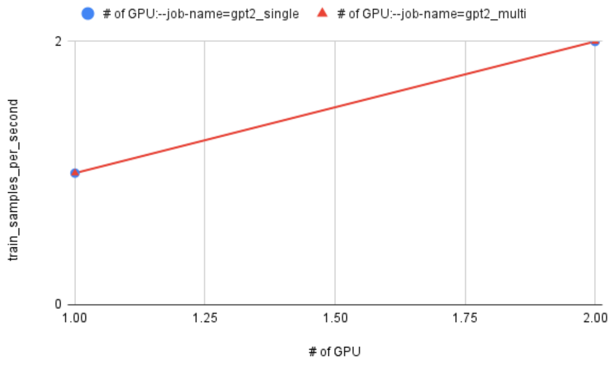
lab7_109062320

Draw the strong scalability of data parallel training



Explain why such observation

Since in whatever training sample, the `train_sample_per_second` will not change, I will only focus on the data of `gpt2_single`.
By observing the data, we can see that using more GPUs allows for more parallel processing of the data. This means that more training samples can be processed simultaneously, leading to a higher throughput (more samples per second).
However, we observe only sub-linear scaling, and I guess it's because of the following reasons:

1. `Communication Overhead`: More GPUs entail more data synchronization, which can become a bottleneck, especially with a higher number of GPUs.
2. `Data Loading and Preprocessing Bottlenecks`: If the data pipeline isn't optimized to match the GPUs' processing speed, it can limit the rate at which data is available for training, thus reducing overall efficiency.
3. `Inefficient Utilization of GPUs`: Not all models or training configurations fully utilize the computational power of multiple GPUs, which can lead to underutilization and decreased scaling efficiency.

Your experiment process

Modifying the Script for Different GPU Counts:
For job-name=gpt2_single/gpt2_multi: Set `--nproc_per_node=1, 2` and `SBATCH --gres=gpu:1, 2`. The experiment use single and 2 GPU to do experiment on `gpt2_single/gpt2_multi`, which has 200/400 training sample