Natural Language Processing: Efficiency

HSE Faculty of Computer Science

Machine Learning and Data-Intensive Systems

Murat Khazhgeriev

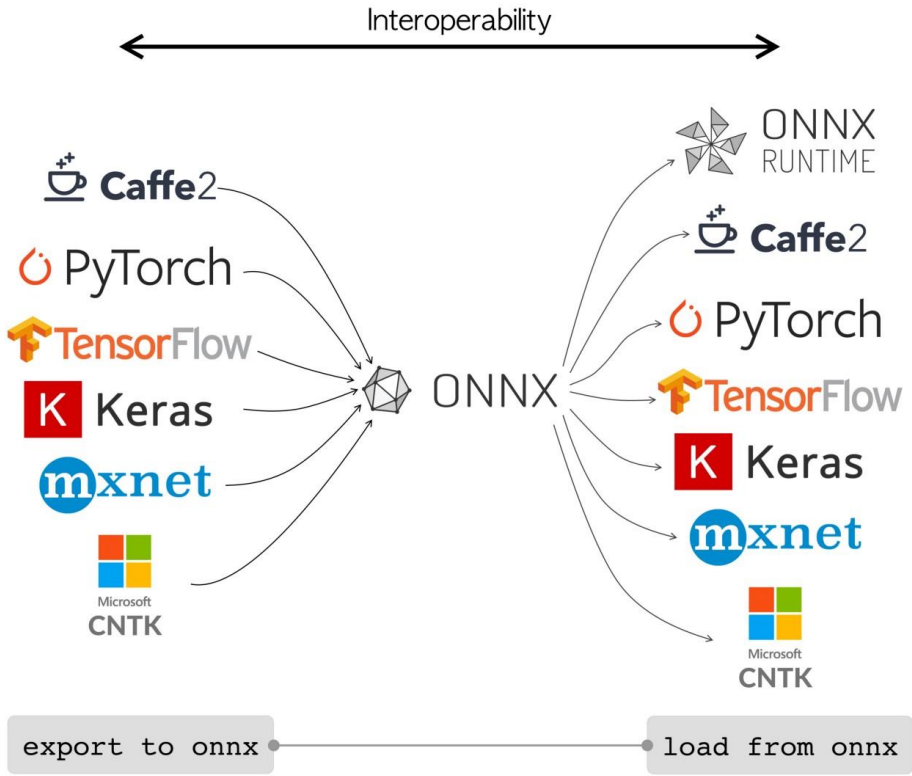
Table of Content

- Just finished training, now what?
- Engineering tricks
- Architectural tricks
- Going distributed

Table of Content

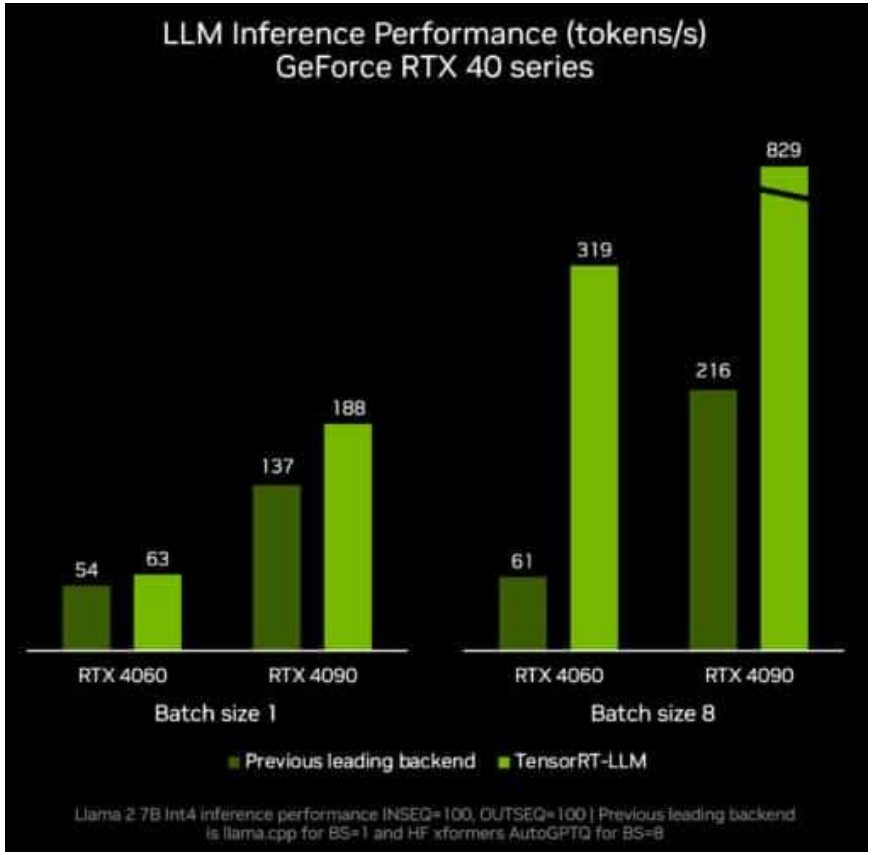
- **Just finished training, now what?**
- Engineering tricks
- Architectural tricks
- Going distributed

Move to engine-agnostic format ONNX



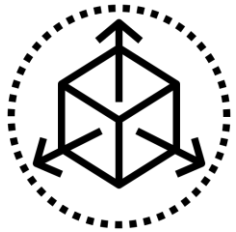
Source: https://onnx.ai/

TensorRT if you have Nvidia GPUs



Source: https://docs.nvidia.com/tensorrt-llm/index.html

Know your enemy



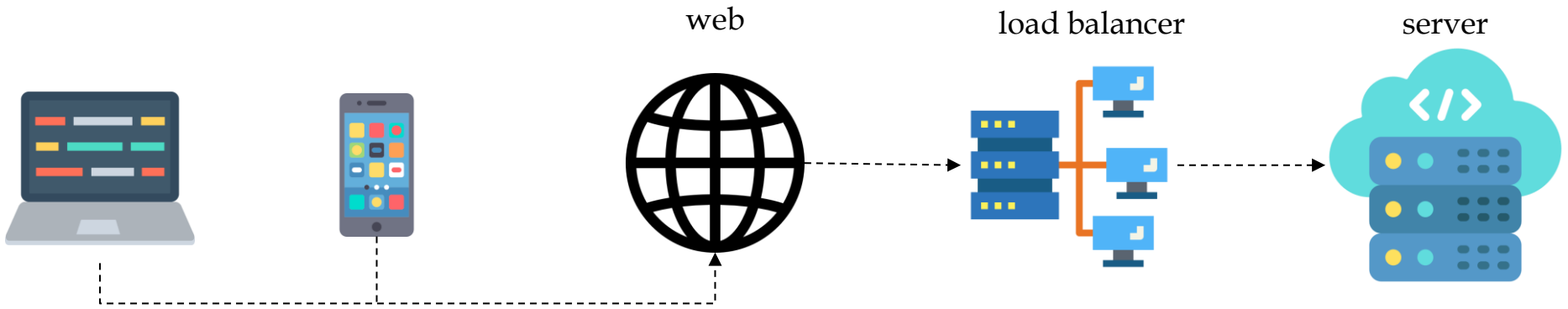
Model size
(do we fit in RAM?)

Throughput
(how many queries per second?)

Latency
(how much ms @ percentile till first token?)

Source: https://docs.nvidia.com/tensorrt-llm/index.html

Inference server



Throughput is often critical

Monitor latency @ different percentiles

Key links:

Prototyping: TorchServe

If you don't like Nvidia: vLLM

Optimized for Nvidia GPUs: Triton Inference Server

Mobile

Memory is often a **huge** problem 

Watch out for high latency 

Key links:

In browsers:	TensorFlow.js
Mobile Platform-agnostic	ExecuTorch
Apple:	CoreML
Android:	NNAPI

Workstation

Memory may be a problem, but often not

Watch out for latency

Key links:

1: LLaMA.CPP

2: OLLAMA

3: ExLLaMA v2

Table of Content

- Just finished training, now what?
- **Engineering tricks**
- Architectural tricks
- Going distributed

Machine numbers refresher

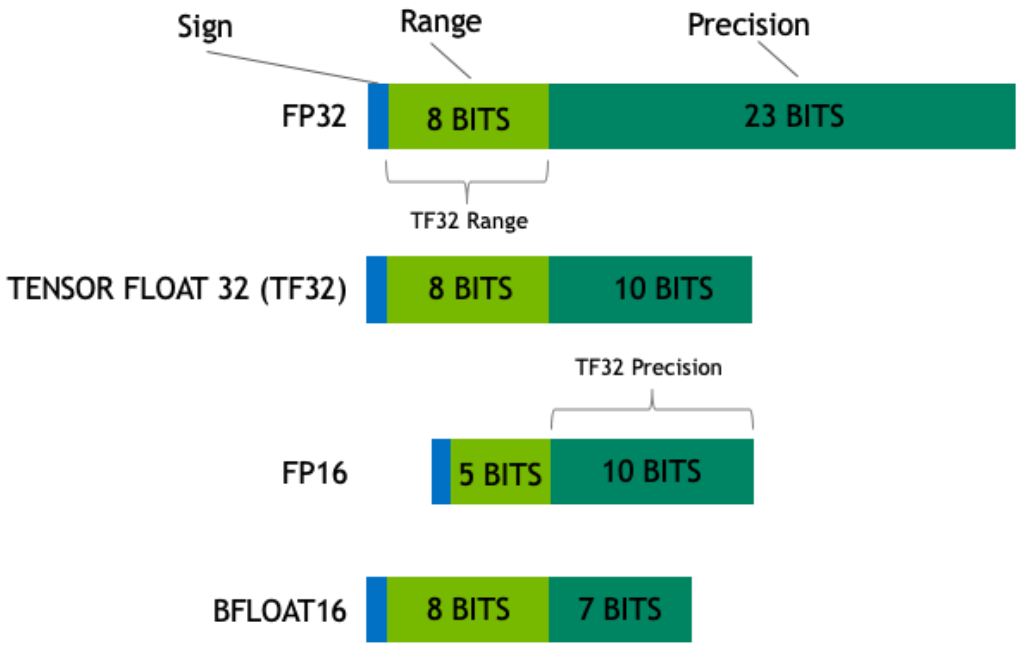
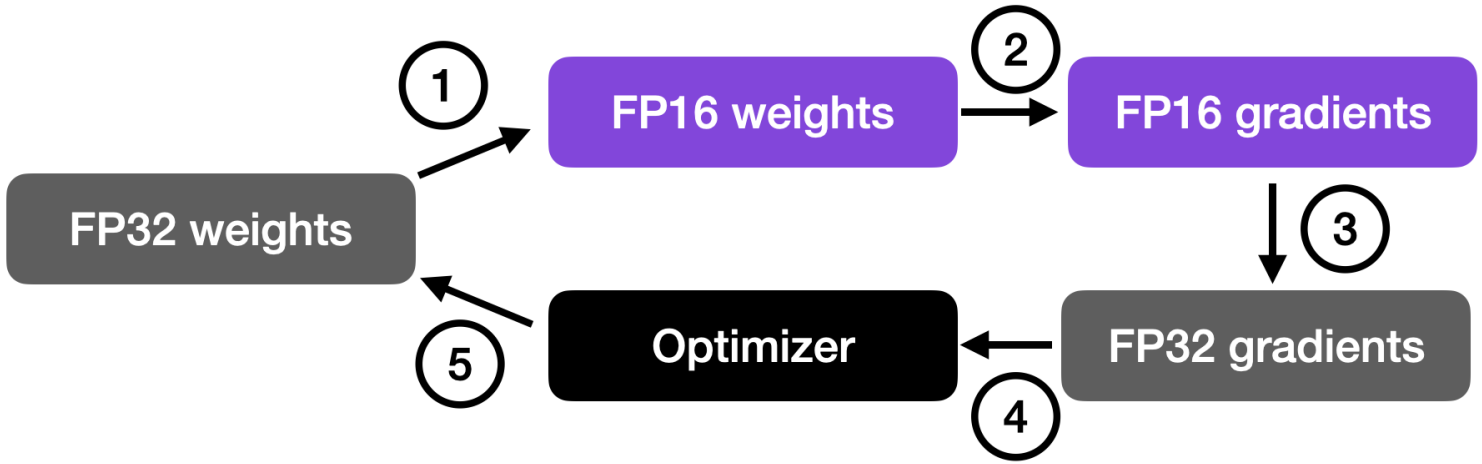


Image source: https://huggingface.co/blog/hf-bitsandbytes-integration

Mixed Precision



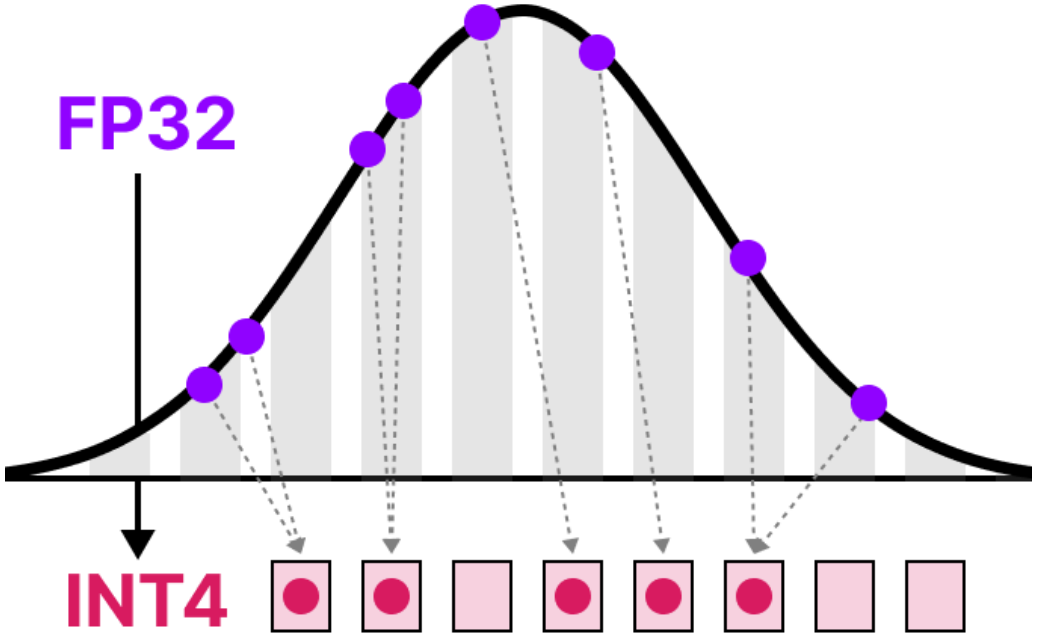
Key links:

1: Automatic Mixed Precision in Torch

2: What Every User Should Know About Mixed Precision Training in PyTorch

Image source: https://lightning.ai/pages/community/tutorial/accelerating-large-language-models-with-mixed-precision-techniques/

Quantization



Key links:

1: Quantization Recipe

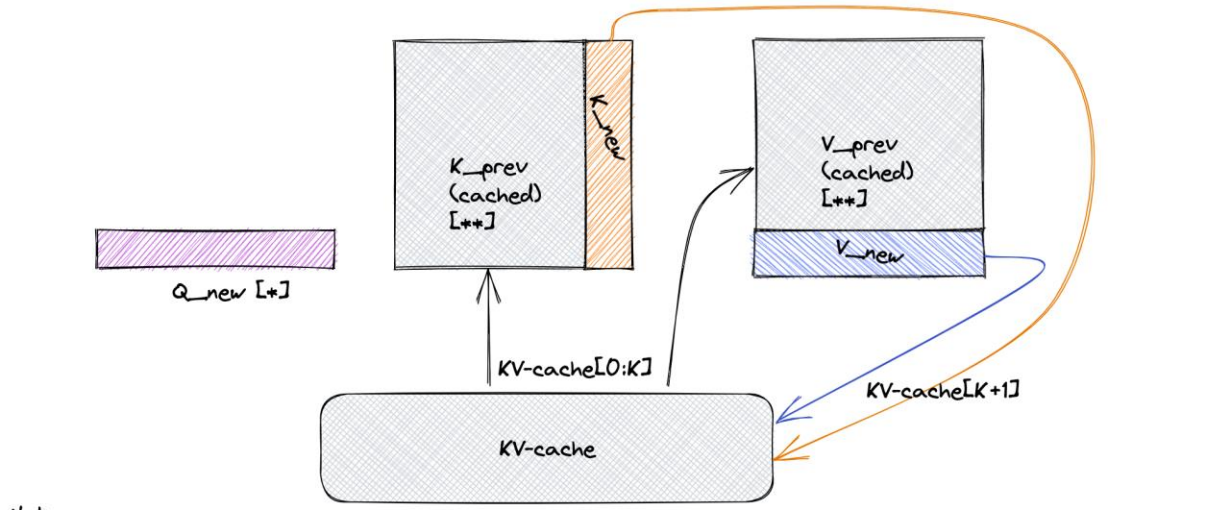
2: HF Optimum

3: Practical Quantization

4: HF bitsandbytes integration

Image source: https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-quantization

KV-cache

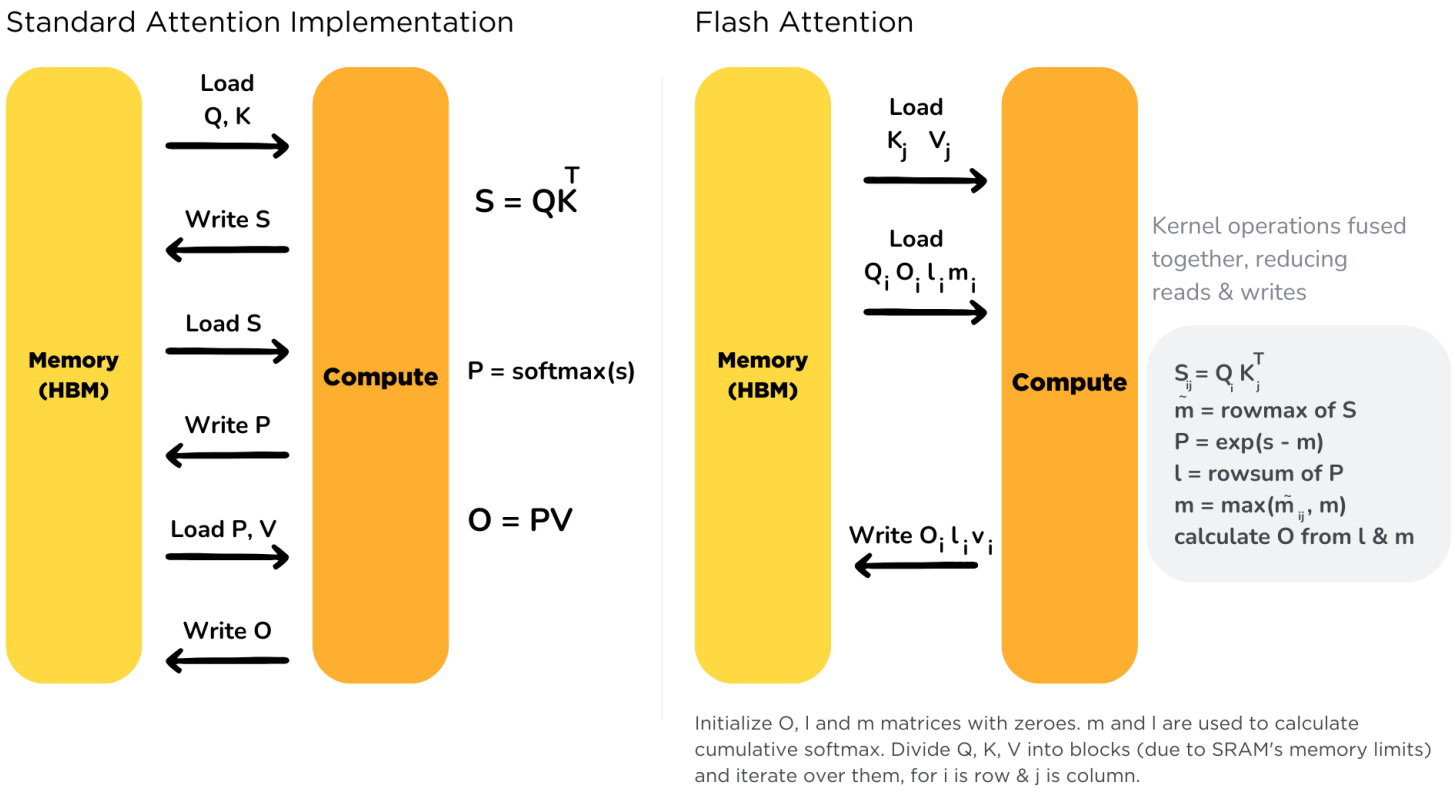


Key links:

1: [Key-Value Cache Quantization]

Image source:https://huggingface.co/blog/kv-cache-quantization

Flash, Grouped-query, Sparse, Paged Attention



Key links:

1: Grouped-query Attention
2: Flash Attention
3: Sparse Attention
4: Paged Attention

Image https://huggingface.co/docs/text-generation-inference/en/conceptual/flash_attention

Fuse optimizer's kernels into one

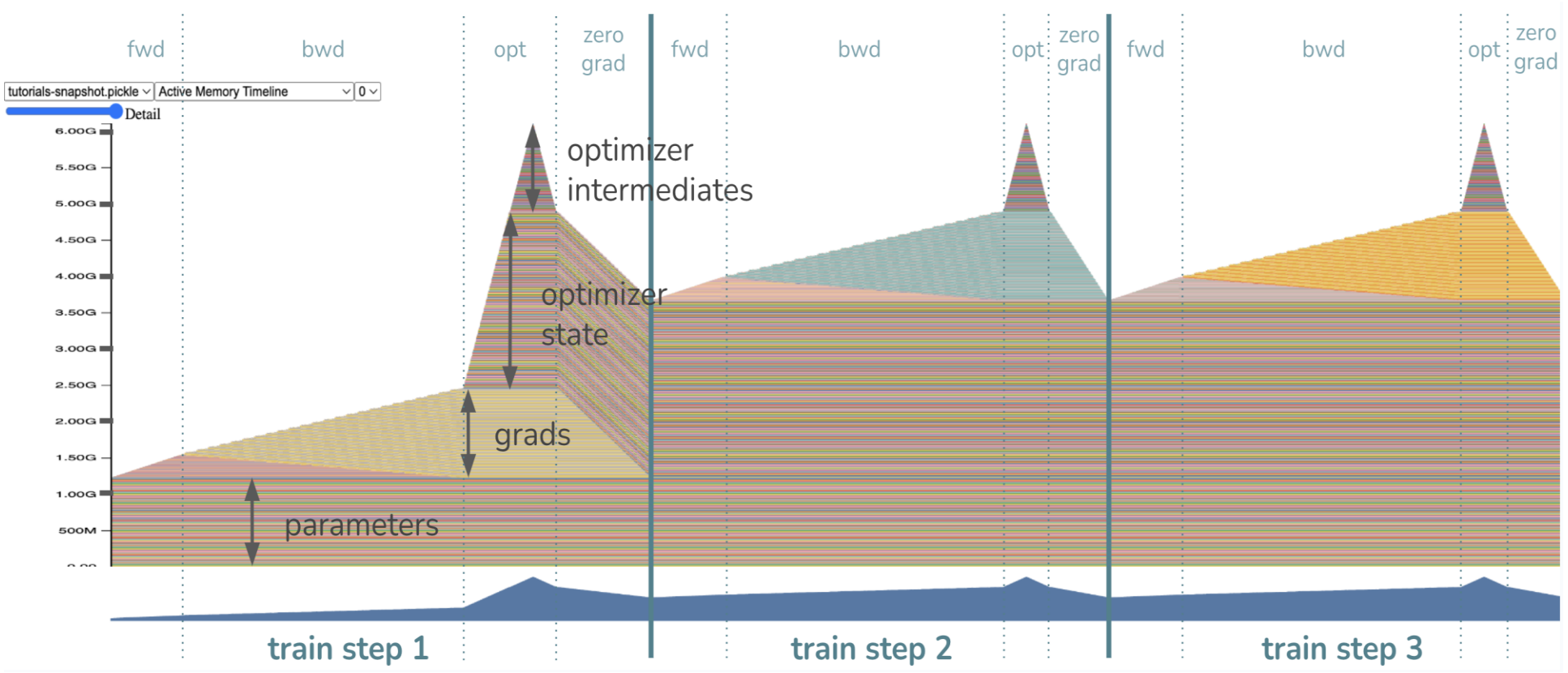
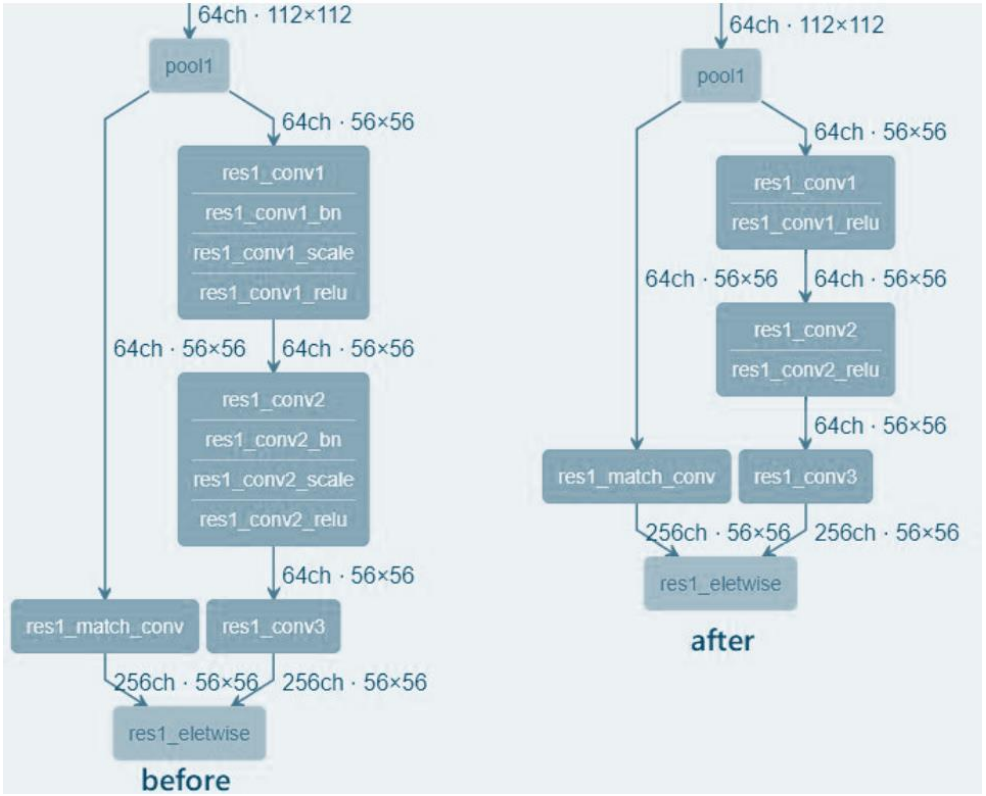


Image source: How to save memory by fusing the optimizer step into the backward pass

Fuse Torch modules

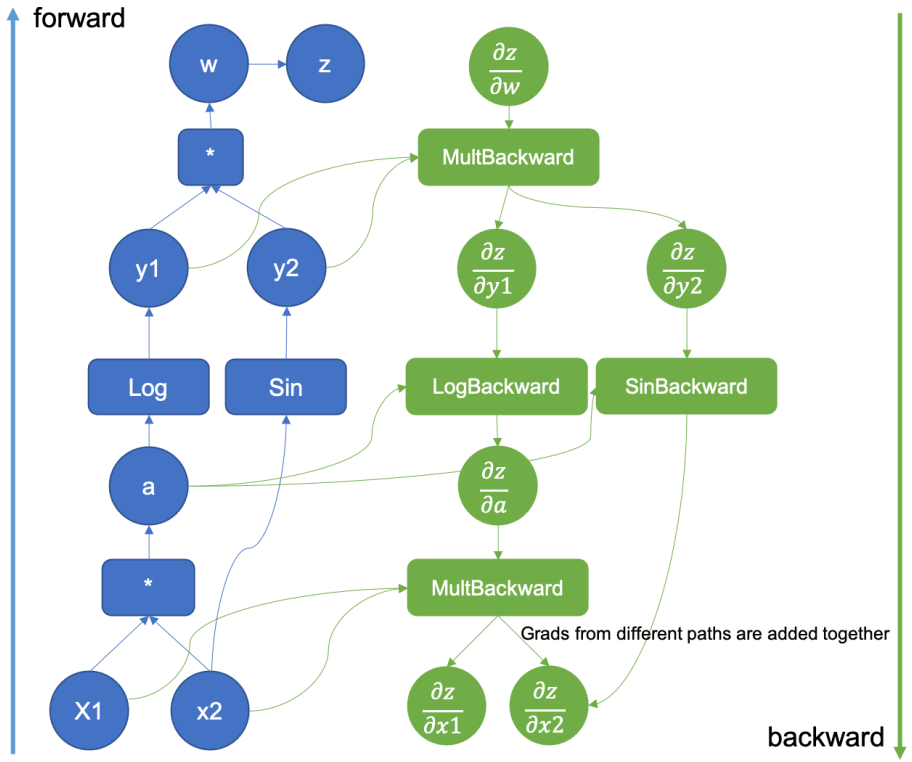


Key links:

1: Fuse Modules Recipe

Image source: -

Make Torch's Dynamic Computation Graph Static



Key links:

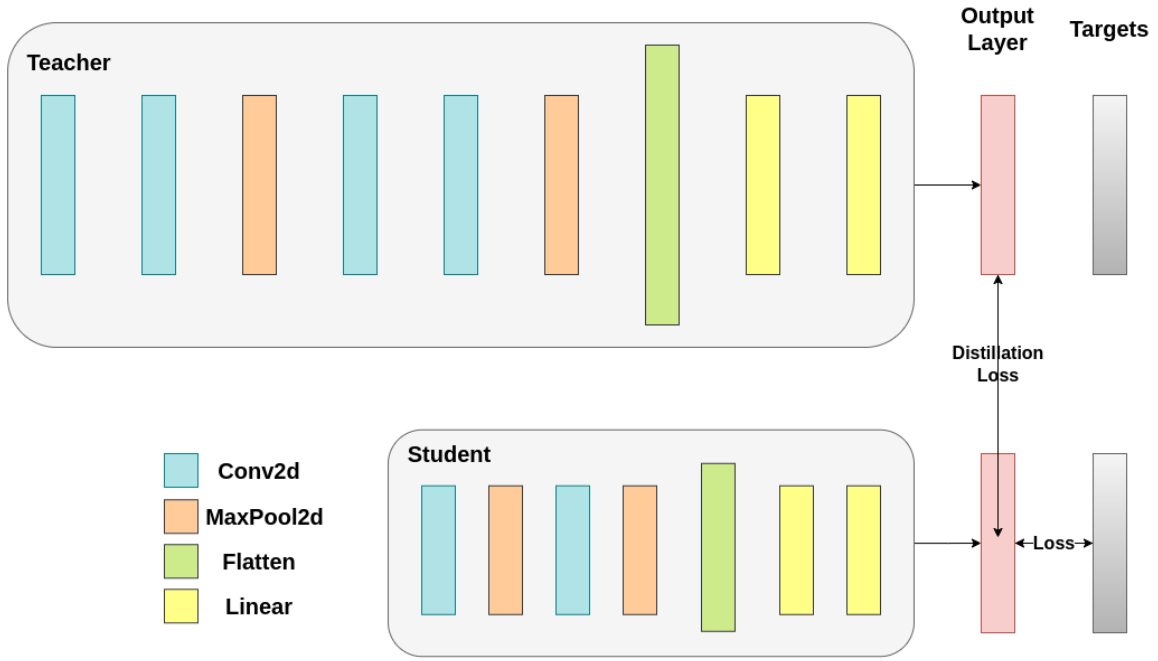
1: torch.compile

Image source: How is computation graph constructed?

Table of Content

- Just finished training, now what?
- Engineering tricks
- **Architectural tricks**
- Going distributed

Knowledge Distillation

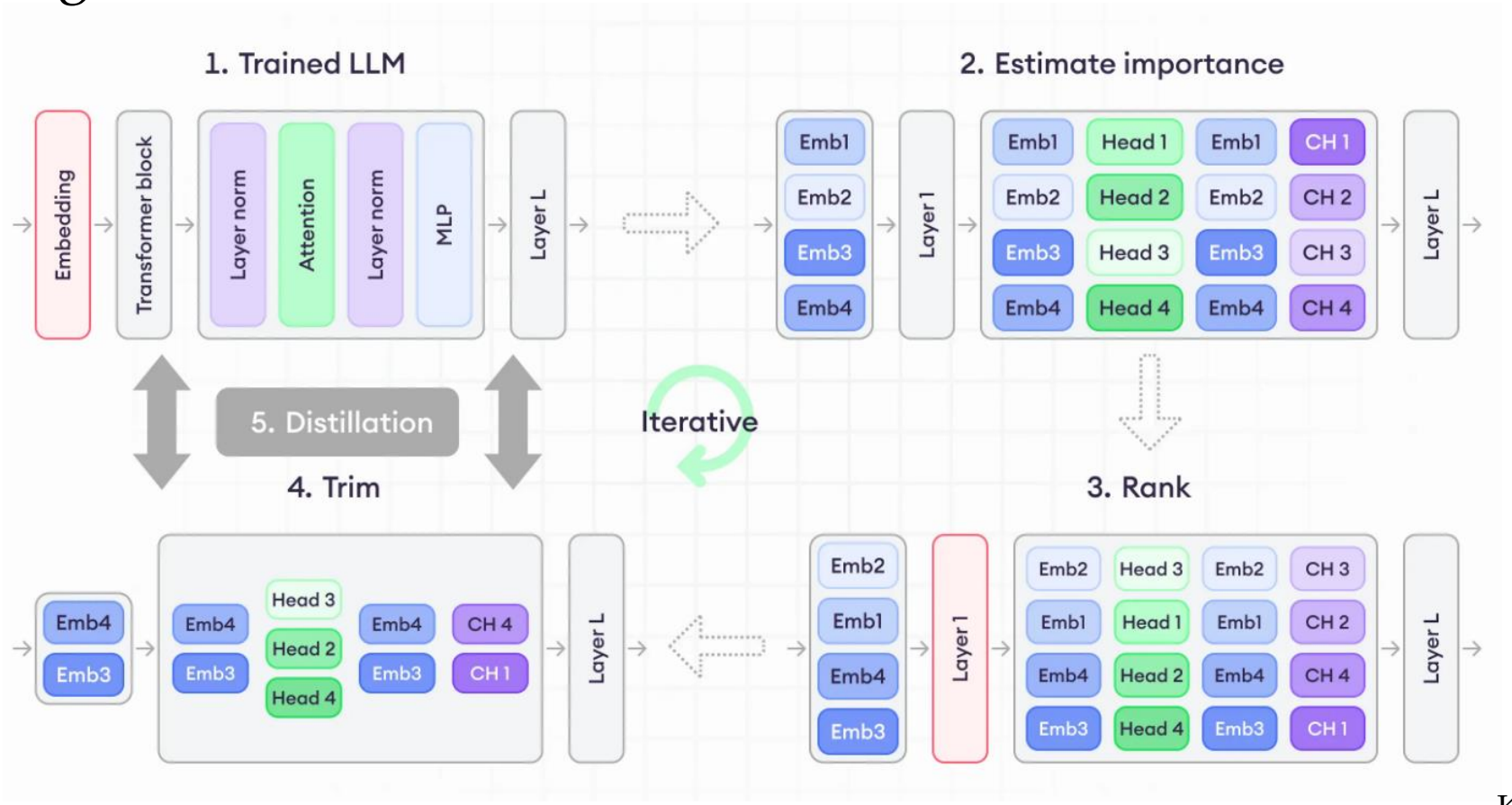


Key links:

1: Distillation Tutorial Torch

Image source: Distillation Tutorial Torch

Pruning

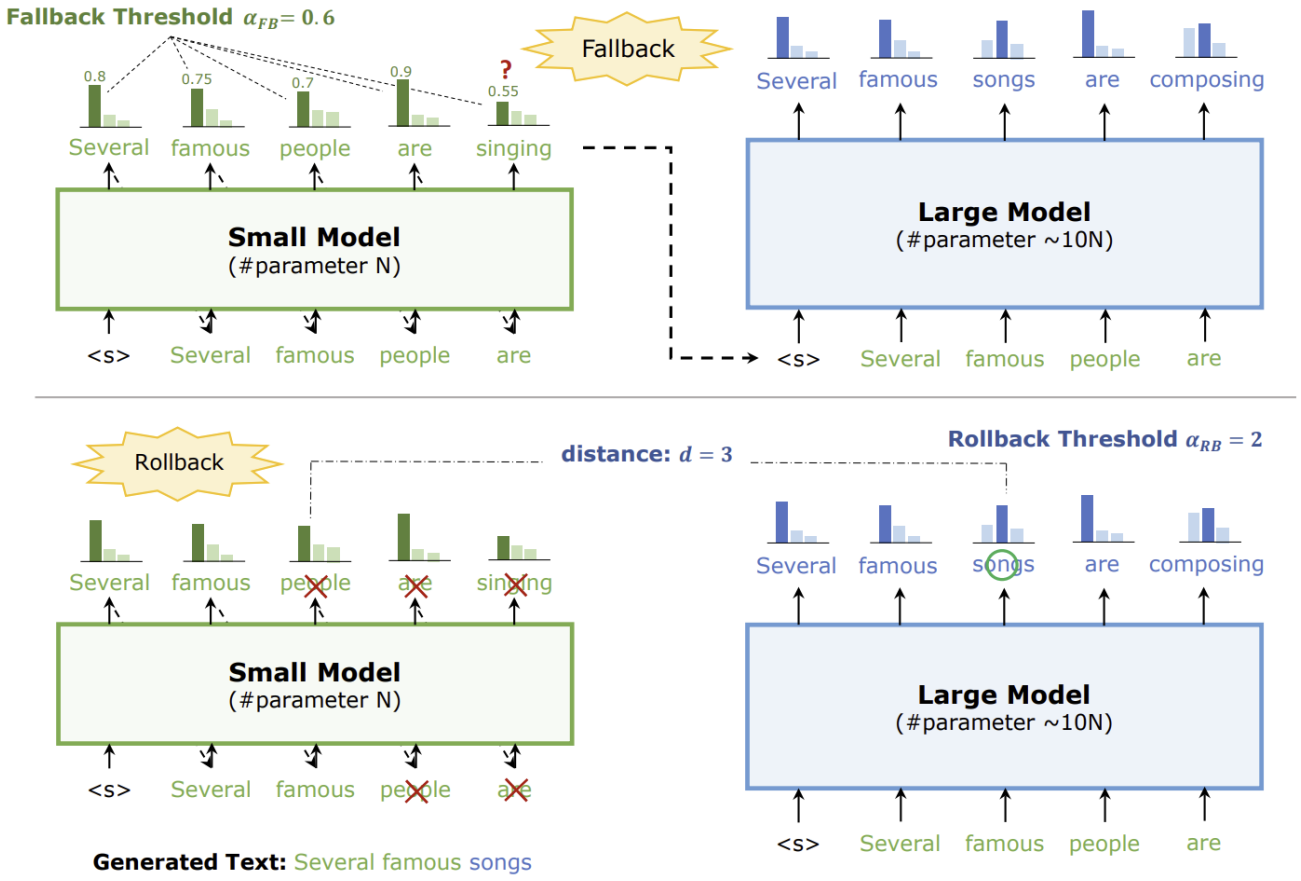


Key links:

1: [Pruning Tutorial Torch]

Image source: https://www.superannotate.com/blog/llm-pruning-distillation-minitron-approach

Speculative Decoding



Key links:

1: Speculative Decoding paper

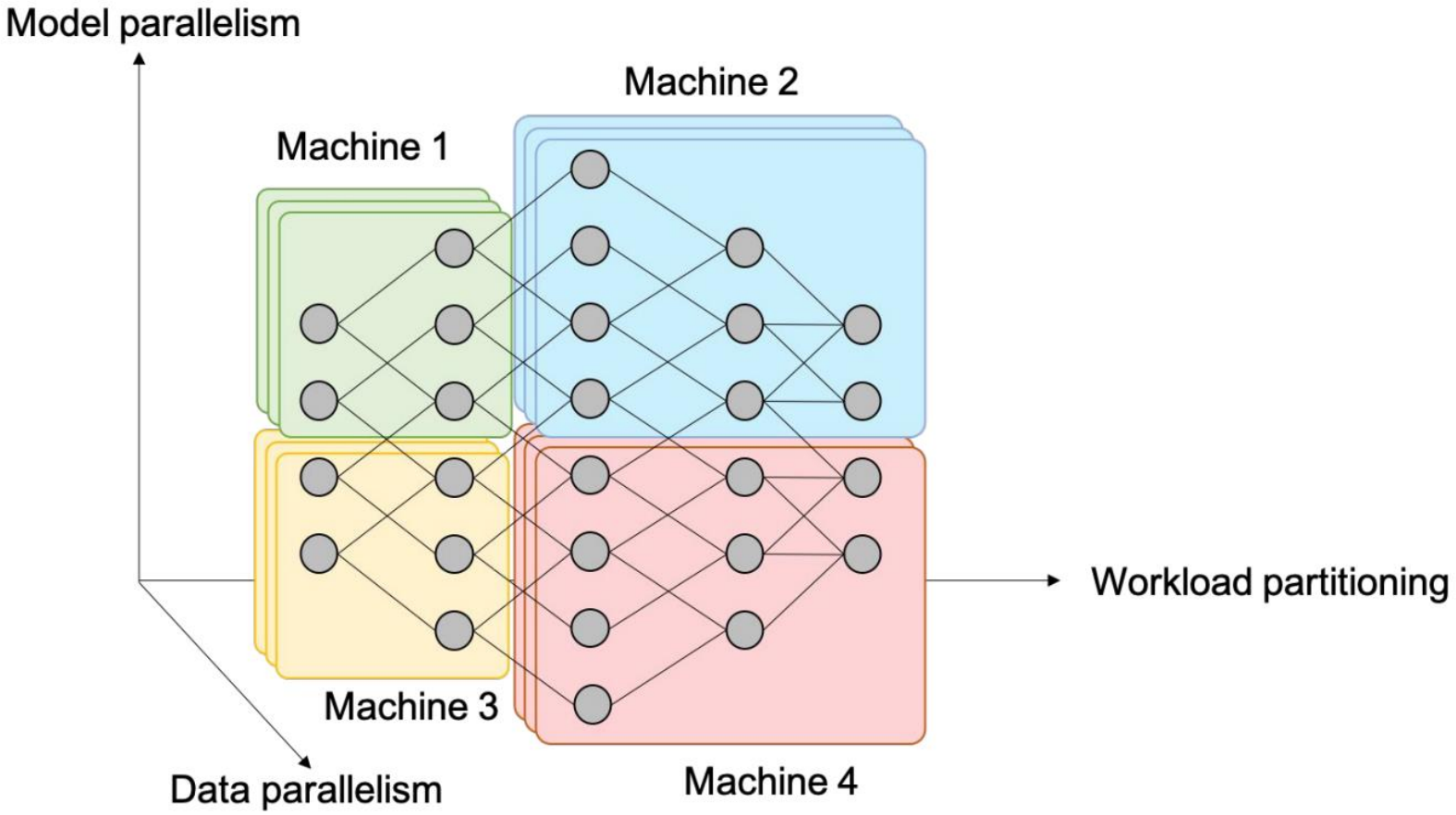
2: A Hitchhiker's Guide to Speculative Decoding

Image source: https://github.com/kssteven418/BigLittleDecoder

Table of Content

- Just finished training, now what?
- Engineering tricks
- Architectural tricks
- **Going distributed**

Distributed is painful on 3 main dimensions



Key links:

1: Torch distributed tutorial

2: Torch Distributed Data Parallel

3: Torch on Model Parallelism

4: On Parameter Servers

5: Accelerate by HF

6: Horovod

7: NCCL by Nvidia

8: DeepSpeed by Microsoft

9: Ray

Image source: Unkown


HSE UNIVERSITY
EST 1992