

DBRX: Training Modern LLMs From Scratch

Abhinav Venigalla & Jonathan Frankle
NLP Architect & Chief AI Scientist



Introducing DBRX

D B R X



 **databricks**
mosaic research

Our Mission

Help everyone build and serve **custom AI models...**

...using their own **unique data...**

...to achieve the highest quality on **their domain...**

...as **efficiently and cost-effectively** as possible.

We can take you from API calls to full pretraining.

What is DBRX?

An open LLM built entirely at Databricks.

132B parameters. Behaves like 36B parameters.

A leading open model on popular benchmarks.

Why did we build DBRX?

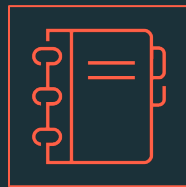
It wasn't to build the world's best model.

We upgraded our LLM training stack.

Stress testing and improving Databricks for GenAI.



Lilac AI for
data exploration
and curation



Notebooks and
Apache Spark for
data cleaning and
processing



Unity Catalog for
data storage and
governance



Mosaic AI
Pretraining to
train the model



MLflow and
Lakeview for
experiment tracking

Why did we build DBRX?

It wasn't to build the world's best model.

We upgraded our LLM training stack.

Stress testing and improving Databricks for GenAI.

This talk: how to build custom DBRX-class models.

Friendly Advice

Friendly Advice

Start small and work your way up.

Don't trust what you read in the literature.
Test everything for yourself.

Don't trust intuition, received wisdom, or a rumor.
Test everything for yourself.

Do the math.

Friendly Advice

Start small and work your way up.

Don't trust what you read in the literature.

Let Databricks be your research team.

Don't trust intuition, received wisdom, or a rumor.

Let Databricks be your research team.

Do the math together. We have your back.

Roadmap

Roadmap

Define success (**evaluation**)

Understand your budget (**model and data size**)

Fill in the details (**which model and data**)

And then you train.. (**scaling and infrastructure**)

Evaluation

You can't make progress until you know what success looks like.

Evaluation: what you need

Something cheap and automatic.

Something somewhat involved and more realistic.

Something close to the real world.
(Can be slow and expensive.)

Evaluation: what you need

Something cheap and automatic.

- Your inner development loop.
- Has right and wrong answers.
- For DBRX: the Mosaic Gauntlet

Evaluation: what you need

Calibrating the Mosaic Evaluation Gauntlet

A good benchmark is one that clearly shows which models are better and which are worse. The Databricks Mosaic Research team is dedicated to finding great measurement tools that allow researchers to evaluate experiments. The Mosaic Evaluation Gauntlet is our set of benchmarks for evaluating the quality of models and is composed of 39 publicly available benchmarks split across 6 core competencies: language understanding, reading comprehension, symbolic problem solving, world knowledge, commonsense, and programming. In order to prioritize the metrics that are most useful for research tasks across model scales, we tested the benchmarks using a series of increasingly advanced models.

by [Tessa Barton](#)

April 30, 2024 in [Mosaic AI Research](#)

Evaluation: what you need

Something somewhat involved and more realistic.

- Evaluates the generative behavior of the model.
- Likely uses LLM-as-a-judge.
- For DBRX: MTBench, IFEval, Arena Hard.

Evaluation: what you need

Something close to the real world.

- Real human evaluation.
- Slots into an existing workflow for A/B testing.
- For DBRX: Human annotation, customer feedback.
- For image models: Human preferences in product.

Read your evaluation sets and results

HellaSwag: Can a Machine *Really* Finish Your Sentence?

Rowan Zellers[♠] Ari Holtzman[♠] Yonatan Bisk[♠] Ali Farhadi^{♠♡} Yejin Choi^{♠♡}

[♠]Paul G. Allen School of Computer Science & Engineering, University of Washington

[♡]Allen Institute for Artificial Intelligence

<https://rowanzellers.com/hellaswag>

Read your evaluation sets and results

A woman is outside with a bucket and a dog. The dog is running around trying to avoid a bath. She...

- A. rinses the bucket off with soap and blow dry the dog's head.
- B. uses a hose to keep it from getting soapy.
- C. gets the dog wet, then it runs away again.**
- D. gets into a bath tub with the dog.

Read your evaluation sets and results



Robert McHardy
@robert_mchardy



 Are We Done with MMLU?

In our new paper "Are We Done with MMLU?" we identify errors in MMLU and find that some subsets are riddled with errors. We propose MMLU-Redux with 3,000 re-annotated questions across 30 subjects.

Read your evaluation sets and results

So many “errors” on BIRD Bench are in fact the model making a reasonable guess at the user’s unclear query or providing helpful context.

We estimate that somewhere around 70% of GPT-4’s “mistakes” on BIRD Bench should be marked as correct.

Internal Slack

Read your evaluation sets and results

Inflection-2.5: meet the world's best personal AI

Palo Alto, CA – March 7, 2024

We also evaluated our models on MT-Bench, a widely used community leaderboard to compare models. However, after evaluating MT-Bench, we realized that a large fraction—nearly 25%—of examples in the reasoning, math, and coding categories had incorrect reference solutions or questions with flawed premises. Therefore, we corrected these examples and release that version of the dataset here.

Inflection Blog

My deepest, darkest fear

Customer: We just spent \$\$\$\$\$ pretraining an LLM with you.

Customer:is it good?

We must know how good it will be beforehand.

We should easily be able to verify afterwards.

Model and Data Size

Understand your budget and constraints. Plan accordingly.

Attempt 1: Training Compute Cost

You have a budget of \$. Train the best model.

The cost of training \approx model size \times data size.

Extreme 1: Train a giant model on very little data.

Extreme 2: Train a tiny model on tons of data.

The answer is somewhere in between. But where?

Attempt 1: Training Compute Cost

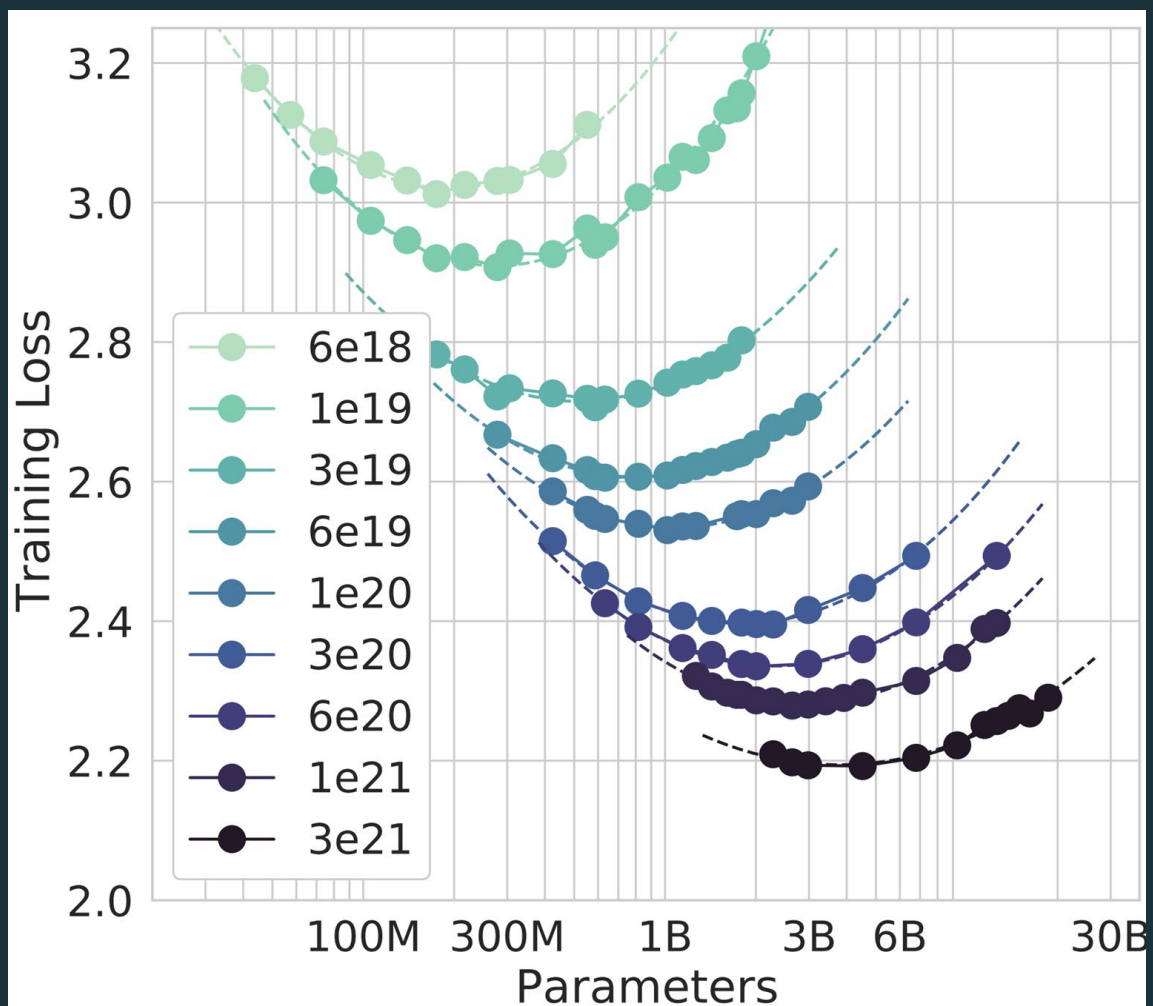
The Chinchilla paper. Tokens = 20 x Parameters.

Training Compute-Optimal Large Language Models

Jordan Hoffmann*, Sebastian Borgeaud*, Arthur Mensch*, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals and Laurent Sifre*

*Equal contributions

Attempt 1: Training Compute Cost



Attempt 2: Lifecycle Compute Cost

Train the best model and perform inference.

Worth training a smaller-than-optimal model to reduce inference cost.

Also has the benefit of simplifying training.

Attempt 2: Lifecycle Compute Cost

Train the best model and perform inference.

LLaMA: Open and Efficient Foundation Language Models

Hugo Touvron*, Thibaut Lavril*, Gautier Izacard*, Xavier Martinet
Marie-Anne Lachaux, Timothee Lacroix, Baptiste Rozière, Naman Goyal
Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin
Edouard Grave*, Guillaume Lample*

Meta AI

Model	Chinchilla	Llama2-7B	Llama2-70B	Llama3-8B	Llama3-70B
TPR Ratio	20	285	28.5	1875	214.2

Attempt 3: Compute + Data Cost



- | | |
|------------------------|----------------------------|
| 1. Pretraining | ~10T tokens, general data |
| 2. Curriculum Learning | ~1T tokens, higher quality |
| 3. Fine-Tuning | ~10K–100K instructions |
| 4. RLHF | ~10K–100K preferences |

Attempt 3: Compute + Data Cost

1

2

3

4



The Pile An 800GB Dataset of Diverse Text for Language Modeling

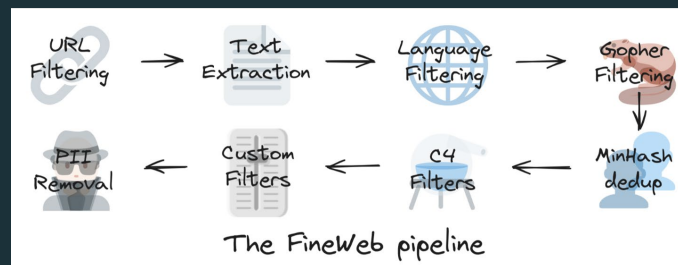
What is the Pile?

The Pile is a **825 GiB** diverse, open source language modelling data set that consists of 22 smaller, high-quality datasets combined together.

[Pile Paper \(arXiv\)](#)



Your Data



Attempt 3: Compute + Data Cost

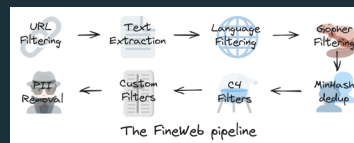
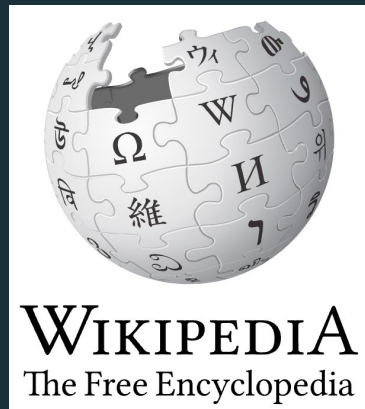
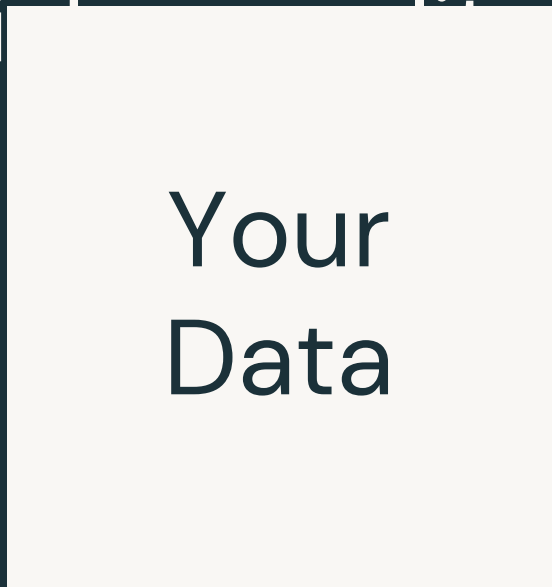
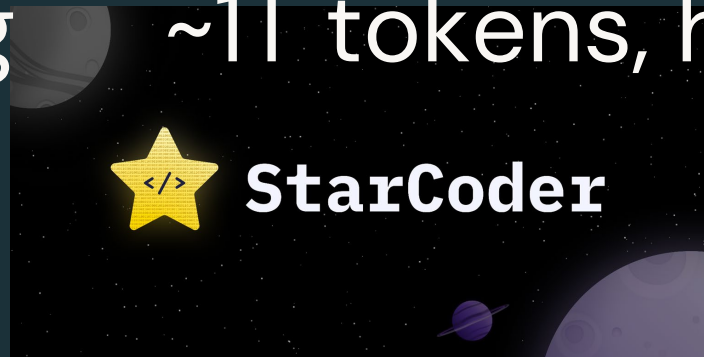
1

2

3

4

2. Curriculum Learning ~1T tokens, high quality



Attempt 3: Compute + Data Cost

1

2

3

4

3. Fine Tuning 10K-100K instructions

Your Data

Human
Annotation
 $O(\$10-100)$

Attempt 3: Compute + Data Cost

1

2

3

4

3. Fine-Tuning

~10K-100K instructions

Quality Is All You Need. Third-party SFT data is available from many different sources, but we found that many of these have insufficient diversity and quality — in particular for aligning LLMs towards dialogue-style instructions. As a result, we focused first on collecting several thousand examples of high-quality SFT data, as illustrated in Table 5. By setting aside millions of examples from third-party datasets and using fewer but higher-quality examples from our own vendor-based annotation efforts, our results notably improved. These findings are similar in spirit to Zhou et al. (2023), which also finds that a limited set of clean instruction-tuning data can be sufficient to reach a high level of quality. **We found that SFT annotations in the order of tens of thousands was enough to achieve a high-quality result. We stopped annotating SFT after collecting a total of 27,540 annotations.** Note that we do not include any Meta user data.

Attempt 3: Compute + Data Cost

1

2

3

4

3. Fine-Tuning

~10K–100K instructions

We also observed that different annotation platforms and vendors can result in markedly different downstream model performance, highlighting the importance of data checks even when using vendors to source annotations. To validate our data quality, we carefully examined a set of 180 examples, comparing the annotations provided by humans with the samples generated by the model through manual scrutiny. Surprisingly, we found that the outputs sampled from the resulting SFT model were often competitive with SFT data handwritten by human annotators, suggesting that we could reprioritize and devote more annotation effort to preference-based annotation for RLHF.

Attempt 3: Compute + Data Cost

1

2

3

4

3. Fine Tuning 10K 100K instructions

Your Data

Synthetic
Data

Human
Annotation
 $O(\$10-100)$

Attempt 3: Compute + Data Cost

1

2

3

4

4. RLHF 10K 100K preferences

Your Data

Human
Annotation
 $O(\$5-20)$

Attempt 3: Compute + Data Cost

1

2

3

4

4. RLHF

~10K–100K preferences

Table 26 shows detailed statistics on Meta human preference data. In total, we collected 14 batches of human preference data (i.e., Meta Safety + Helpfulness) on a weekly basis, consisting of over 1 million binary model generation comparisons. In general, later batches contain more samples as we onboard more annotators over time and the annotators also become more familiar with the tasks and thus have better work efficiency. We also intentionally collect more multi-turn samples to increase the complexity of RLHF data and thus the average number of tokens per sample also increase accordingly over batches.

Which Data?

You are what you train on.

Exercise: Build a 1T Token Pretraining Set

Dataset	Size
Web data	2.4T tokens
Code data	400B tokens
Wikipedia English	7B Tokens
Wikipedia Other	47B Tokens
Science papers	60B Tokens
Literature	5B Tokens

Distribute evenly? Upsample certain datasets?

Exercise: Build a 1T Token Pretraining Set



What is your goal with this model? Evaluation!

Exercise: Build a 1T Token Pretraining Set

Dataset	Size
Web data	2.4T tokens
Code data	400B tokens
Wikipedia English	7B Tokens
Wikipedia Other	47B Tokens
Science papers	60B Tokens
Literature	5B Tokens

Distribute evenly? Upsample certain datasets?

The Original Llama Dataset

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

The Value of Better Data

Arch.	Tokens	Dataset	Gauntlet Score
MPT-7B	1000B	MPT (Apr 2023)	30.9%

The Value of Better Data

Arch.	Tokens	Dataset	Gauntlet Score
MPT-7B	1000B	MPT (Apr 2023)	30.9%
MPT-7B	1000B	DBRX (Jan 2024)	39.0%

Updated dataset leads to 8.1pp improvement.

The Value of Better Data

Arch.	Tokens	Dataset	Gauntlet Score
MPT-7B	1000B	MPT (Apr 2023)	30.9%
MPT-7B	500B	DBRX (Jan 2024)	32.1%

With a better dataset, we get a better model with half as much data.

Key Questions About Data

How should you mix data? Freshness vs. repetition.

Quality vs. Quantity

Should you deduplicate your data?

Run experiments. Let science be your guide.

How to run experiments

Start small and work your way up.

How to run experiments

Start with small models and see how your metrics improve as you scale.

Risk: Your metrics may not have signal until a certain scale.

Must train a 7B model on 2T tokens to get signal on a popular coding benchmark (HumanEval).

How to run experiments

Get a running start.

How to run experiments

1

2

3

4

Do something reasonable during pretraining.

Experiment extensively during curriculum learning.

As our customer, you can get intermediate DBRX checkpoints to use for curriculum learning

Data Logistics

You have TBs or PBs of data.

Where do you put it?

How do you get it to your training cluster?

How do you shuffle it?

How do you ensure determinism?

Data Logistics

www.github.com/mosaicml/streaming



Fast, accurate streaming of training data from cloud storage

[\[Website\]](#) - [\[Getting Started\]](#) - [\[Docs\]](#) - [\[We're Hiring!\]](#)

python 3.8 | 3.9 | 3.10 | pypi v0.5.2 | Test passing | Downloads/month 40k | docs passing | slack chat | License Apache 2.0

 **Welcome**

Which Model?

Spoiler: It's going to be a transformer.

Our Advice: Follow the Beaten Path

Train a transformer.

Perform next-token prediction.

Use quadratic attention.

Follow the Llama scaling rules.

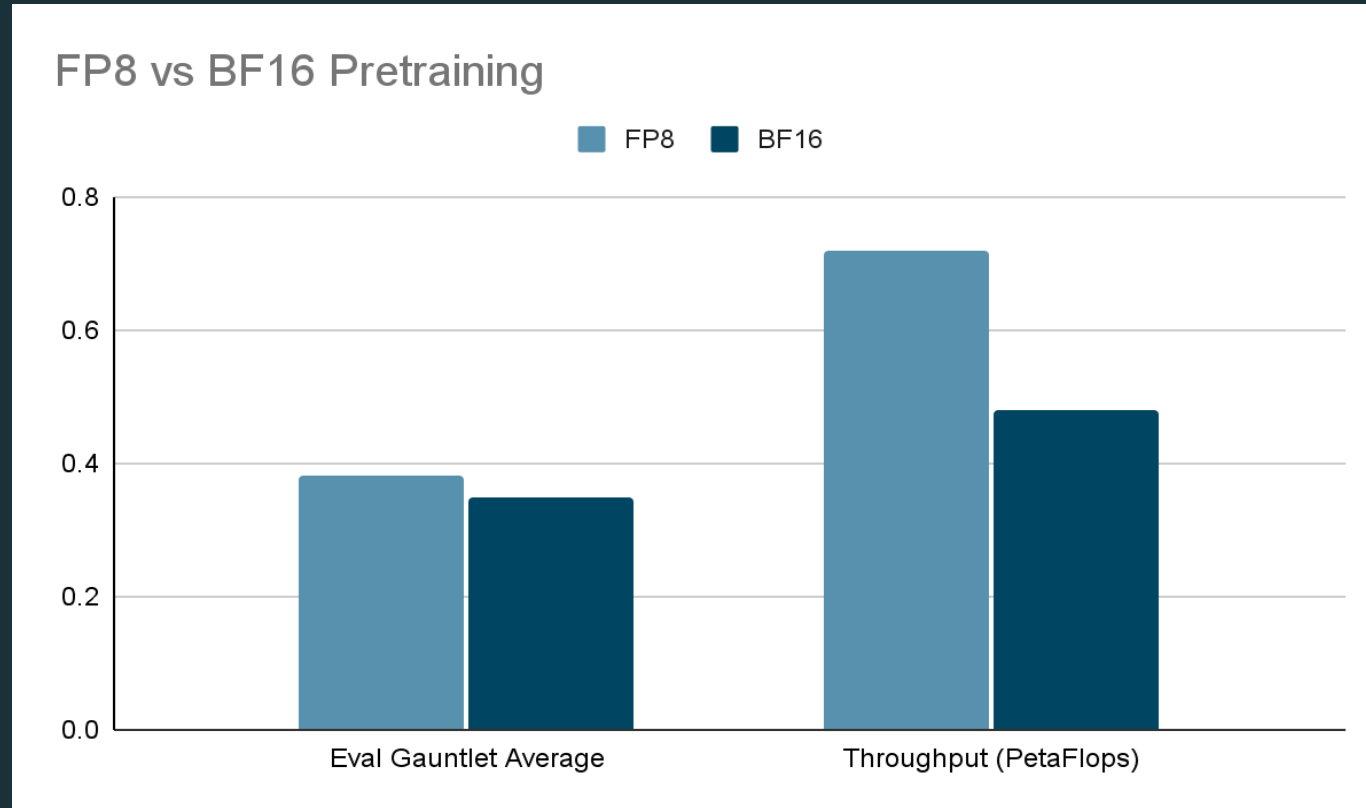
For advanced users: Use RoPE and Adam

Our Latest Stack

FP8 and Mixture-of-Experts (MoE)

- FP8 = precision we use for matrix multiplication
- MoE = model architecture

Training in FP8



FP8 training on H100 is 1.5x faster than BF16 in practice.

Mixture-of-Experts: TLDR

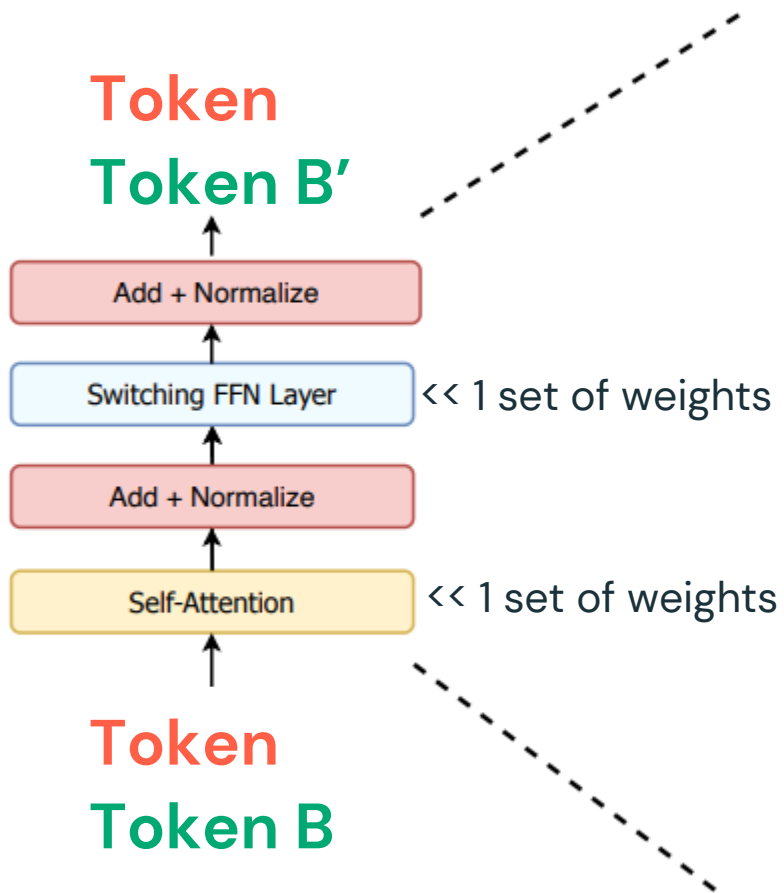
Bigger models are better than smaller ones.

Bigger models are slower than smaller ones.

Insight: Use a big model, but only activate a small part of it for any input.

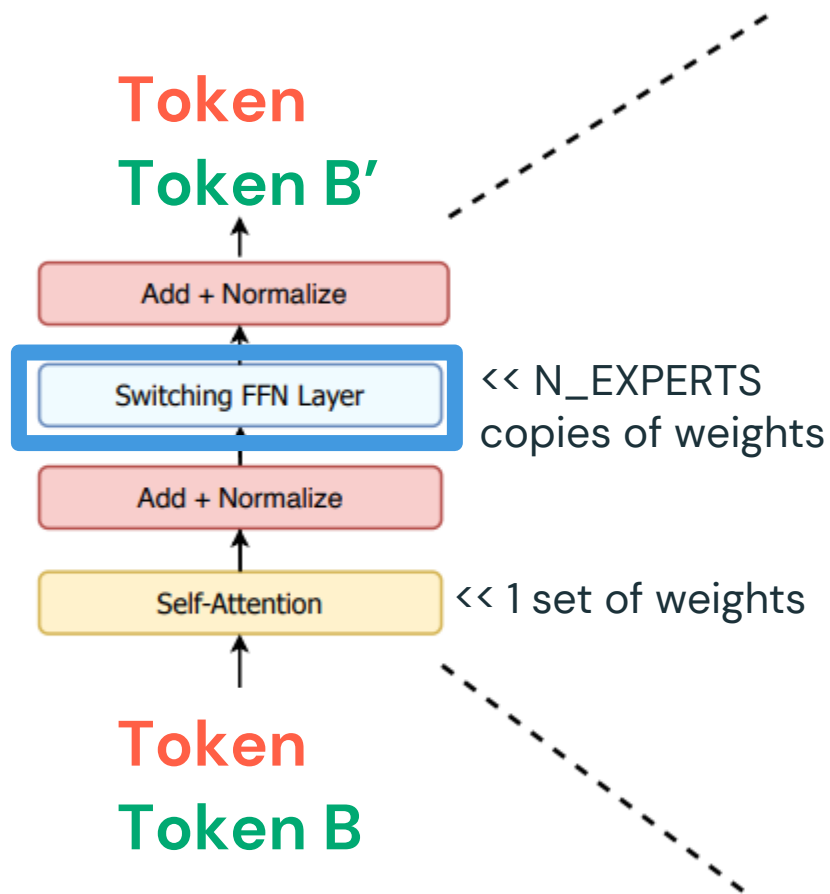
Quality of a bigger model, speed of a smaller one.

Mixture-of-Experts: Implementation Details



BEFORE: The Transformer Block processes each Token with the same **single** set of weights

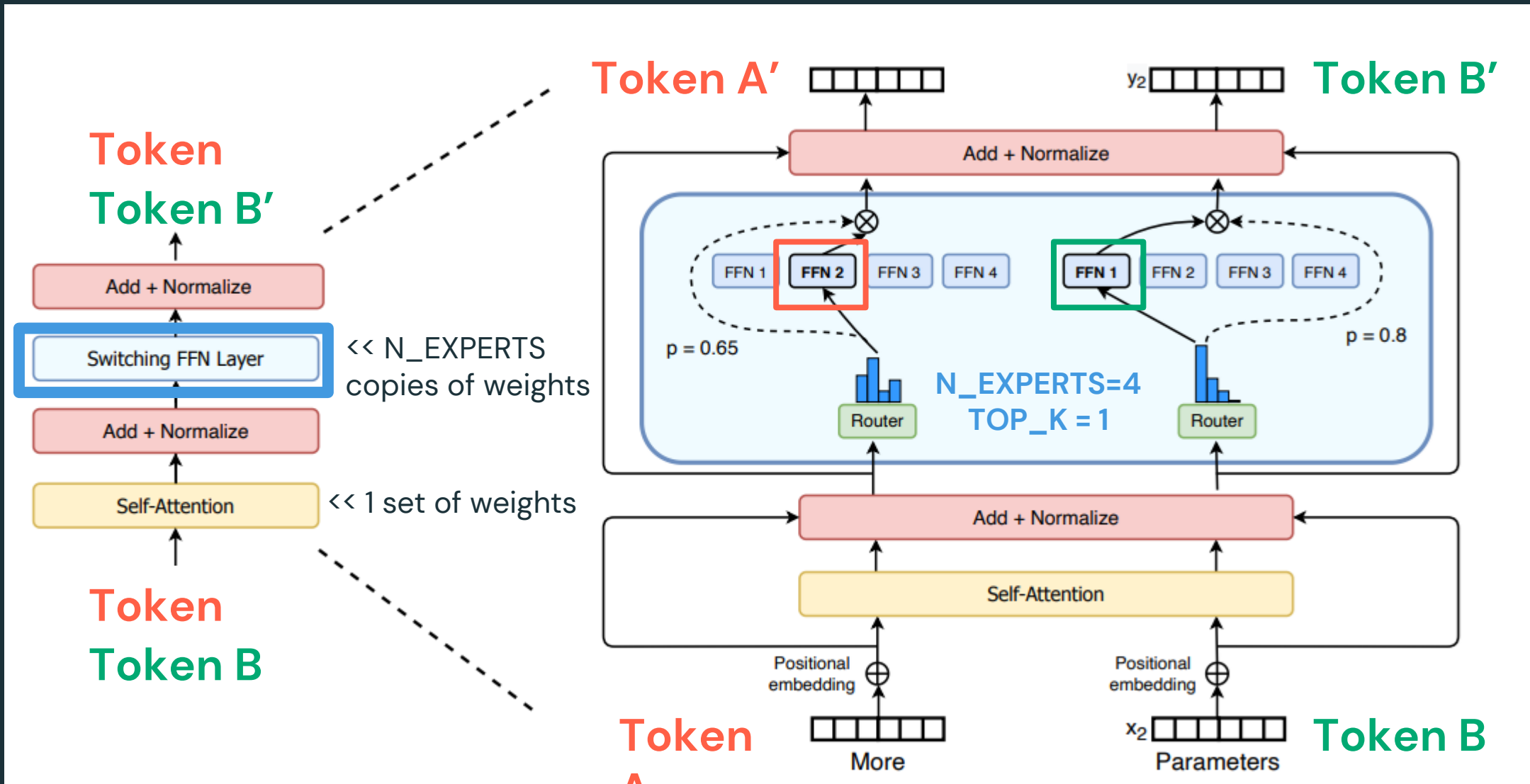
Mixture-of-Experts: Implementation Details



~~BEFORE: The Transformer Block processes each Token with the same **single** set of weights~~

AFTER: create multiple copies of **the FFN layer**, learn the weights independently, and route each Token to the best copy

Mixture-of-Experts: Implementation Details



The Value of Mixture-of-Experts

Arch.	Active Params	Relative FLOPs	Gauntlet Score
Llama2-13B	13B	1.7x	43.8%

The Value of Mixture-of-Experts

Arch.	Active Params	Relative FLOPs	Gauntlet Score
Llama2-13B	13B	1.7x	43.8%
DBRX Small	6.6B	1x	45.5%

The Value of Mixture-of-Experts

Arch.	Active Params	Relative FLOPs	Gauntlet Score
Llama2-13B	13B	1.7x	43.8%
DBRX Small	6.6B	1x	45.5%

Our mixture-of-experts training recipe scored higher, used 1.7x less training compute, and behaves like a model 2x smaller at inference time.

Do the math

How long will it take to train?

How long will it take to train my model?

How Long to Train
7B Param Model
Chinchilla Tokens
64 H100s

Cheatsheet

FLOPs = 6 x Parameters x Tokens

Tokens = 20 x Parameters (Chinchilla)

A100 = 312 TFLOP/sec = $3.12e14$

FLOP/sec

Data = $20 \times 7e9 = 1.8e11$

FLOPs = $6 \times 7e9 \times 1.8e11 = 5.88e21$

Cluster FLOP/sec = $3.12e14 \times 64 = 2e16$

Time = FLOPs / Cluster FLOP/sec = $5.88e21 / 2e16 = 3.4$ days

How long will it take to train my model?

How Long to Train
7B Param Model
Chinchilla Tokens
64 H100s

Cheatsheet

$\text{FLOPs} = 6 \times \text{Parameters} \times \text{Tokens}$

$\text{Tokens} = 20 \times \text{Parameters (Chinchilla)}$

$\text{A100} = 312 \text{ TFLOP/sec} = 3.12e14$

FLOP/sec

There's something missing here!

How long will it take to train my model?

How Long to Train
7B Param Model
Chinchilla Tokens
64 H100s

Cheatsheet

$\text{FLOPs} = 6 \times \text{Parameters} \times \text{Tokens}$

$\text{Tokens} = 20 \times \text{Parameters (Chinchilla)}$

$\text{A100} = 312 \text{ TFLOP/sec} = 3.12e14$

FLOP/sec

There's something missing here!

How long will it take to train my model?

How Long to Train
7B Param Model
Chinchilla Tokens
64 H100s

Cheatsheet

$\text{FLOPs} = 6 \times \text{Parameters} \times \text{Tokens}$

$\text{Tokens} = 20 \times \text{Parameters (Chinchilla)}$

$\text{A100} = 312 \text{ TFLOP/sec} = 3.12e14$

FLOP/sec

You won't fully utilize your GPU.

There are other bottlenecks in the system.

This is the theoretical peak. You will get power limited.

How long will it take to train my model?

How Long to Train
7B Param Model
Chinchilla Tokens
64 H100s

Cheatsheet

$\text{FLOPs} = 6 \times \text{Parameters} \times \text{Tokens}$

$\text{Tokens} = 20 \times \text{Parameters (Chinchilla)}$

$\text{A100} = 312 \text{ TFLOP/sec} = 3.12e14$

FLOP/sec

MFU = Model Flop Utilization

What fraction of the peak GPU FLOP/sec is your model getting?

Only counts $6 \times \text{Parameters} \times \text{Tokens}$, not recomputation.

For this configuration, **50.7% MFU.**

How long will it take to train my model?

How Long to Train
7B Param Model
Chinchilla Tokens
64 H100s

Cheatsheet

FLOPs = 6 x Parameters x Tokens

Tokens = 20 x Parameters (Chinchilla)

A100 = 312 TFLOP/sec = $3.12e14$

FLOP/sec

Data = $20 \times 7e9 = 1.8e11$

FLOPs = $6 \times 7e9 \times 1.8e11 = 5.88e21$

Cluster FLOP/sec = $3.12e14 \times 64 = 2e16$

Time = FLOPs / Cluster FLOP/sec = $5.88e21 / 2e16 = 3.4$ days

How long will it take to train my model?

How Long to Train

7B Param Model

Chinchilla Tokens

64 H100s

Data = $20 \times 7e9 = 1.8e11$

FLOPs = $6 \times 7e9 \times 1.8e11 = 5.88e21$

Cluster FLOP/sec = $3.12e14 \times 64 \times 50.7\% = 1.01e16$

Time = FLOPs / Cluster FLOP/sec = $5.88e21 / 1.01e16 = 6.7$

days

Cheatsheet

FLOPs = $6 \times \text{Parameters} \times \text{Tokens}$

Tokens = $20 \times \text{Parameters}$ (Chinchilla)

A100 = 312 TFLOP/sec = $3.12e14$

FLOP/sec

Nuts and Bolts

The technologies we built on.

We share our training code open source

Composer. Training library built for scalability.

Streaming. Stream efficiently from object stores.

LLM Foundry. Highly efficient and scalable training and fine-tuning code for popular LLMs.

MegaBlocks. Mixture-of-Experts implementation.

We share our training code open source

Composer. github.com/mosaicml/composer

Streaming. github.com/mosaicml/streaming

LLM Foundry. github.com/mosaicml/llm-foundry

MegaBlocks. github.com/databricks/megablocks

We partner with the community



MegaBlocks



Pytorch FSDP



vLLM (Inference)



TensorRT-LLM
(Inference)



EleutherAI LLM
evaluation harness



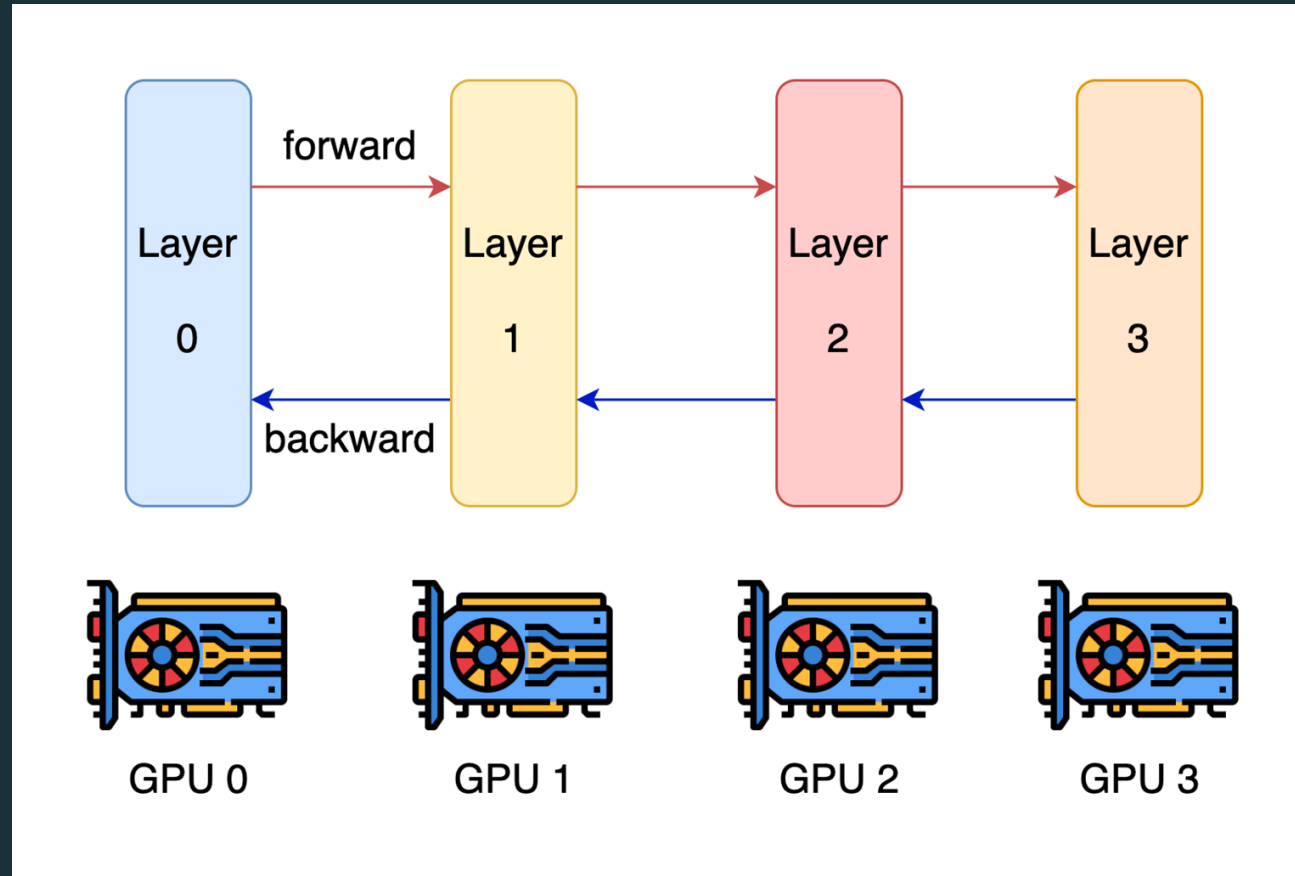
Amazing collaborators

Scaling Up

Problem 1: DBRX is too big to fit on one GPU.

Problem 2: To finish our training run in a reasonable amount of time, we need to train on 3072 GPUs.

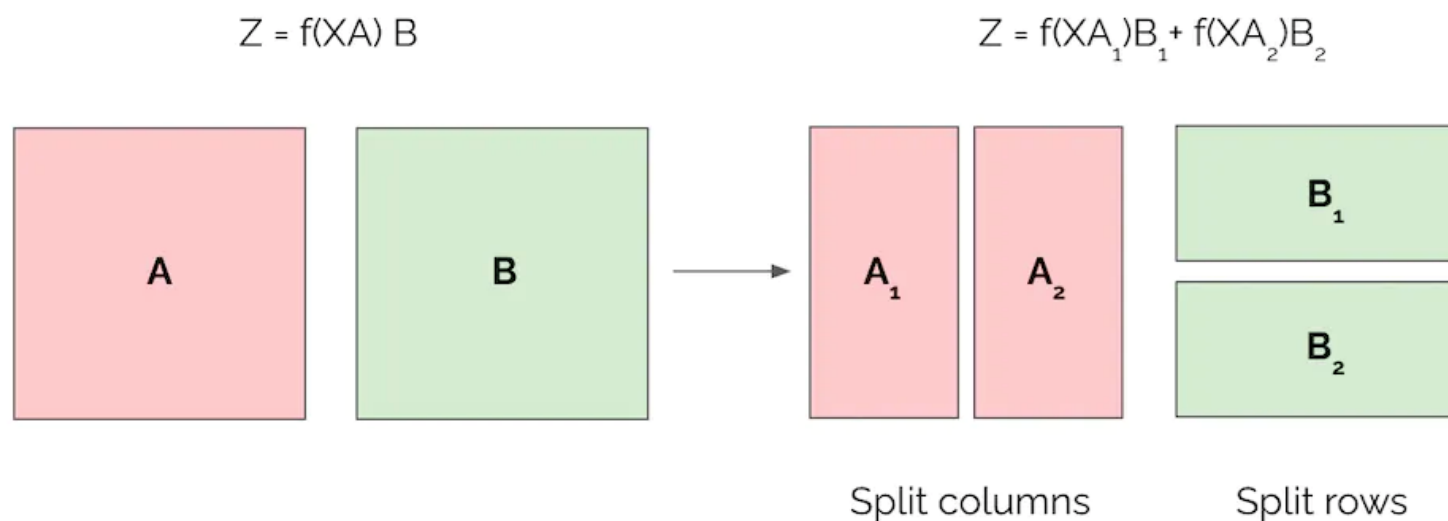
Scaling Up: Pipeline Parallelism?



Colossal AI

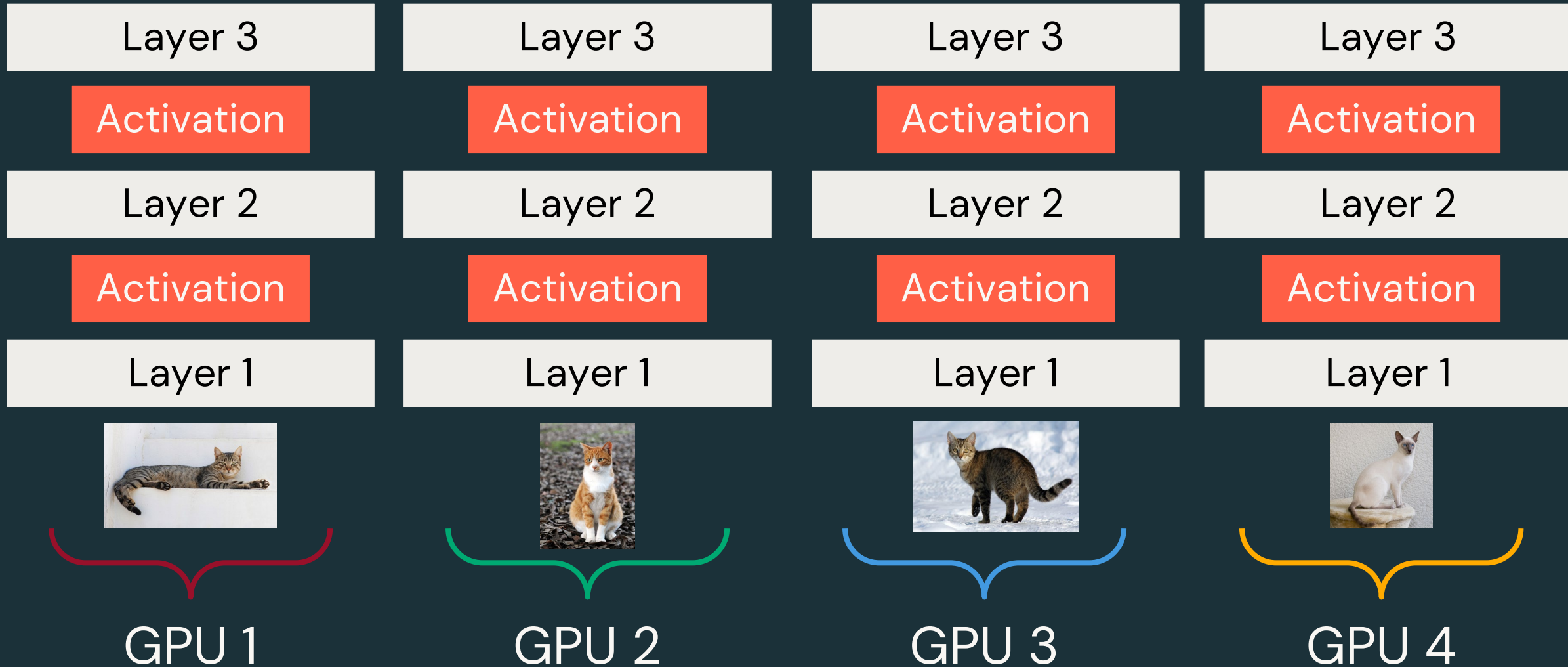
Scaling Up: Tensor Parallelism?

Splitting a two-layer MLP across two devices

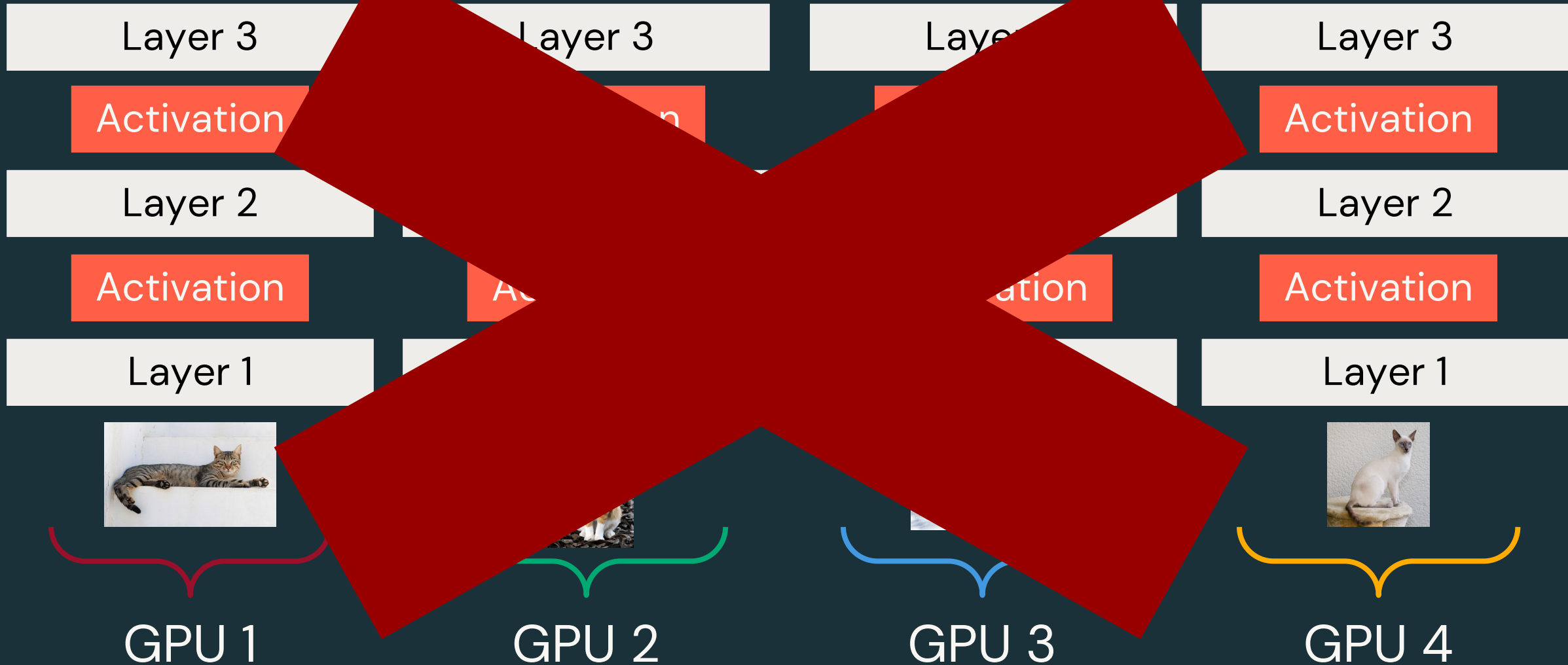


Misha Laskin

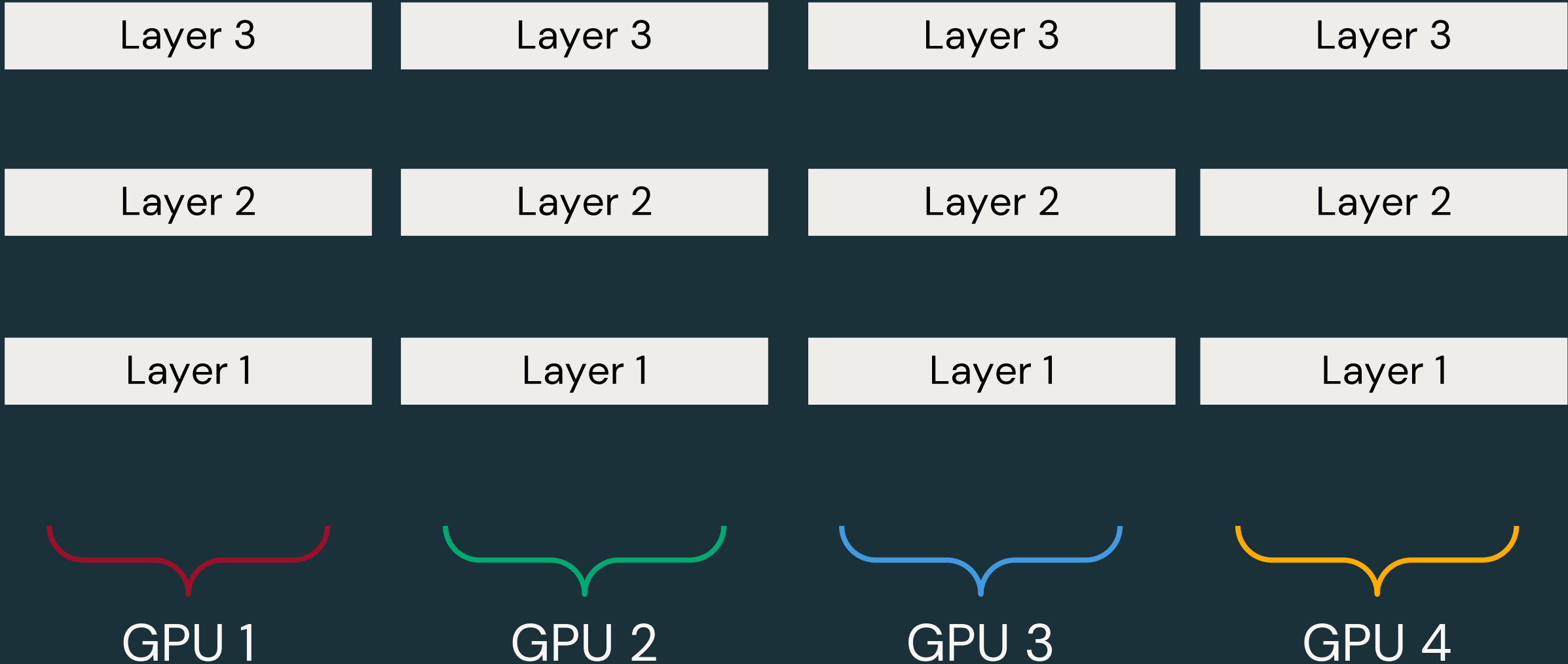
Scaling Up: Data Parallelism



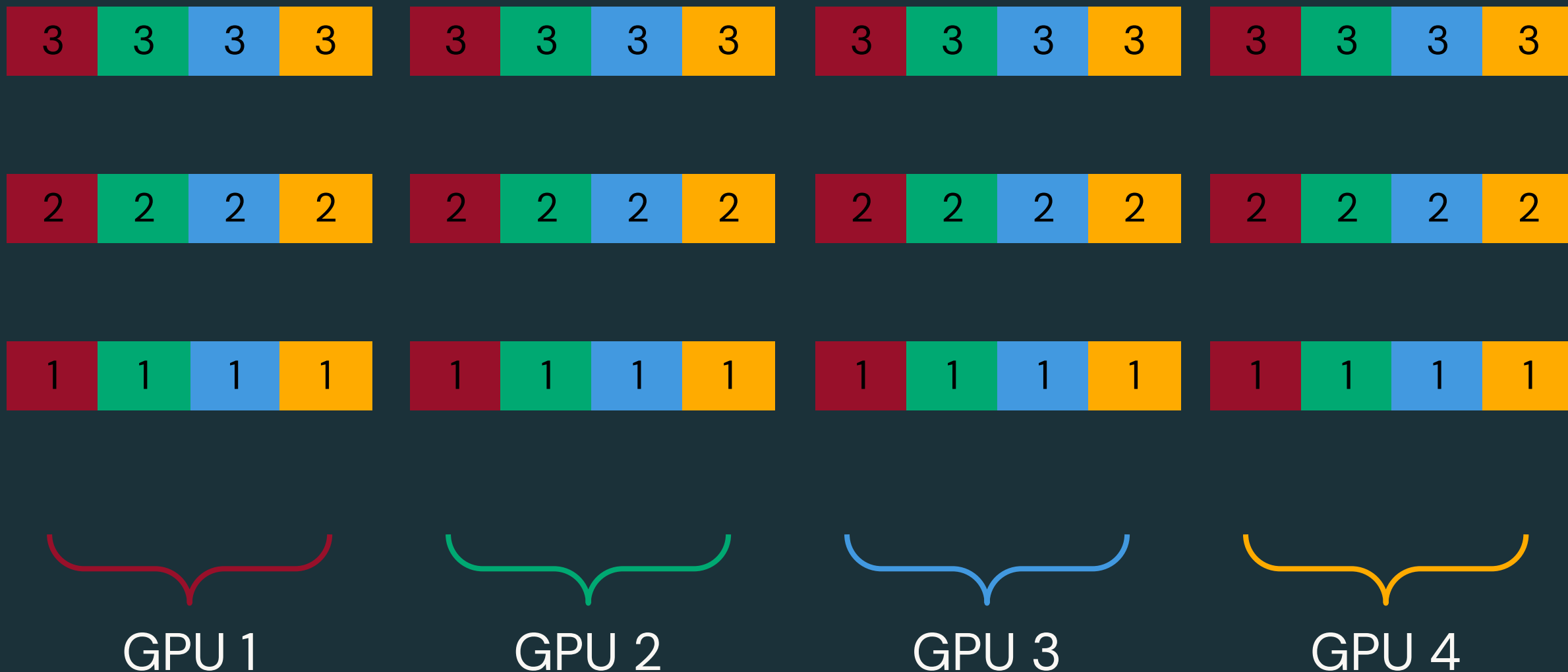
Scaling Up: Data Parallelism



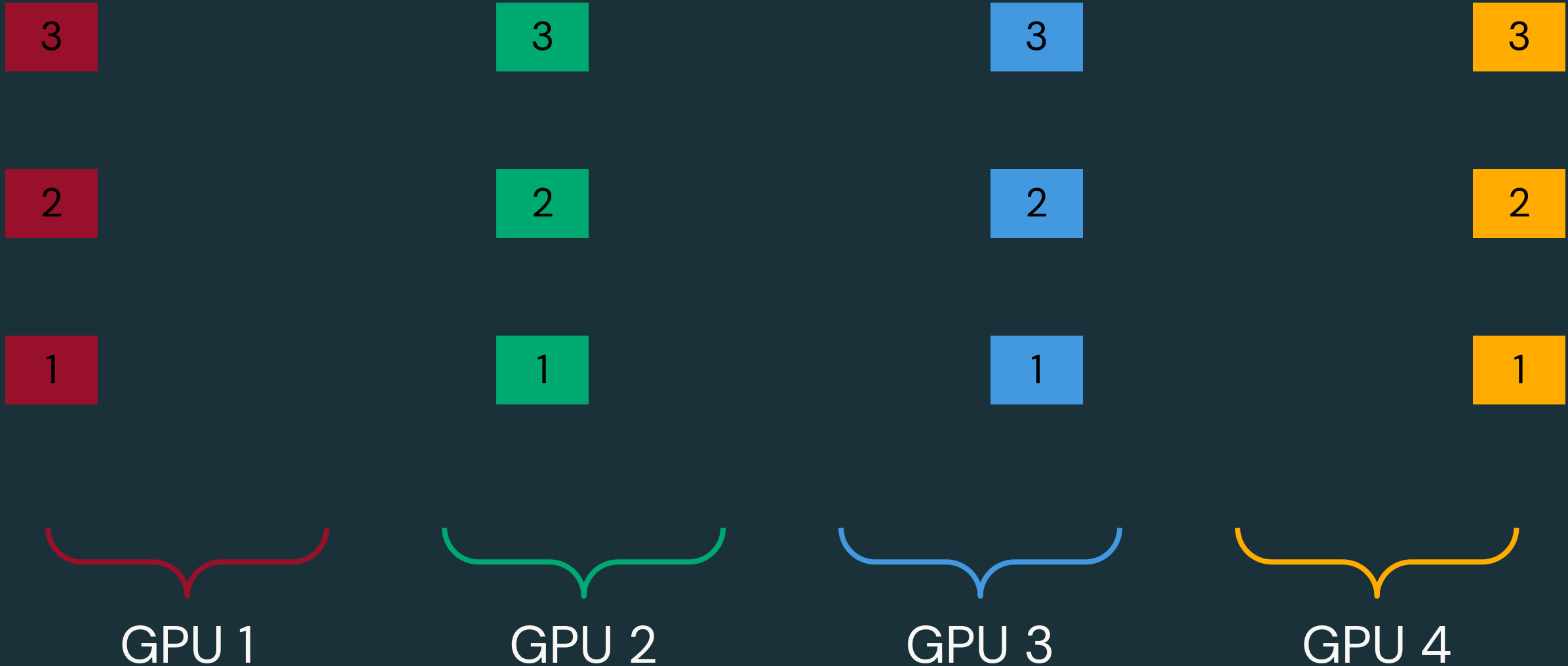
Scaling Up: Data Parallelism + FSDP



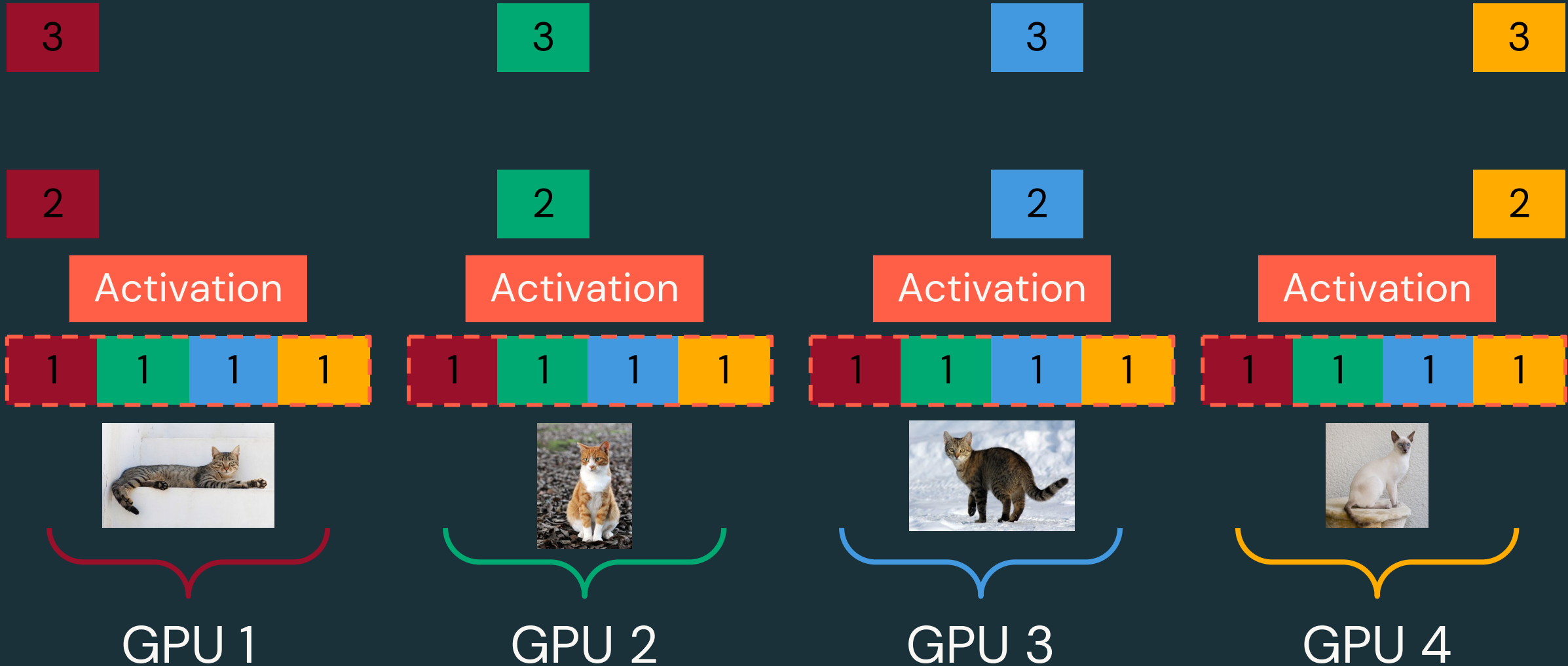
Scaling Up: Data Parallelism + FSDP



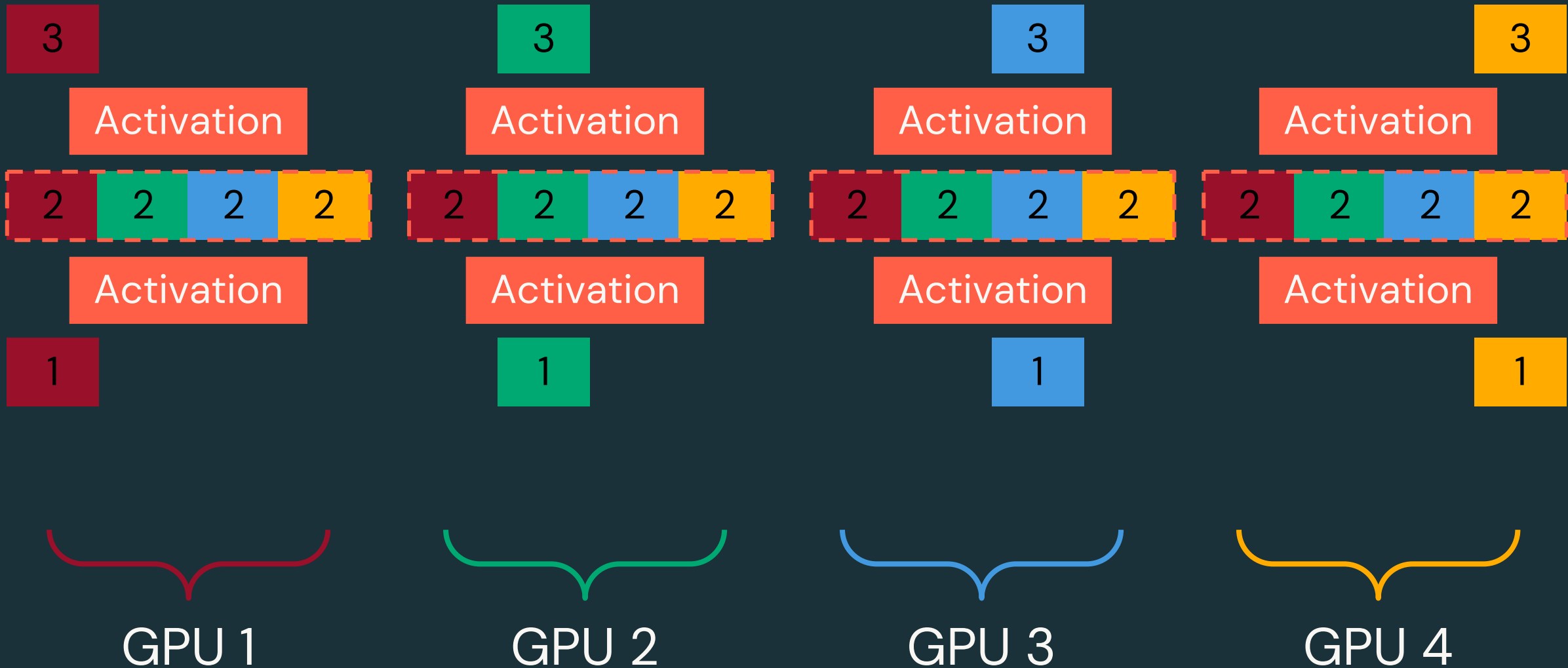
Scaling Up: Data Parallelism + FSDP



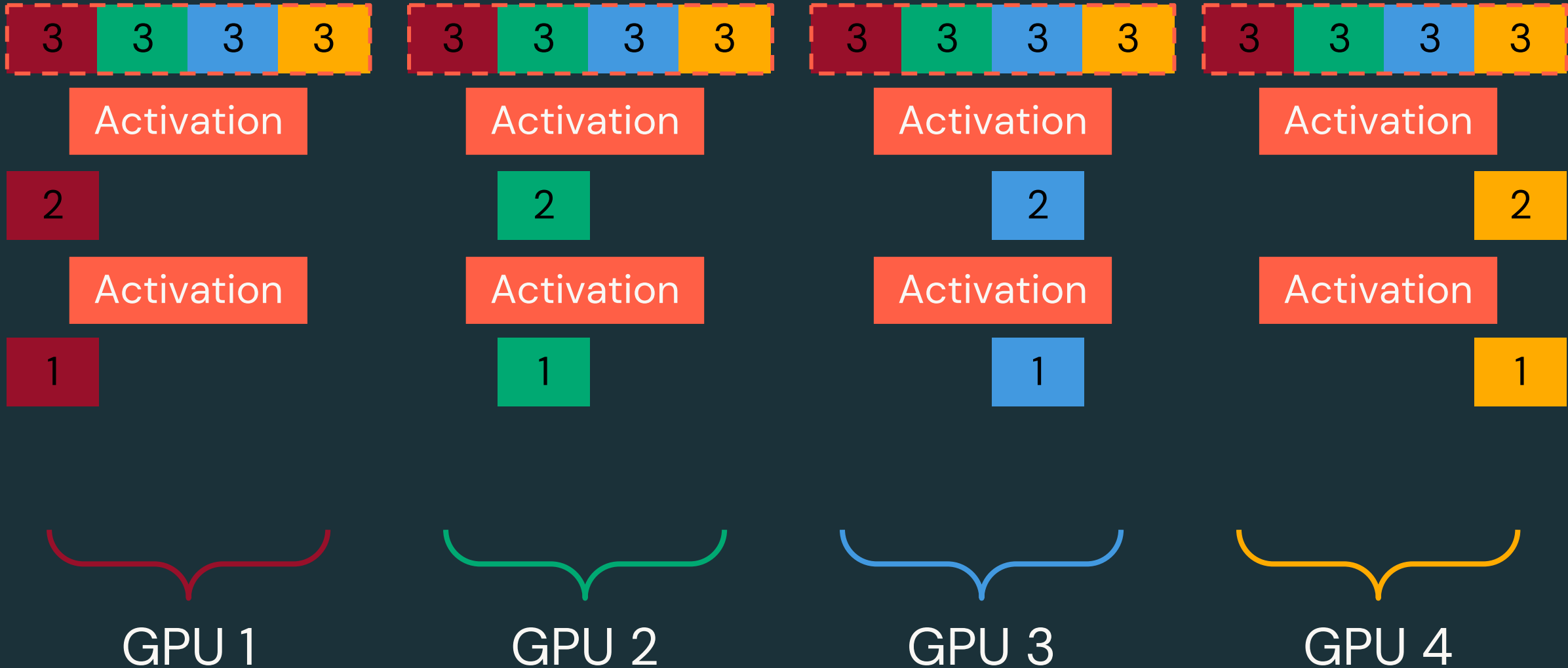
Scaling Up: Data Parallelism + FSDP



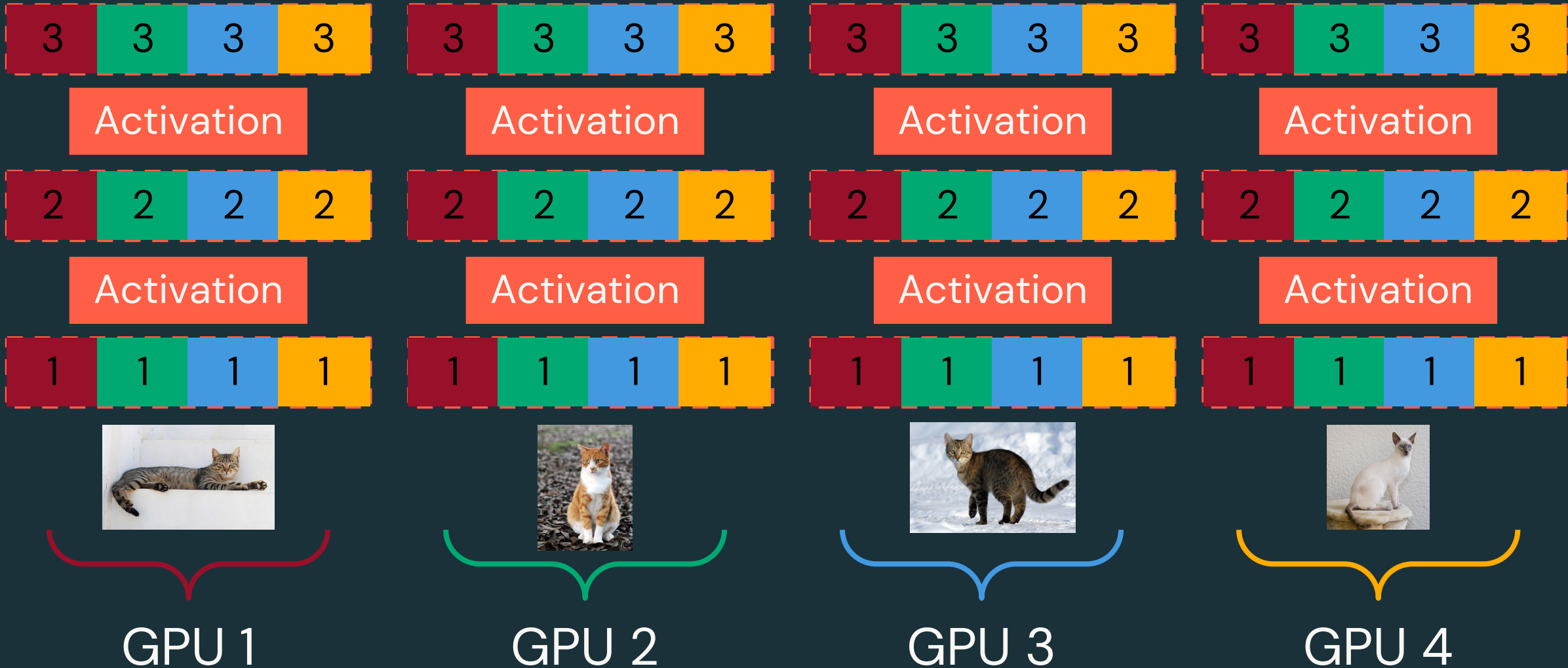
Scaling Up: Data Parallelism + FSDP



Scaling Up: Data Parallelism + FSDP



Scaling Up: Data Parallelism + FSDP



Scaling Up: Data Parallelism + FSDP

Zero Redundancy Optimizer



DeepSpeed

Stage 1: The optimizer states (e.g., for [Adam optimizer](#), 32-bit weights, and the first, and second moment estimates) are partitioned across the processes, so that each process updates only its partition.

Stage 2: The reduced 32-bit gradients for updating the model weights are also partitioned such that each process retains only the gradients corresponding to its portion of the optimizer states.

Stage 3: The 16-bit model parameters are partitioned across the processes. ZeRO-3 will automatically collect and partition them during the forward and backward passes.

Tutorials > Advanced Model Training with Fully Sharded Data Parallel (FSDP)

ADVANCED MODEL TRAINING WITH FULLY SHARDED DATA PARALLEL (FSDP)

Author: [Hamid Shojanazeri](#), [Less Wright](#), [Rohan Varma](#), [Yanli Zhao](#)

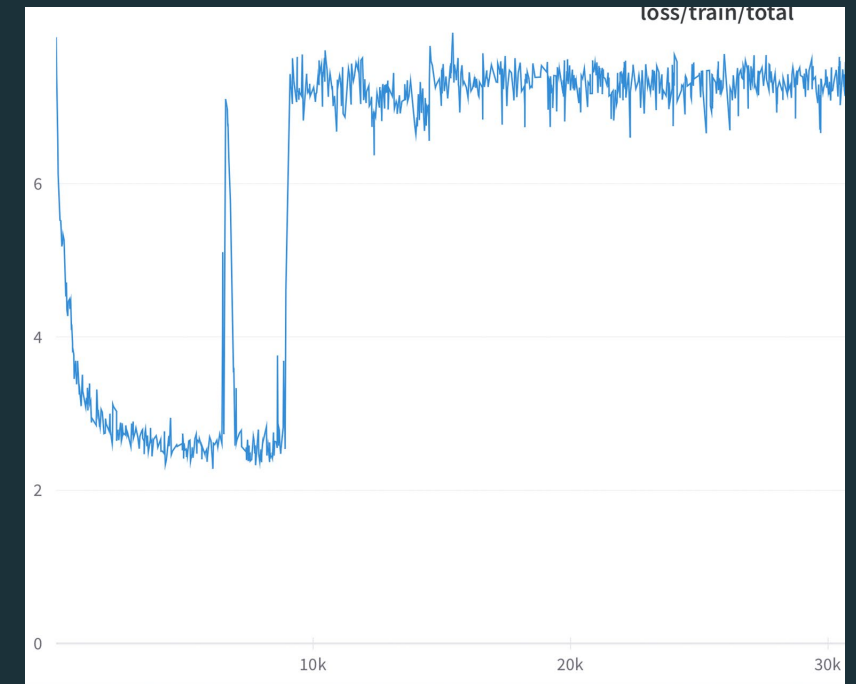
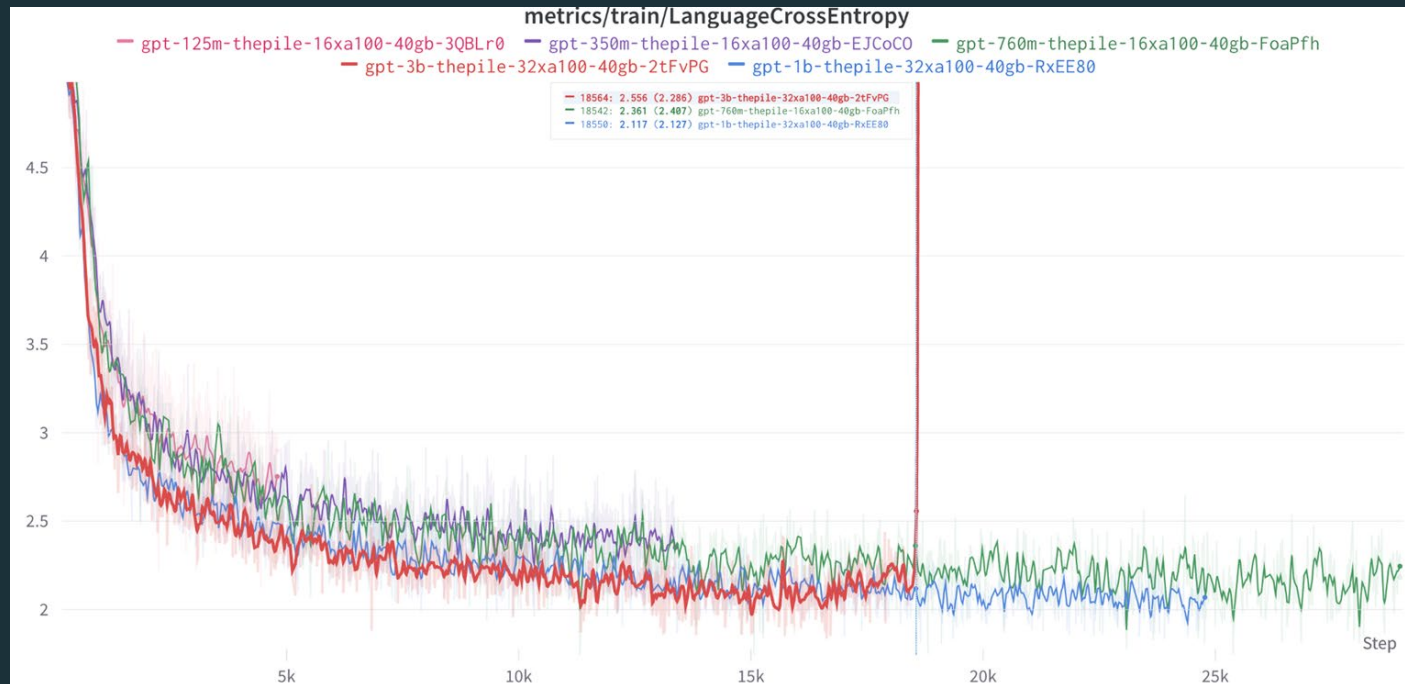
This tutorial introduces more advanced features of Fully Sharded Data Parallel (FSDP) as part of the PyTorch 1.12 release. To get familiar with FSDP, please refer to the [FSDP getting started tutorial](#).

In this tutorial, we fine-tune a HuggingFace (HF) T5 model with FSDP for text summarization as a working example.

And then you train...

**and all hell breaks loose
and Databricks is here to help!**

Problem: Training Instability and Loss Spikes



Our solution: Special hyperparameters and optimization strategies that mitigate loss spikes

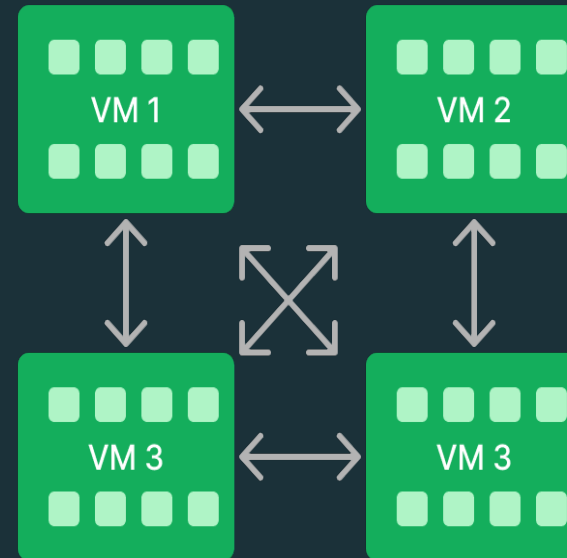
Problem: Hardware Failures

- Roughly once every 1000 H100-days
- GPUs, Switches, Communication Libraries (NCCL)


Normally: $O(N)$ things can go wrong




Training : $O(N^2)$ things can go wrong



Problem: Hardware Failures

 **Jonathan Frankle** 🇺🇸 12 days ago
Today has been a bad day for GPUs. Please press **F** to pay your respects

F 18 🤔

 **Node-Health-Bot** APP 6:54 PM

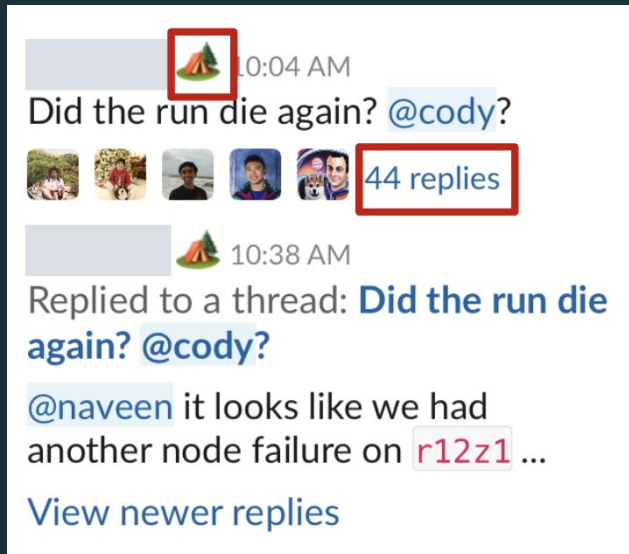
This little piggy (🐷 node `inst-pwx1x-r7z2-workers`) is 💀 **DEAD** 💀 on cluster `r7z2`

Priority	Type
<i>Critical</i>	<i>Node Died</i>
Reason	Message
<i>GPU is lost</i>	<i>GPU at index 2 was detected to be not ready: GPU is lost</i>

Our solution: Automatic failure detection and blazingly fast job restart times.

Problem: Hardware Failures

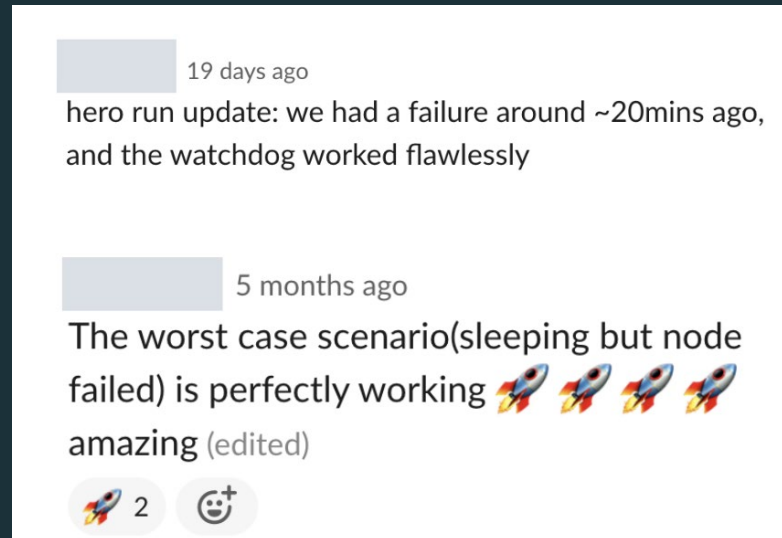
BEFORE



10:04 AM
Did the run die again? @cody?
44 replies

10:38 AM
Replied to a thread: Did the run die again? @cody?
@naveen it looks like we had another node failure on r12z1 ...
View newer replies

AFTER



19 days ago
hero run update: we had a failure around ~20mins ago, and the watchdog worked flawlessly

5 months ago
The worst case scenario(sleeping but node failed) is perfectly working 🚀🚀🚀🚀
amazing (edited)
2

Our solution: Automatic failure detection and blazingly fast job restart times.

Friendly Advice

Friendly Advice

Start small and work your way up.

Don't trust what you read in the literature.
Test everything for yourself.

Don't trust intuition, received wisdom, or a rumor.
Test everything for yourself.

Do the math.

Friendly Advice

Start small and work your way up.

Don't trust what you read in the literature.

Let Databricks be your research team.

Don't trust intuition, received wisdom, or a rumor.

Let Databricks be your research team.

Do the math together. We have your back.

DBRX: Training Modern LLMs From Scratch

Abhinav Venigalla & Jonathan Frankle
NLP Architect & Chief AI Scientist

