MegatronLM

목차

Problem to solve

- Memory용량의 한계로 large model을 학습하기 어려운 문제가 있음

Points to learn

- transformer model의 memory 요구량 계산 방법
- 이 논문에서 제안하는 model parallelism(Tensor parallel)
- Tensor parallel size를 최대 8로 하는 이유
- 분산처리시 scaling efficiency에 대한 이해

Background

- 모델이 커짐에 따라 메모리 최적화 기술이 개발되어옴
    - activation checkpoining
    - ADAM optimizer state partitioning
- GPipe, Mesh-tensorflow와 같은 기술이 있는데 이걸 쓰려면 custom compiler가 필요하고 model을 수정해야 함
- 이 논문에서 제시하는 idea는 pytorch내부에 구현함으로써 쉽게 적용이 가능함 (model은 조금만 수정하면 됨)

Background(Transformer)

Operation analysis

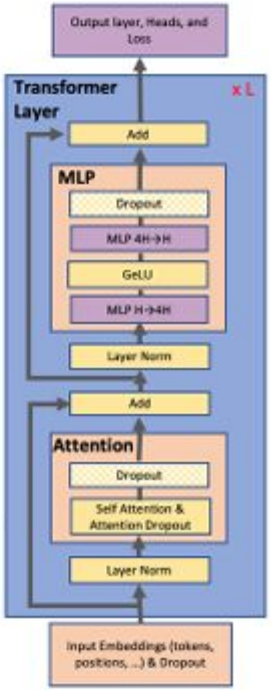
Batch size	Embedding dimension	Layer size	Attention head size	input length	output length
B	D	L	H	T	O

Attention

name	vector1	vector2	output
q	(B x T x D)	(H x D x D/H)	(B x H x T x D/H)
k	(B x T x D)	(H x D x D/H)	(B x H x T x D/H)
v	(B x T x D)	(H x D x D/H)	(B x H x T x D/H)
attention score	(B x H x T x D/H)	(B x H x D/H x T)	(B x H x T x T)
attention value	(B x H x T x T)	(B x H x T x D/H)	(B x H x T x D/H) = (B x T x D)
attention output	(B x T x D)	(D x D)	(B x T x D)

Feed Forward Neural Network

name	vector1	vector2	output
ffnn1	(B x T x D)	(D x 4D)	(B x T x 4D)
ffnn2	(B x T x 4D)	(4D x D)	(B x T x D)



Background(Transformer)

Operation analysis

Batch size	Embedding dimension	Layer size	Attention head size	input length	output length
B	D	L	H	T	O

Attention

name	vector1	vector2	output
q	(B x T x D)	(H x D x D/H)	(B x H x T x D/H)
k	(B x T x D)	(H x D x D/H)	(B x H x T x D/H)
v	(B x T x D)	(H x D x D/H)	(B x H x T x D/H)
attention score	(B x H x T x D/H)	(B x H x D/H x T)	(B x H x T x T)
attention value	(B x H x T x T)	(B x H x T x D/H)	(B x H x T x D/H) = (B x T x D)
attention output	(B x T x D)	(D x D)	(B x T x D)

Feed Forward Neural Network

name	vector1	vector2	output
ffnn1	(B x T x D)	(D x 4D)	(B x T x 4D)
ffnn2	(B x T x 4D)	(4D x D)	(B x T x D)

1. Forward
   a. FLOPS
      - $2*BH(12TD^2 + 2T^2D)$
   b. Memory
      i. Model weight
         - $2*12HD^2$
      ii. Activation
         - model weight에 비해 매우 작음

2. Backward
   a. model size * 20 Bytes

Background

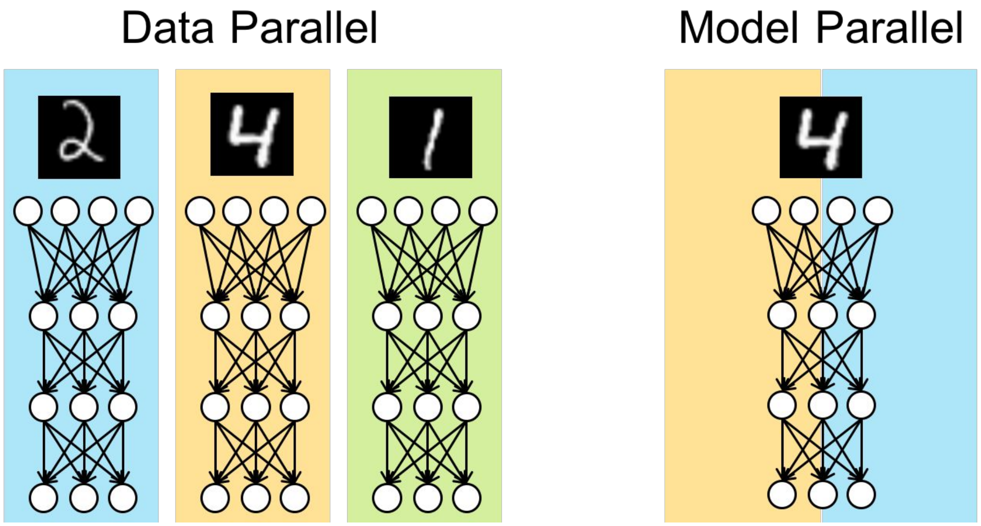
NVIDIA V100 사양

	V100 for NVLink	V100 for PCIe	V100S for PCIe
PERFORMANCE with NVIDIA GPU Boost™	DOUBLE-PRECISION 7.8 teraFLOPS	DOUBLE-PRECISION 7 teraFLOPS	DOUBLE-PRECISION 8.2 teraFLOPS
	SINGLE-PRECISION 15.7 teraFLOPS	SINGLE-PRECISION 14 teraFLOPS	SINGLE-PRECISION 16.4 teraFLOPS
	DEEP LEARNING 125 teraFLOPS	DEEP LEARNING 112 teraFLOPS	DEEP LEARNING 130 teraFLOPS
INTERCONNECT BANDWIDTH Bi-Directional	NVLINK 300 GB/s	PCIE 32 GB/s	PCIE 32 GB/s
MEMORY CoWoS Stacked HBM2	CAPACITY 32/16 GB HBM2		CAPACITY 32 GB HBM2
	BANDWIDTH 900 GB/s		BANDWIDTH 1134 GB/s

Target model
- BERT, 8.3B

Training시 필요 memory
= 8.3B * 20 = 166GBytes

Background(Data parallelism)



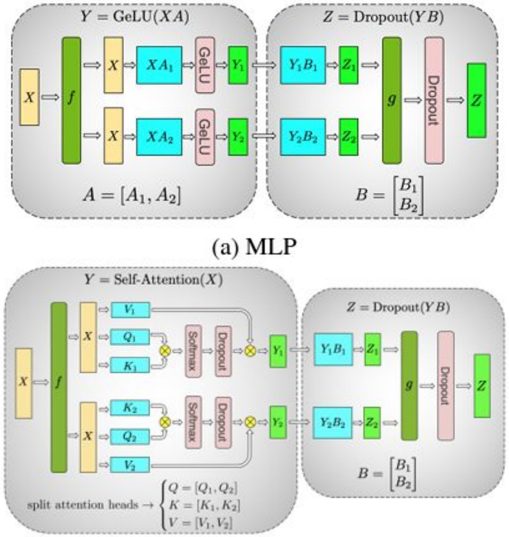
Data parallel

- input data, 즉 batch size에 대해서 parallel
- training시 backward pass에 대해서 all reduce 필요함

Model parallel

- 하나의 model 연산에 필요한 memory를 parallel
- 이 과정에서 연산도 parallel됨
- forward pass과정에서도 all reduce 필요함

Solution



(a) MLP

(b) Self-Attention

Figure 3. Blocks of Transformer with Model Parallelism. f and g are conjugate. f is an identity operator in the forward pass and all reduce in the backward pass while g is an all reduce in the forward pass and identity in the backward pass.

Attention

1. model weight를 device에 나눠서 load
2. input X를 각각의 device에 복사
3. QKV를 구할 때 head로 나눠서 구함(column parallel)
4. attention output weight는 row parallel로 계산
5. attention output에 대해서 all reduce

FFNN

1. ffnn1의 weight는 column parallel
2. ffnn2의 weight는 row parallel
3. all reduce

Solution

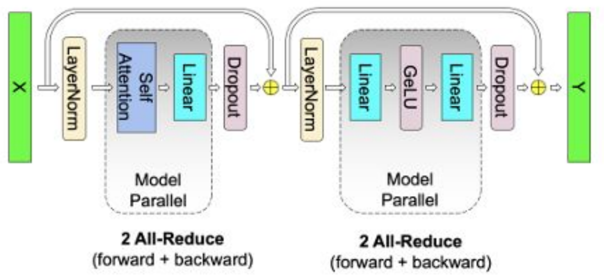


Figure 4. Communication operations in a transformer layer. There are 4 total communication operations in the forward and backward pass of a single model parallel transformer layer.

Solution

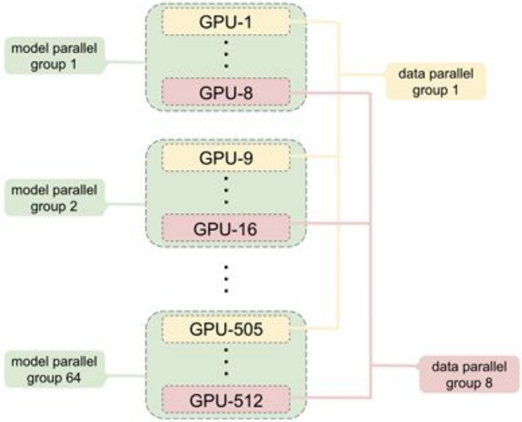
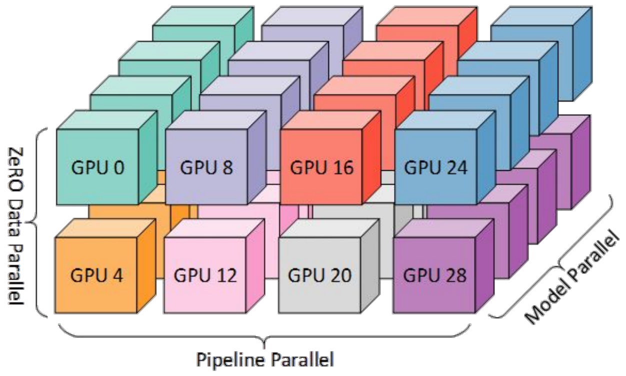


Figure 8. Grouping of GPUs for hybrid model and data parallelism with 8-way model parallel and 64-way data parallel.

Solution(3D Parallel)



Experiments

Table 1. Parameters used for scaling studies. Hidden size per attention head is kept constant at 96.

Hidden Size	Attention heads	Number of layers	Number of parameters (billions)	Model parallel GPUs	Model +data parallel GPUs
1536	16	40	1.2	1	64
1920	20	54	2.5	2	128
2304	24	64	4.2	4	256
3072	32	72	8.3	8	512

32 DGX-2H
total 512 V100 32GB GPUs

Results

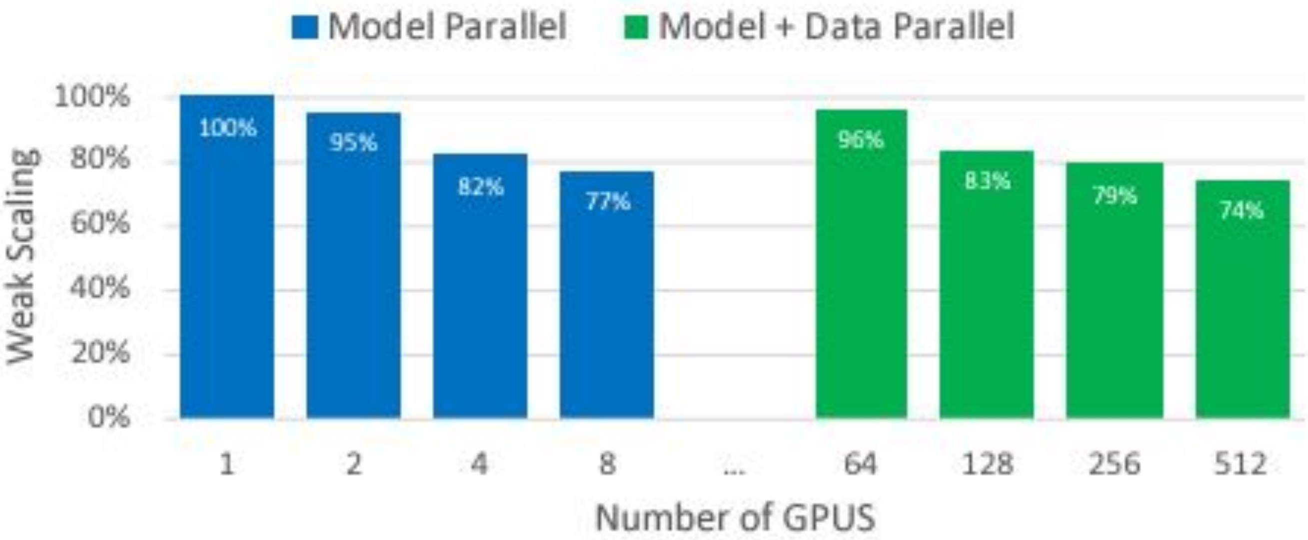


Figure 5. Model and model + data parallel weak scaling efficiency as a function of the number of GPUs.