

[HW2] Distributed DL training on Parallax

2018-28588 KANG SUKYONG

Find an open-sourced TensorFlow model to run

<https://github.com/tensorflow/models>에서 제공하고 있는 tensorflow모델인 im2txt를 선택하였습니다. 구글이 공개한 image captioning모델이고 Show and Tell이라는 이름으로 arXiv에 논문이 공개되어 있습니다.

Make the model run on a distributed environment using vanilla TensorFlow, Horovod, and Parallax

<https://github.com/tensorflow/models>의 release에 im2txt를 사용하였습니다. 3가지 distribute environment에서 동작하는 코드를 작성하기 위해 im2txt내부의 train.py를 참고하였습니다. 모델은 inception_v3를 사용하였습니다. 우분투 16.04에서 개발하였고, 시간상의 이유로 dataset은 cocodataset.org의 2017 Val images를 사용하였습니다. 각 script에 대한 실행 방법은 script상단에 주석으로 표기하였습니다. Required package나 training data를 preprocessing하는 과정 등은 <https://github.com/tensorflow/models/tree/master/research/im2txt>를 참조하시면 됩니다.

Check your distributed programs run correctly by comparing the results

checkpoint파일을 확인해본 결과 tf, horovod, parallax 세 경우의 결과가 모두 달랐습니다. tf에서는 synchronous training을 위해 SyncReplicasOptimizer를 사용하는데, 이는 모든 worker들이 동일한 variable을 읽어 들이도록 하기 위해 모든 replica로부터 gradient를 collect하고 이 값들의 평균을 이용하여 variable에 적용시킵니다. 이 방식으로 stale한 gradient를 방지하는 barrier역할을 합니다. MapReduce방식을 사용하는 horovod에서는 느린 worker를 탐지한 이후부터 backup worker를 실행시키고 parallax에서는 생성된 n개의 update중 시간 순서대로 m개의 update를 취하게 됩니다.

이러한 특성으로 인하여 서로의 결과가 다르게 나오는 것이라고 판단하였습니다.

Compare the performance of vanilla TensorFlow, Horovod, and Parallax

모든 training은 CPU(i7-7700HQ)로 하였습니다. 약 20기가의 training set을 돌리는 데에 시간적인 문제가 따르기도 했고 또한 상당히 많은 트레이닝 횟수가 필요했기 때문에 1기가 크기의 validation set을 사용했습니다. 200번의 트레이닝을 했을 때 처음 100번 까지의 결과를 제외하고 그 이후의 throughput의 평균은 tf, horovod, parallax 순서대로 0.157594, 0.150986, 0.137008이 나왔습니다. 기존 vanilla tf의 경우 따로 straggler에 대한 체크를 하지 않고 단순한 barrier의 역할만 하기 때문에 throughput이 가장 높은 것으로 생각됩니다. horovod와 parallax모두 straggler를 체크합니다. 하지만 horovod의 경우 straggler가 탐지된 이후부터 backup worker를 실행하고, parallax는 straggler가 탐지된 경우 가장 빠르게 실행된 m개의 update를 취해야 하기 때문에 backup worker가 계속 동작하는 것으로 판단하였습니다. 때문에 horovod의 throughput이 parallax에 비해 높은 값을 가지는 것으로 분석하였습니다.