 보였으며, 특히 파라미터 수가 많은 BERT 모델에서 그 효과가 두드러졌다. 이는 대규모 모델에서 Tensor Parallelism 의 효율성이 높음을 시사한다.

그러나 본 연구는 병렬화 기법에서 중요한 요소인 통신 비용과 동기화 기법에 대한 분석이 부족했으며, 이러한 요소가 학습 성능의 병목 현상으로 작용할 수 있음을 확인했다. 따라서 향후 연구에서는 다양한 하드웨어 환경과 모델을 대상으로 통신 비용 및 동기화 기법의 영향을 보다 정밀하게 분석하고, 이를 통해 대규모 NLP 모델의 분산 학습 성능을 최적화할 수 있는 구체적인 연구를 수행할 계획이다.

Acknowledgement

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 학석사연계 ICT 핵심인재양성사업의 연구결과로 수행되었음 (IITP-2024-RS-2023-00259867)

참고문헌

[1] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su, "Scaling Distributed Machine Learning with the Parameter Server," USENIX Symposium on OSDI, 2014.

[2] Daniel Nichols, “A Survey and Empirical Evaluation of Parallel Deep Learning Frameworks”, ACM Symposium on HPDC, 2020.

[3] Dehghani and Yazdanparast Journal of Big Data (2023) 10:158 https://doi.org/10.1186/s40537-023-00829-x