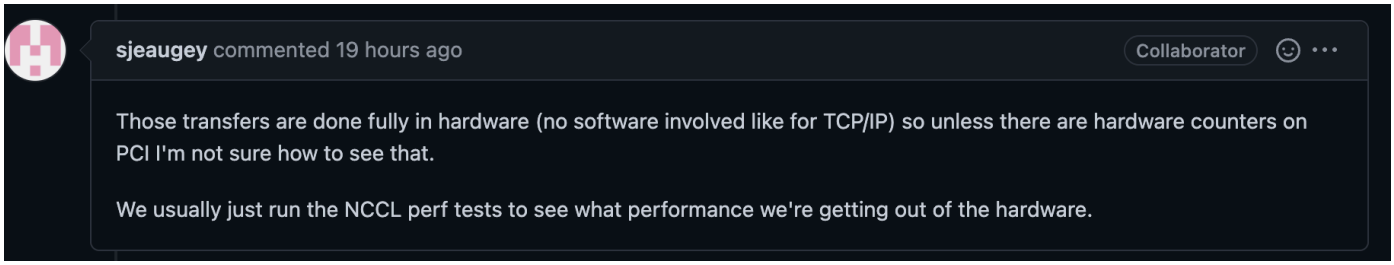
How to detect the bandwidth of nccl transfer

The answer is we can only calculate the time of `dist.recv` and the bytes of data.

https://github.com/NVIDIA/nccl/issues/669



Results:

Backbone	**MoblieNetV2**
Dataset	CIFAR10
Batchsize	64
Chunk	8
Separate	First 1 last 1

compression method	**average bandwidth**	**Average_batch_time**
None	8.85GB/S,1.19GB/S	0.39s~0.2s
Quantization 8 bits	2.23GB/S,0.37GB/S	0.41s~0.2s
Sort Quantization8bits (6bits 2bits split)	2.20GB/S,0.38GB/S	0.51s~0.2s
Quantization 10bits	2.81GB/S, 0.49GB/S	0.63s~0.1s
Quantization 12bits	3.17GB/S ,0.70GB/S	0.74s~0.1s
Quantization 16bits	4.41GB/S,0.80GB/S	0.43s~0.2s
Sort Quantization 12bits 4bits split	4.40GB/S,0.76GB/S	0.61s~0.2s

There is a linear relationship between bandwidth and data size.

The command of reproduction is shown in my repo.

However it shows that each time bandwidth and time has a little difference. I got the average of them.

About Deadlock

First I met a deadlock when I use two GPUs in one distributed group(default).

https://github.com/pytorch/pytorch/issues/75795

```
I could draw a graph for you.
time line ------------------------------------------------------------------------------
----------------
gpu0: input[0]->layer1->output(isend)
------------------------------------- input[1]->layer1->output(isend)
---------------------------------------------------------------------- recv1(recv)-
>hangs(x)(😡)->layer3
gpu1:
------------------------------------- recv1 -> layer2->output(isend)
----------------------------------------------------------------------recv2 -> layer2-
>output(isend) over(😄)
```

The solution is to give multi distributed groups. the group number is equal to the chunk number.

This works well.

However this will cause another problem if you use 3 or 4 GPUs and the chunk number is more than 4, **this will cause Deadlock, too.**

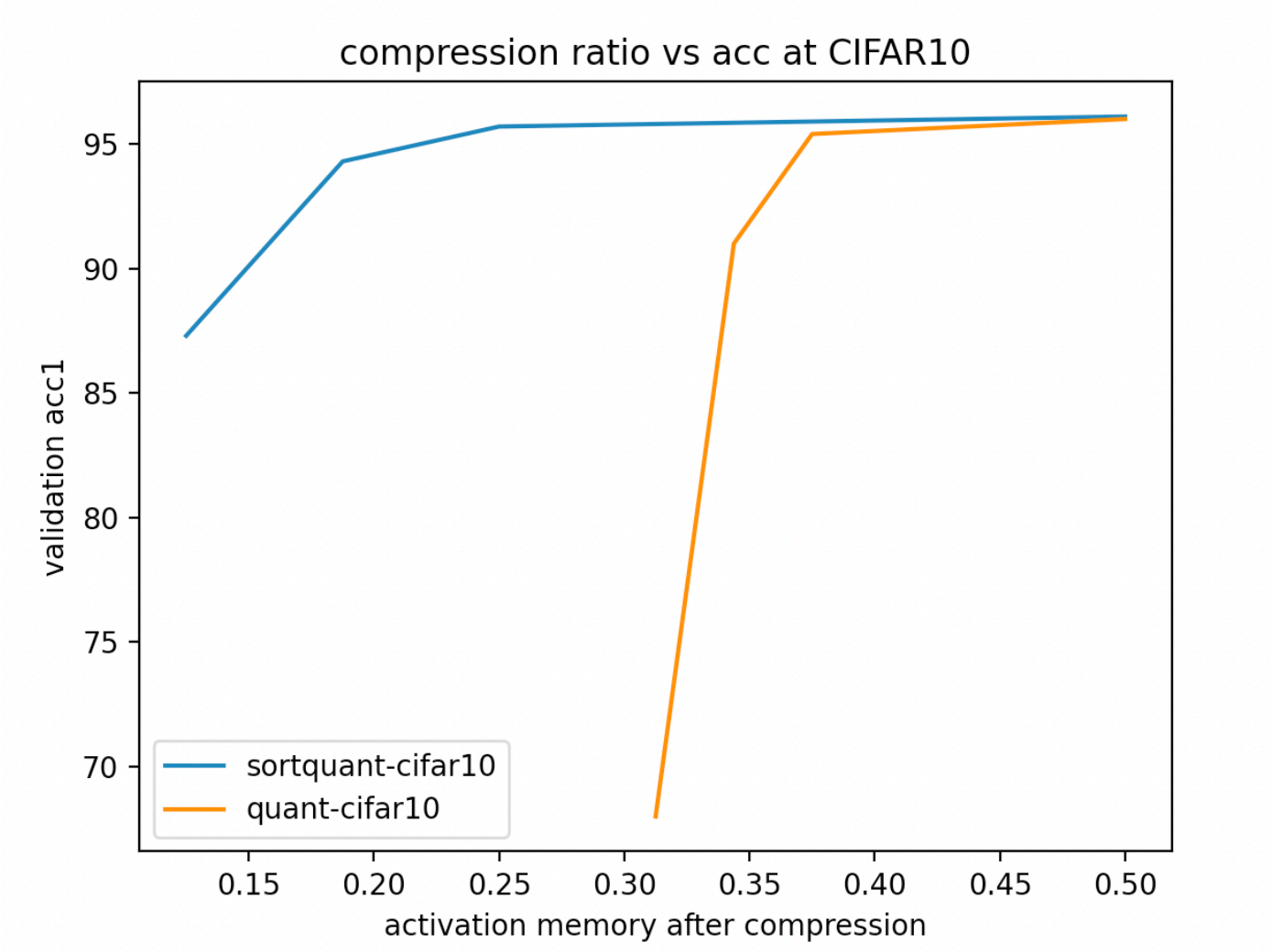
So, the conclusion is if I just use two GPUs, I create multiple groups. If we use more than two GPUs, we create one group.

I have tested it several times, and it is proved to be true.

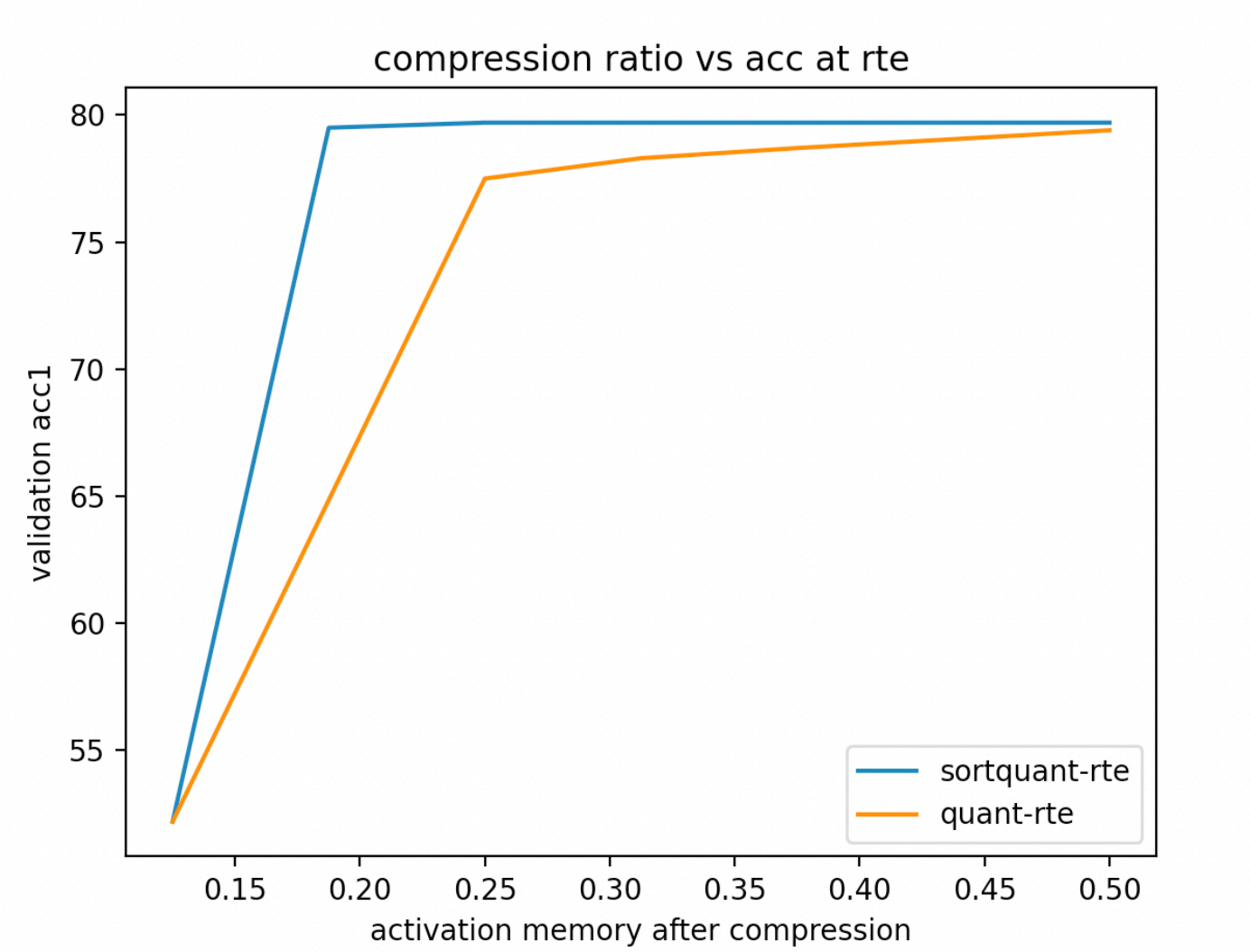
I will continue to find the cause of these problems, but now they are not a threaten of our experiment.

Ablation study and other results

MobileNetV2 with CIFAR10



RTE dataset with Roberta



comparing to kcluster

Settings	Method	Input size	Time per batch	Acc
CIFAR10 MobileNetV2 10epoch	K-means 4bits(20 iter)	[16,24,56,56]	0.66s	93.01%
CIFAR10 MobileNetV2 10epoch	K-means 4bits(50 iter)	[16,24,56,56]	1.33s	93.17%
CIFAR10 MobileNetV2 10epoch	Quantization 4bits	[16,24,56,56]	0.10s	89.42%
CIFAR10 MobileNetV2 10epoch	Sort Quantization 4bits(4splits,2bits)	[16,24,56,56]	0.10s	93.38%
RTE Roberta-base 20epochs	K-meas 6bits(50 iter)	[8,128,786](the last two layer)	3.05s	79.4%
RTE Roberta-base 20epochs	Quantization 6bits	[8,128,786](the last two layer)	0.4s	52.2%
RTE Roberta-base 20epochs	Sort Quantization 6bits(3bits, 8splits)	[8,128,786](the last two layer)	0.4s	75.0%
Cola Roberta-base 20epochs	K-meas 6bits(100 iter)	[8,128,786](the last two layer)	1.32s	0.633～0.006(Matthew)
Cola Roberta-base 20epochs	Sort Quantization 6bits(3bits, 8splits)	[8,128,786](the last two layer)	0.4s	0.591～0.006(Matthew)
Cola Roberta-base 20epochs	Quantization 6bits	[8,128,786](the last two layer)	0.4s	0.587 ～0.007(Matthew)