COMPUTING AND
INFORMATION TECHNOLOGY
Research Computing and Data

LLM fine-tuning

How can we maximize computational efficiency?

CLEMSON

Batch size considerations

Batch size is the number of training samples processed simultaneously during training.

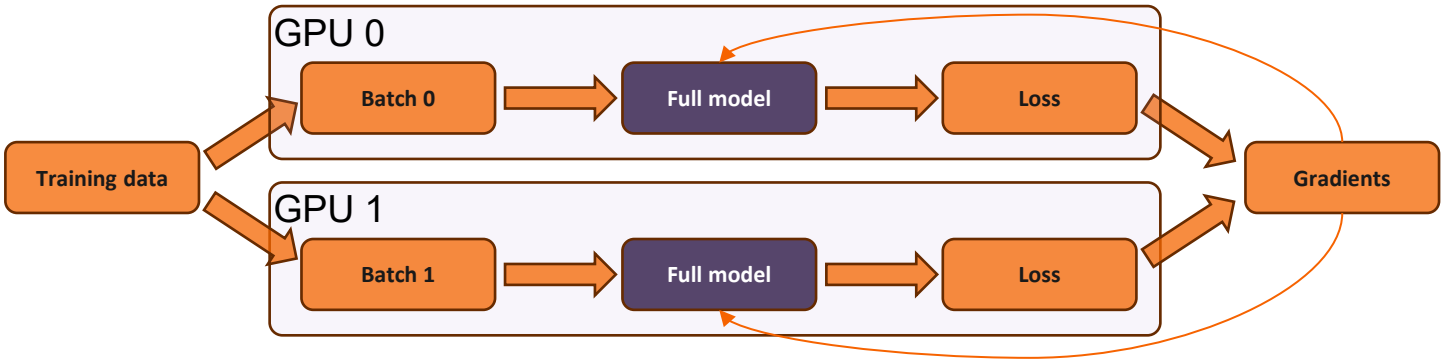
Batch size affects how much memory is used during training, **and** affects training outcomes.

Batch size	Advantages	Disadvantages
Large	• Faster training per epoch • More stable gradients • Better GPU utilization	• Risk of poor generalization • Can lead to training instability • High memory usage
Small	• More frequent updates and faster convergence • Better generalization • Lower memory usage	• Noisier gradients and training instability • Slower per-epoch training • May require gradient accumulation to match large batch performance

Data parallelism

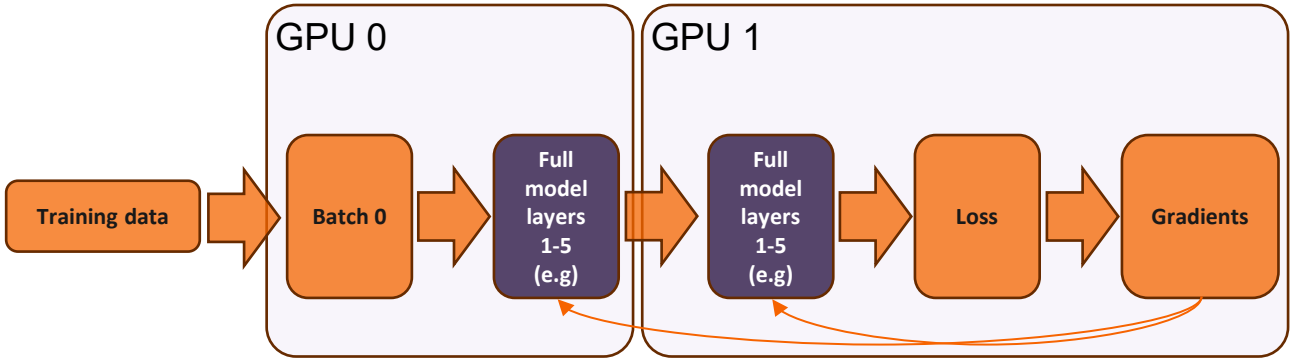
There are many ways to use **multiple GPUs** during training, with different goals and outcomes.

Data parallelism puts a copy of the model on **each** GPU, speeding training.



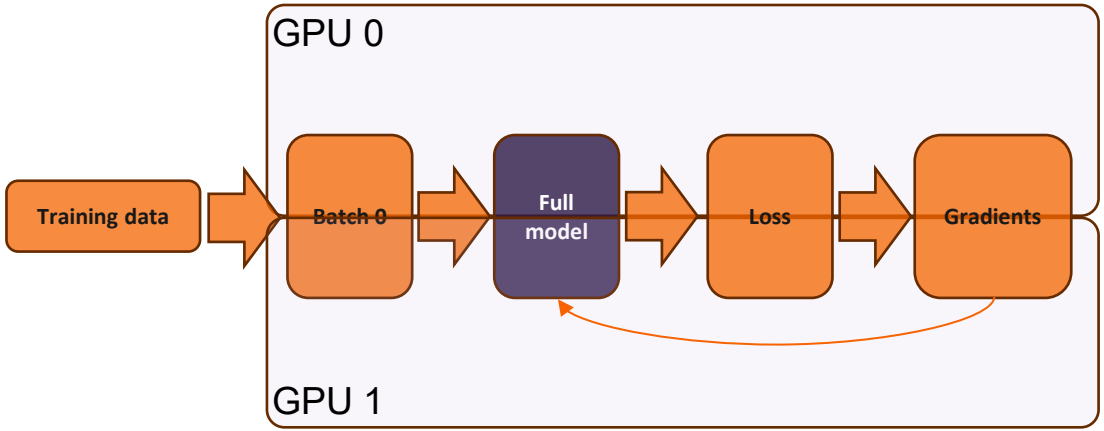
Model parallelism

Model parallelism puts a some of the layers of the model on each GPU, allowing the use of a model that is too large to fit on a single GPU.



Tensor parallelism

Tensor parallelism splits individual layers of the model across GPUs, allowing the use of a model with individual layers so large they can't fit on a single GPU.



Comparison of parallelism strategies

Parallelism type	What it splits	Best for	Pros	Cons
Data parallelism	Dataset (batch split across GPUs)	• Large datasets • Small to medium models	• Easy to implement • Works with most models	• Requires full model copy on each GPU • Some overhead
Model parallelism	Model (some layers on each GPU)	Large models that don't fit on one GPU	• Reduces memory load per GPU • Usually easy to implement	• Slower due to inter-GPU communication • Can be tough to implement
Tensor parallelism	Individual layers (weights split across models)	Extremely large models with layers too big for 1 GPU	• Enables massive model scaling • More memory efficient than model parallelism	• High communication overhead • Very difficult to implement, requires specialized libraries