이기종 환경에서의 자원 인식 병렬화 기술 분석

김현정[1], 하지원[2], 윤현상[3], 테오도라 아두푸[4], 김윤희[4]
[1]숙명여자대학교 통계학과
[2]고려대학교 컴퓨터학과
[3]홍익대학교 컴퓨터·데이터공학부
[4]숙명여자대학교 컴퓨터과학과
email: hjkimmm@sookmyung.ac.kr, jwh0245@naver.com, fivefar@naver.com,
theoadufu@sookmyung.ac.kr, yulan@sookmyung.ac.kr

An Analysis Of Resource Aware Parallelization Techniques On Heterogeneous Environments

Hyunjeong Kim[1], Jiwon Ha[2], Hyunsang Yoon[3], Theodora Adufu[4],
Yoonhee Kim[4]

1. 서 론

AlphaGo와 같은 딥 러닝(DL) 모델의 성공으로 다양한 분야에서 기계 학습 모델 연구가 활발해졌다. 모델의 크기가 증가함에 따라 모델 처리에 multi-GPU와 같은 리소스가 사용된다. 그러나 예측 모델의 정확성에 의문이 제기되었고, GPT-3 모델은 학습 데이터에 사실로 입증되지 않은 정보가 포함되어 있기 때문에 누적 학습 접근법에서의 정확한 추론을 위해서는 미세 조정이 필요하다 [1]. 따라서 짧은 학습 시간 내에 모델이 추가 학습을 위한 데이터에 적응하도록 모델을 최적화해야 하며, 이러한 필요성은 계산 용량이 다른 컴퓨팅 장치가 학습 시간과 모델의 학습 곡선 길이에 영향을 미칠 수 있는 이기종 환경에서 훨씬 두드러진다.

이기종 환경에서는 서로 다른 컴퓨팅 기능의 자원에 대한 병렬성을 최대화하기 위해 데이터를 적절하게 분할해야 한다. 하드웨어 환경을 고려하지 않고 데이터 병렬화 기법을 단독으로 사용하면 모델 크기가 증가함에 따라 병목 현상이 발생한다.

본 연구에서는 이기종 환경에서 데이터 크기가 모델의 학습 시간에 어떠한 영향을 미치는지에 대해 연구한다. 먼저 DCGM 프로파일링 툴을 이용하여 최적화된 BERT 모델을 프로파일링하고 이를 통해 SM 및 Memory와 같은 GPU 리소스 사용량을 관찰한다 [2].

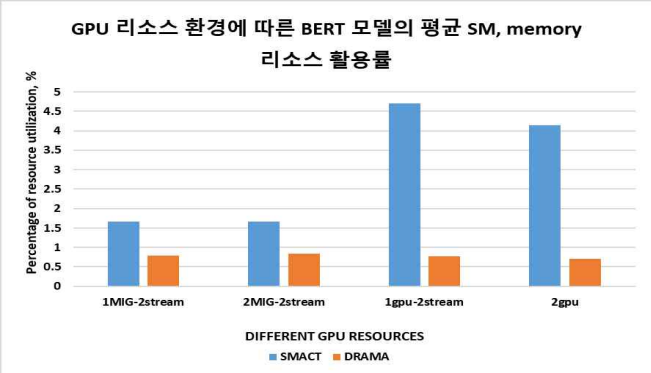
이어서 DDP(Distributed Data Parallelism)을 통해 이기종 자원 환경에서 자원 활용을 극대화하고 딥 러닝 모델의 학습 시간을 단축시킬 수 있는 자원 인식 병렬 최적화 기술을 연구한다.

2. 관련 연구

2.1 선행 실험

우리는 BERT와 같은 NLP 모델을 학습하고 추론하는 과정에서 서로 다른 리소스에 배치될 때의 리소스 사용률을 관찰하였다. 이를 위해 1g.6gb(1 Multi-Instance GPU, MIG), 2g.12gb(2MIG) 및 멀티 GPU(2GPU) 환경에서 BERT[2]의 동작을 조사하였다. 모든 실험에서 검증 결과는 2.0e-06의 오류 수준을 유지한다.

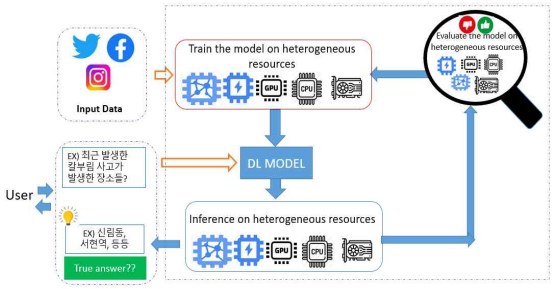
우리는 그림 1에서와 같이 BERT가 서로 다른 GPU 리소스에서 실행될 때 SM 및 메모리와 같은 GPU 리소스의 평균 사용률이 상대적으로 낮다는 것을 관찰하였다. 또한 GPU 리소스 전체에 걸쳐 SM 활동(SMACT)에 눈에 띄는 차이가 있는 반면, 메모리 대역폭을 통한 데이터 이동을 보여주는 DRAM 활동(DRAMA)은 리소스 전체에서 거의 동일함을 확인하였다. 이어서 이기종 환경에서 데이터 사이즈가 모델의 수행 시간에 미치는 영향에 대해 연구하고자 한다.



(그림 1) GPU 리소스 환경에 따른 BERT 모델의 평균 SM, memory 리소스 활용률

2.2 연구 내용

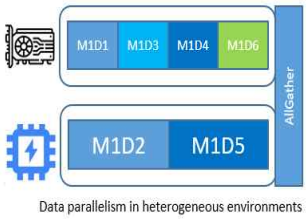
본 연구에서는 이기종 자원 환경에서 DDP 기법이 성능에 어떤 영향을 미치는지에 대해 연구한다. 우리가 제안하는 접근법은 그림 2와 같이 기본 모델 학습 구조를 따르되, 이기종 자원에서의 실행을 고려한다.



(그림 2) 제안 모델

DDP는 단일 노드의 여러 GPU 간에 또는 다중 노드 환경의 이기종 GPU 간에 모델 학습을 위한 데이터셋이 공유되는

data parallelism 기술이다. 이기종 환경에서 DDP를 사용하는 동안, 서로 다른 종류의 GPU에 대한 데이터의 비동기적 이동을 최대화하기 위해 데이터를 CPU 메모리에 고정시킨다.

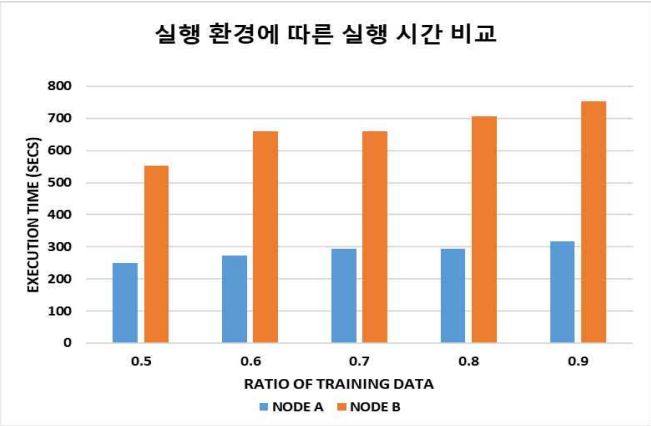


(그림 3) 데이터 병렬화

이어서 distributed data sampler를 통해 데이터셋이 사용 가능한 모든 GPU 리소스들로 분할된다. Distributed data sampler는 GPU 리소스 간에 데이터가 겹치지 않도록 보장한다. 그림 3과 같이 동일한 모델(M1)을 모든 GPU 리소스에 배포하여 서로 다른 데이터셋(D1-D5)에서 각 작업을 수행하고, AllGather 프로세스를 통해 학습 결과가 반환된다. GPU 간의 데이터 이동은 NVLink를 통해 이루어진다. DDP는 Multi-GPU 모델에서 방대한 양의 데이터가 포함된 크기가 큰 모델을 학습시킬 때 특히 더 효율적이다. Whale[4]과 같은 선행 연구들은 모델들 간의 의존성을 식별하고, 이기종 환경에서 그림 3과 같은 데이터 병렬화 기법과 모델 병렬화 기법을 결합하는 방법을 제안한다.

2.3 실험 결과

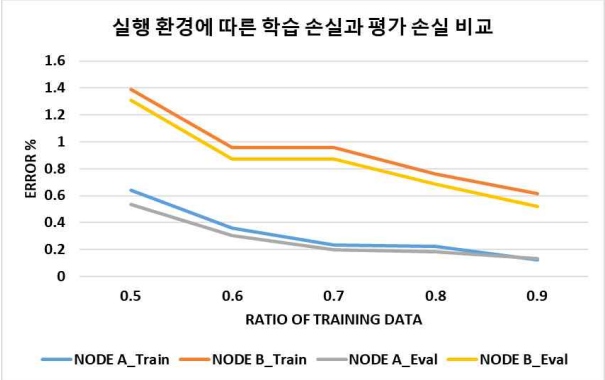
우선 PyTorch의 DDP 기법을 통해 minGPT[7] NLP 모델을 각각 NODE A와 NODE B에서 학습시킨다. NODE A는 NVIDIA A30(Compute capability 8.0, Device memory 24GB) GPU 2대를, NODE B는 NVIDIA TITAN XP(Compute capability 8.0, Device memory 24GB) GPU 4대를 가진다. 데이터셋으로는 CharGPT의 벤치마크 데이터셋인 TinyShakespeare을 사용한다.



(그림 4) 실행 환경에 따른 실행 시간 비교

Torchrun 패키지를 이용해 minGPT 모델을 실행하여 학습 및 평가 데이터셋을 각 노드에 다양한 백분율로 분할한다. 그림 4는 각 경우의 학습 손실, 평가 손실을, 그림 5는 NODE A, B에 대한 총 실행 시간 측정 결과를 보여준다.

실험 결과, NODE B에 더 많은 GPU가 있음에도 불구하고, NODE B에서 minGPT의 수행 시간이 NODE A에서의 수행 시간의 2배임을 확인하였다. 이는 다양한 유형의 GPU에 데이터를 분할할 때 GPU 성능을 고려해야 한다는 것을 의미한다.



(그림 5) 실행 환경에 따른 학습 손실과 평가 손실 비교

또한 NODE B의 경우와 같이 학습 데이터 비율이 줄어들수록, 학습 단계와 평가 단계 모두에서 오류 퍼센티지가 증가하는 것을 확인하였다.

3. 결론

본 연구의 실험 결과에 따르면, 모델 학습에 **DDP** 기술을 적용하는 것은 기본 리소스의 성능이 높을 때만 효과적일 수 있다. 학습 시간을 줄이면서 정확도를 향상시키는 데 필요한 데이터셋의 크기도 고려해야 한다. 향후 연구로는 NLP 모델이 아닌 다른 DL 모델의 리소스 사용량 추가 조사 및 FSDP(Fully Sharded Data Parallel) 등 다른 병렬 최적화 기술에 대한 연구를 진행하려고 한다.

이 논문은 2023년도 정부재원(과학기술정보통신부 여대학원생 공학연구팀제 지원사업)으로 과학기술정보통신부와 한국여성과학기술인육성재단의 지원을 받아 연구되었습니다.

참 고 문 헌

[1] "Microsoft launches the new Bing, with ChatGPT built in", https://techcrunch.com/2023/02/07/microsoft-launches-the-new-bing-with-chatgpt-built-in/ (Last accessed 6th March, 2023)

[2] NLP-Fast, https://github.com/SNU-HPCS/NLP-Fast/tree/main

[3] Hanhwi Jang, Joonsung Kim, Jae-Eon Jo, Jaewon Lee, and Jangwoo Kim. 2019. MnnFast: a fast and scalable system architecture for memory-augmented neural networks. In Proceedings of the 46th International Symposium on Computer Architecture (ISCA '19). Association for Computing Machinery, New York, NY, USA, 250 - 263. https://doi.org/10.1145/3307650.3322214

[4] Wang, Ang & Jia, Xianyan & Jiang, Le & Zhang, Jie & Li, Yong & Lin, Wei. (2020). Whale: A Unified Distributed Training Fram9ework.

[5] "Getting Started with Distributed Data Parallel – PyTorch Tutorials 2.0.1+cu117 documentation", https://pytorch.org/tutorials/intermediate/ddp_tutorial.html

[6] https://pytorch.org/tutorials/intermediate/ddp_tutorial.html

[7] minGPT, https://github.com/karpathy/minGPT