불균형한 DNN 모델의 효율적인 분산 학습을 위한 파라미터 샤딩 기술 성능 평가

최기봉, 고윤용, 김상욱†
한양대학교 컴퓨터소프트웨어학과
rlqhd26@hanyang.ac.kr, yyko@hanyang.ac.kr, wook@hanyang.ac.kr

Performance Evaluation: Parameter Sharding approaches for DNN Models with a Very Large Layer

Ki-Bong Choi, Yun-Yong Ko, Sang-Wook Kim
Dept. of Computer and Software, Hanyang University

요 약

최근 딥 러닝 (deep learning) 기술의 큰 발전으로 기존 기계 학습 분야의 기술들이 성공적으로 해결하지 못하던 많은 문제들을 해결할 수 있게 되었다. 이러한 딥 러닝의 학습 과정은 매우 많은 연산을 요구하기에 다수의 노드들로 모델을 학습하는 분산 학습 (distributed training) 기술이 연구되었다. 대표적인 분산 학습 기법으로 파라미터 서버 기반의 분산 학습 기법들이 있으며, 이 기법들은 파라미터 서버 노드가 학습의 병목이 될 수 있다는 한계를 갖는다. 본 논문에서는 이러한 파라미터 서버 병목 문제를 해결하는 파라미터 샤딩 기법에 대해 소개하고, 각 기법 별 학습 성능을 비교하고 그 결과를 분석하였다.

1. 서론

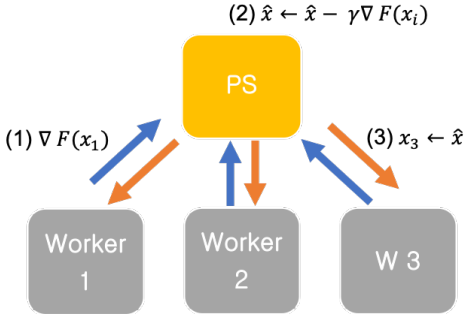
딥 러닝 기술의 발전으로 인하여 기존의 기계 학습 기술들이 성공적으로 해결하지 못했던 다양한 문제들을 해결할 수 있게 되었다. 하지만 최대 수억 개 이상의 파라미터로 이루어진 거대한 모델을 빅 데이터로 학습하는 것은 매우 많은 연산을 요구하기에 다수의 학습 노드들이 네트워크로 서로 연결되어 있는 분산 컴퓨팅 환경에서 모델을 학습하는 분산 학습 기술이 연구되었다[1][2][3].

대표적인 분산 학습 방법으로 파라미터 서버 기반 분산 학습 방안이 있다. 파라미터 서버 기반 분산 학습에서 각 노드는 전체 학습 데이터를 분할하여 저장하고 해당 데이터를 학습하는 워커 노드와, 각 워커 노드의 학습 결과를 합산하고 학습 모델의 파라미터를 관리하는 파라미터 서버 노드로 나누어진다.

(그림 1) 은 파라미터 서버 기반 분산 학습의 과정을 보여준다. (1) 각 워커는 자신의 모델을 학습하여 그 결과를 파라미터 서버에 전송하고, (2) 파라미터 서버는 각 워커의 학습 결과로 전역 모델 파라미터를 업데이트하고 (3) 해당 워커에게 업데이트된 전역 모델 파라미터를 전송한다. 워커는 파라미터 서버 에게 받은 모델 파라미터를 기반으로 다음 학습을 진행한다.

파라미터 서버 기반 분산 학습에서는 모든 워커의 학습 결과를 파라미터 서버에서 처리하므로 파라미터 서버가 통신, 연산 의 두 측면에서 전체 학습과정의 병목이 될 수 있다. 이러한 병목 문제를 완화시키는 방안으로 다수의 파라미터 서버를 두어 각 파라미터 서버가 전체 모델 파라미터의 일부만을 관리하도록 하는 파라미터 샤딩 기술이 있다[4].

본 논문에서는 두개의 파라미터 샤딩 기술을 소개하고, 각 샤딩 기술의 효과를 여러 분산 환경의 노드 구성에서 보인다.



(그림 1) 파라미터 서버 기반 분산 학습.

† 교신저자

2. 파라미터 샤딩 기술

계층 기반 파라미터 샤딩

각 계층을 파라미터 서버에게 할당하는 방법으로, 대부분의 기계 학습 프레임워크 (TensorFlow, PyTorch) 에서 계층을 연산의 기본 단위로 삼기에 계층을 기본 단위로 한 계층 기반 파라미터 샤딩이 가장 일반적으로 사용된다. 계층 기반 파라미터 샤딩을 사용할 경우, 각 파라미터 서버에 할당된 파라미터의 수가 다를 수 있다.

메모리 기반 파라미터 샤딩

계층을 기본 단위로 하지 않고, 메모리를 기본 단위로 하여 각 파라미터 서버 에 동일한 수의 파라미터를 할당하는 방법이다. 계층 기반 파라미터 샤딩의 경우, 일부 계층에 대부분의 모델 파라미터가 집중되어있는 모델을 학습 할 때 파라미터 샤딩의 효과를 거의 보지 못하지만 메모리 기반 파라미터 샤딩은 모델의 구조에 상관없이 파라미터 샤딩의 효과를 볼 수 있다.

본 논문의 실험 파트에서는 일부 계층에 대부분의 학습 파라미터가 집중되어있는 모델을 학습 할 때, 각 파라미터 샤딩 기술의 효과를 여러 분산 환경의 노드 구성에서 보인다.

3. 실험

3.1 실험 환경

실험 데이터 및 모델 : 실험에 사용한 데이터는 1000 개의 레이블과 1,280,000 개의 학습 데이터, 50,000 개의 테스트 데이터로 구성된 imagetNet-1K 데이터를 사용하였다. 모델은 VGG-16[2] 모델을 사용하였다. VGG-16 모델은 총 138M 개의 모델 파라미터로 이루어져있고, 그 중 거의 대부분인 100M 개의 모델 파라미터가 하나의 계층에 집중되어있다.

하드웨어 구성 : 실험에 사용한 컴퓨터는 i7-9700K CPU, 64GM RAM, RTX2080Ti GPU*2 가 장착되었으며 각 컴퓨터는 Mellanox ConnetX-4Lx 를 사용하여 10G 이더넷으로 서로 연결되어있다.

구현 : 모든 구현은 ubuntu 18.04, TensorFlow 1.15 MPICH 3.14 를 사용하여 이루어졌다.

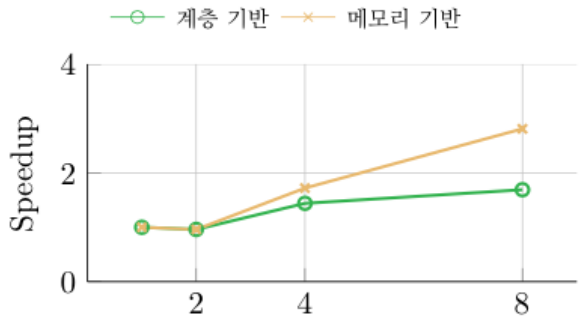
분산 환경 노드 구성 : 워커 노드 2 개 당 1 개의 파라미터 서버 노드를 사용하였다.

3.2 실험 결과

워커 수 증가에 따른 계층 기반 파라미터 샤딩과 메모리 기반 파라미터 샤딩의 학습 정확도의 차이가 없기에 본 논문의 실험결과에서는 학습 속도의 차이만 보인다.

(그림 2) 은 워커 수 증가에 따른 학습 속도의 증가 비율을 나타낸다. 워커의 수가 1 개, 2 개일 경우에는 파라미터 서버의 수가 1 개이기 때문에 계층 기반 파라미터 샤딩과 메모리 기반 파라미터 샤딩이 동일한 성능을 보이지만 워커의 수가 4 개인 경우에는 메모리 기반 파라미터 샤딩이 계층 기반 파라미터 샤딩에 비하여 20%, 8 개인 경우에는 66% 의 성능 차이를 보인다.

이는 VGG-16 같은 일부 계층에 대부분의 모델 파라미터가 집중되어있는 모델의 경우, 계층 기반 파라미터 샤딩을 하였을 때 대부분의 학습 파라미터가 집중되어있는 계층을 담당하는 파라미터 서버가 병목이 되기 때문이다.



(그림 2) 워커 수 증가에 따른 학습 속도 증가

4. 결론

본 논문에서는 현재 활발히 연구되고 있는 파라미터 서버 기반 분산 학습과 파라미터 샤딩 기술을 소개하였다. 실험을 통하여 일부 계층 에 대부분의 학습 파라미터가 집중되어있는 모델을 학습할 경우 계층 기반 파라미터 샤딩과 메모리 기반 파라미터 샤딩의 성능 차이를 확인하였다.

사사

이 성과는 (1)정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원 (No.NRF-2020R1A2B5B03001960), (2)정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원 (No. 2020-0-01373, 인공지능대학원지원(한양대학교)), (3)2017 년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No. NRF-2017M3C4A7069440).

참고문헌

[1] Krizhevsky et al, "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[2] Very deep con- volutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations. 2015.

[3] Ho, Qirong, et al. "More effective distributed ml via a stale synchronous parallel parameter server." Advances in neural information processing systems. 2013.

[4] Dean, Jeffrey, et al. "Large scale distributed deep networks." Advances in neural information processing systems. 2012.