Deep learning engineering

Why does this matter?

1. We need accelerators (GPU or TPU) to run neural networks efficiently.
2. Accelerators have constraints. (e.g. a fixed amount of memory)
3. Neural nets keep getting bigger.

(2018 - 2022
ELMo PaLM
100M 540B
5000x increase)

$$\hat{y}_i = f_\theta(x_i)$$

$$\ldots \underbrace{W h \ldots}_{f_\theta}$$

$$\partial\theta = \sum_{i=1}^{N} \nabla_\theta L(\hat{y}_i, y_i)$$

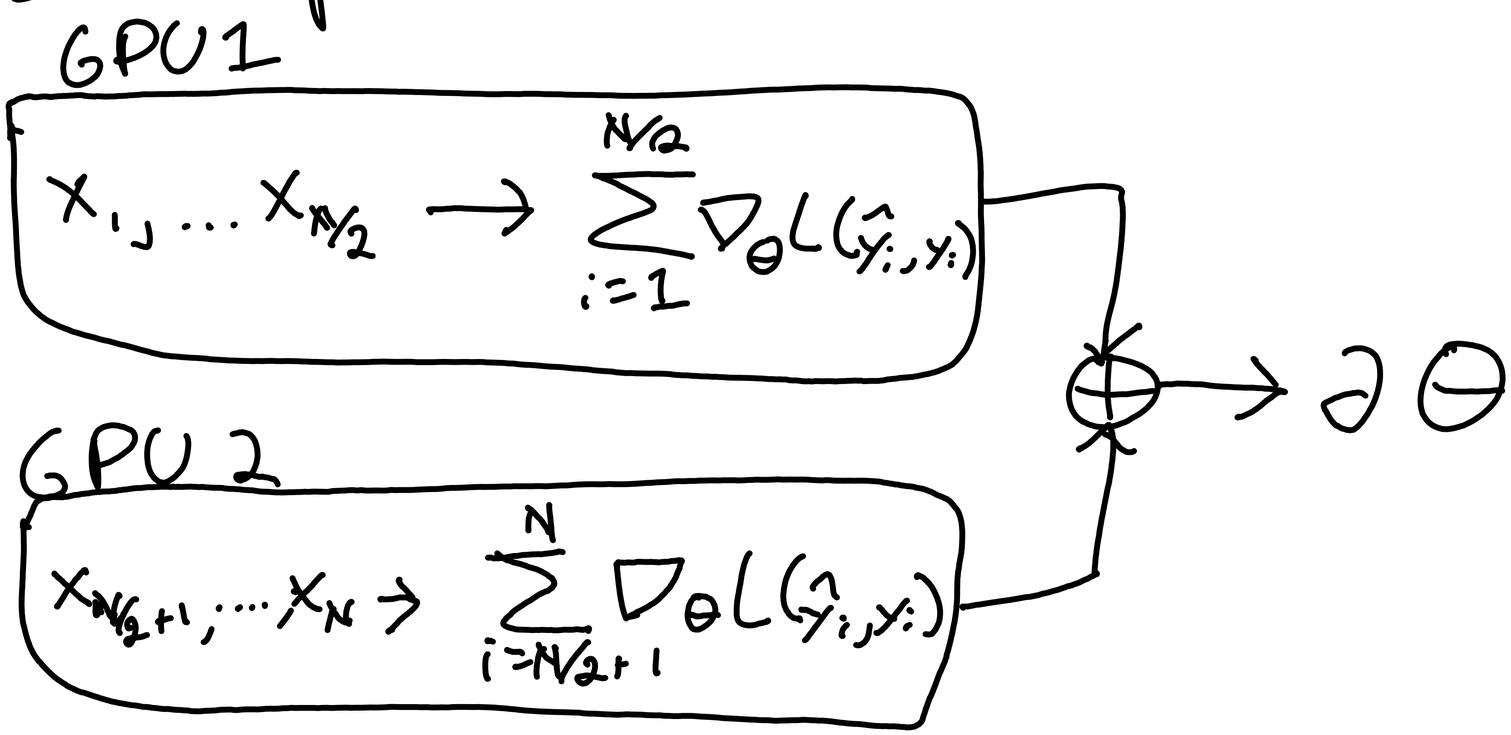
$$\theta \leftarrow \theta + \text{optimizer}(\partial\theta)$$

Memory needs:
- Storing h for backprop
- Parameters
- Batch size
- Optimizer states

Compute needs:
- Computing f_θ
- Computing ∇_θ
- Batch size

What do we do when a single GPU won't suffice?

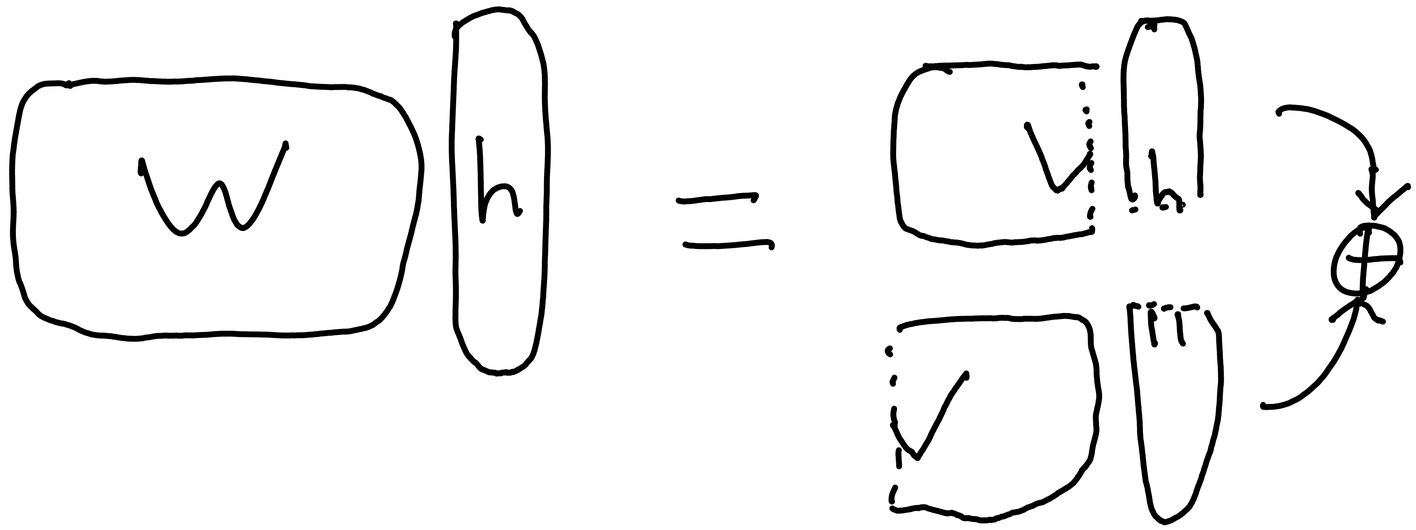
1. Data parallelism



Pros: Simple. Low communication.

Cons: Depends on batch size.
Can't support a larger model than can fit on one GPU.

2. Tensor parallelism

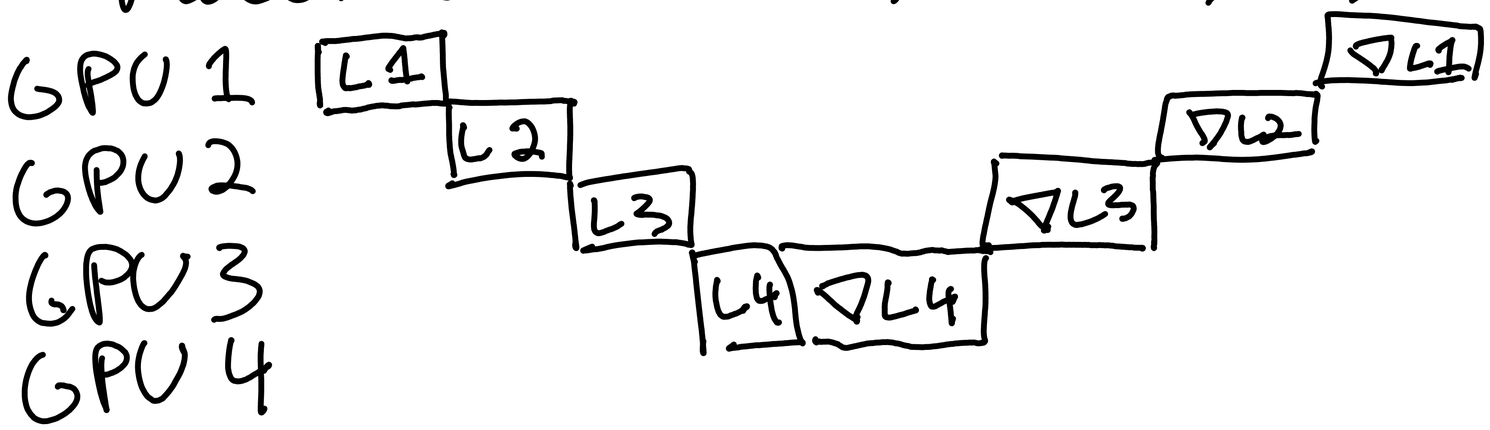


Pros: Fit larger weight matrices (larger models)

Cons: Communication intensive.

3. Pipeline parallelism

Model with 4 layers L1, L2, L3, L4



Pros: Easy to split the model, low-ish communication

Cons: Devices are mostly idle (bubble)

How to make things faster in general?

1. Floating point precision
   Typical, but large: float32 (8 bits for exponent)
   Smaller, but unstable: float16 (5 bits for exponent)
   Compromise: bfloat16 (8 bits for exponent)
2. "Mixed precision": Use high-precision weights and low precision for everything else.
3. "Gradient accumulation": Accumalate gradients over "micro batches" (split up batch)
4. "Gradient rematerialization": Recompute activations rather than caching them

Fairness, Accountability, Transparency

Task: Classify people

Remark: System (ML model) could have different performance on different subgroups.

ex Identify/diagnose skin cancer, what if it worked worse on certain skin colors?

ex Facial recognition accuracy of early APIs was shown to be dramatically worse for dark-skinned women.

ex Prisoning sentencing system trained on biased data.

How to measure this?

Simplest: Measure performance seperately on subgroups

Question: How to do this objectively?

Where does this come from?

One way: Train on $x, y \sim p(x, y)$

Test on $x, y \sim \hat{p}(x, y)$

(violates i.i.d. assumption)

Another way: Issues with the dataset.

ex/ Language model trained on problematic data (doesn't reflect my values), it can (and should) mimic that data.

Problem: Datasets are often "unfathomably large" - can't manually audit.

Representativeness:

Who does this data represent? Whose values?
Who decided how to collect/curate the data?

ex/ Wikipedia is a "clean" source of text
but it was overwhelmingly written by
white men.

ex "Bad words" filters in C4.

Privacy / consent:

ex/ Internet users didn't have LMs in mind.

LMs can memorize content.

→ privacy issue.

Did they "want" this?

→ consent issue.

Unclear how much a model "remixing"

Is this legal?

Issues with modeling:
Data has an impact on model behavior, but other choices can have an impact too.

ex/ Dataset with 49 images of light-skinned people, 50 of dark-skinned people.
Consider two generative models:
- Fits a "zero-covariance Gaussian" (average)
- Outputs median face.

ex/ Model compression: Get a smaller/faster model that achives ~ the same accuracy.
Current techniques tend to sacrifice accuracy on the "long tail"

Auditing

who should be paying attention / responsible for notiLing all of these issues?

- Indastry? Misaligned incentives
- Watch dog group? "Hole plugging"
- Impacted people? Double burden
- Researchers? "Dual use"

Alignment

Models are increasingly being deployed in high-stakes situations where they directly interacting with people and expected to do anything.

How to characterize the desired behavior?

- Helpful (does what it's asked)
- Harmless (don't cause harm to people)
- Honest (accurately reflects information)

Issues could be exacerbated if we expand the action space.