Speeding Up Learning and Inference

Danna Gurari
University of Colorado Boulder
Spring 2025

https://dannagurari.colorado.edu/course/neural-networks-and-deep-learning-spring-2025/

Review

- Last lecture on model compression:
  - Motivation
  - Pruning
  - Knowledge distillation
  - Final project report: Overleaf tutorial
- Assignments (Canvas):
  - Final project outline due in one week
- Expect a Piazza announcement from Supriya Naidu about course feedback
- Questions?

Today’s Topics

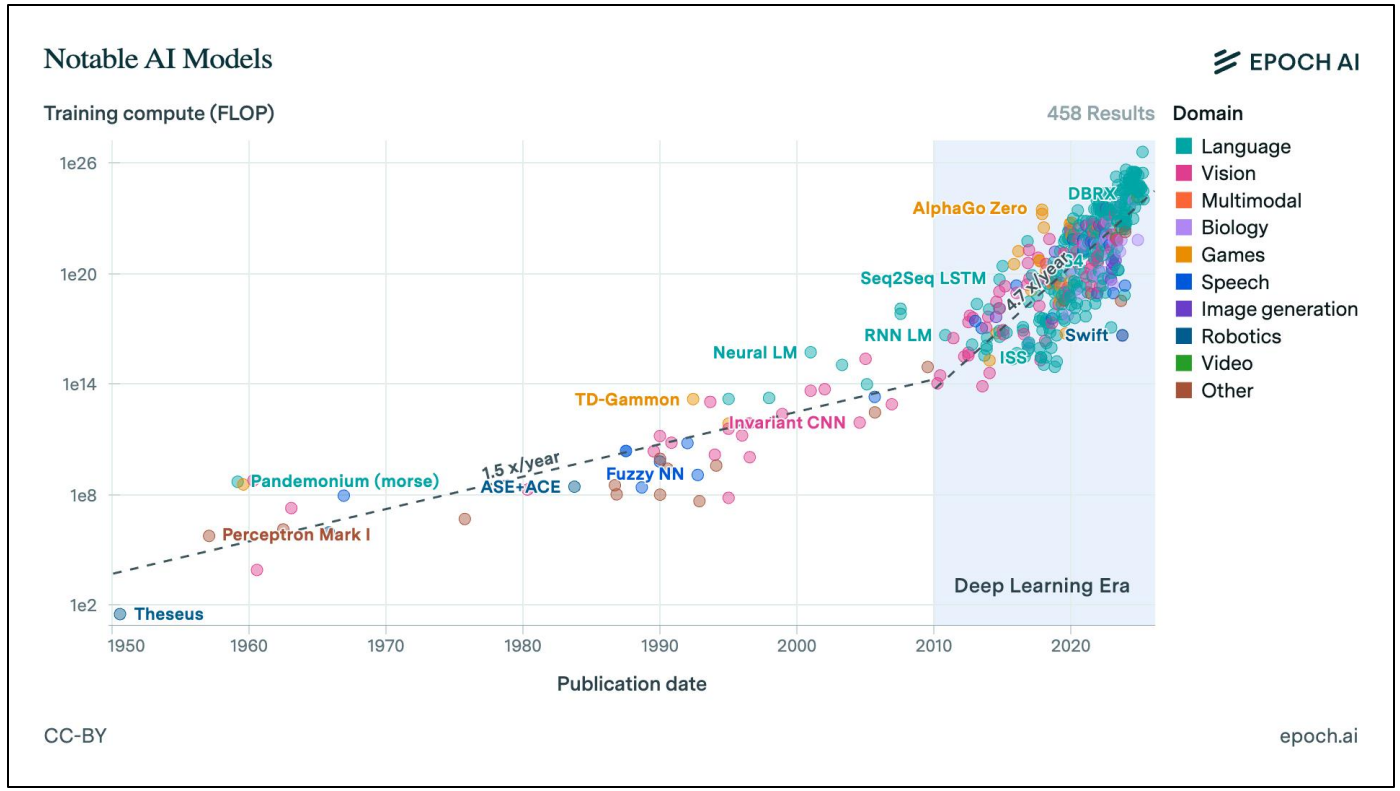
- Motivation
- Hardware tricks
- Architectural tricks
- Training tricks
- Programming tutorial

Today’s Topics

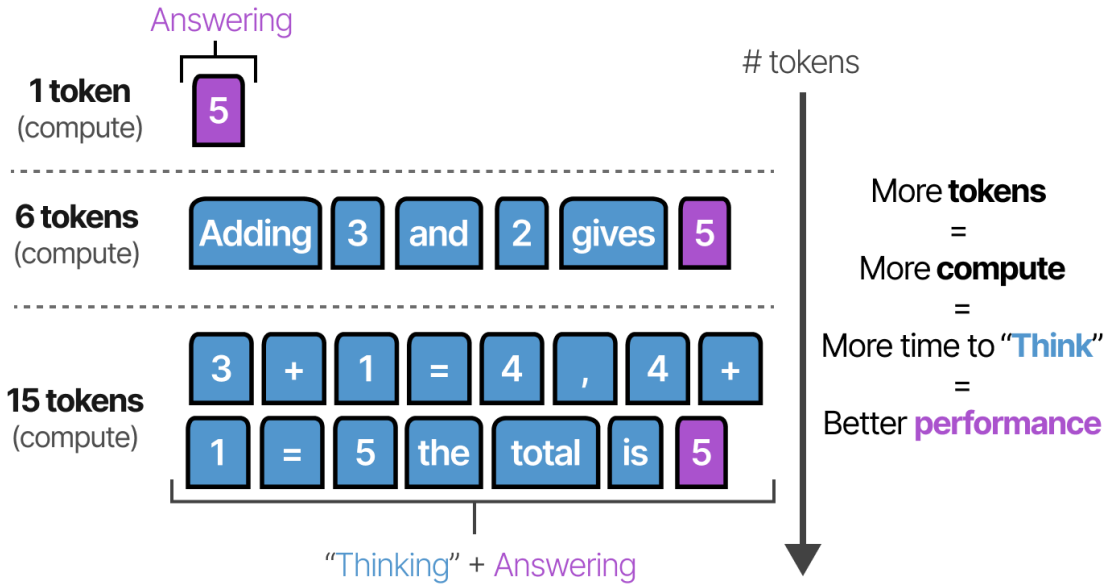
- Motivation
- Hardware tricks
- Architectural tricks
- Training tricks
- Programming tutorial

Trend: More Compute-Intensive Models

Training Time
(more FLOPs)



Inference Time
(more generated tokens)



e.g., A Common Training Situation

Boss: What did you do last month?

You: Trained the model for one epoch. 

Boss: Umm, fine, what is your plan for next month?

You: Train… train the model for one more epoch? 

https://hanlab.mit.edu/files/course/slides/MIT-TinyML-Lec13-Distributed-Training-I.pdf

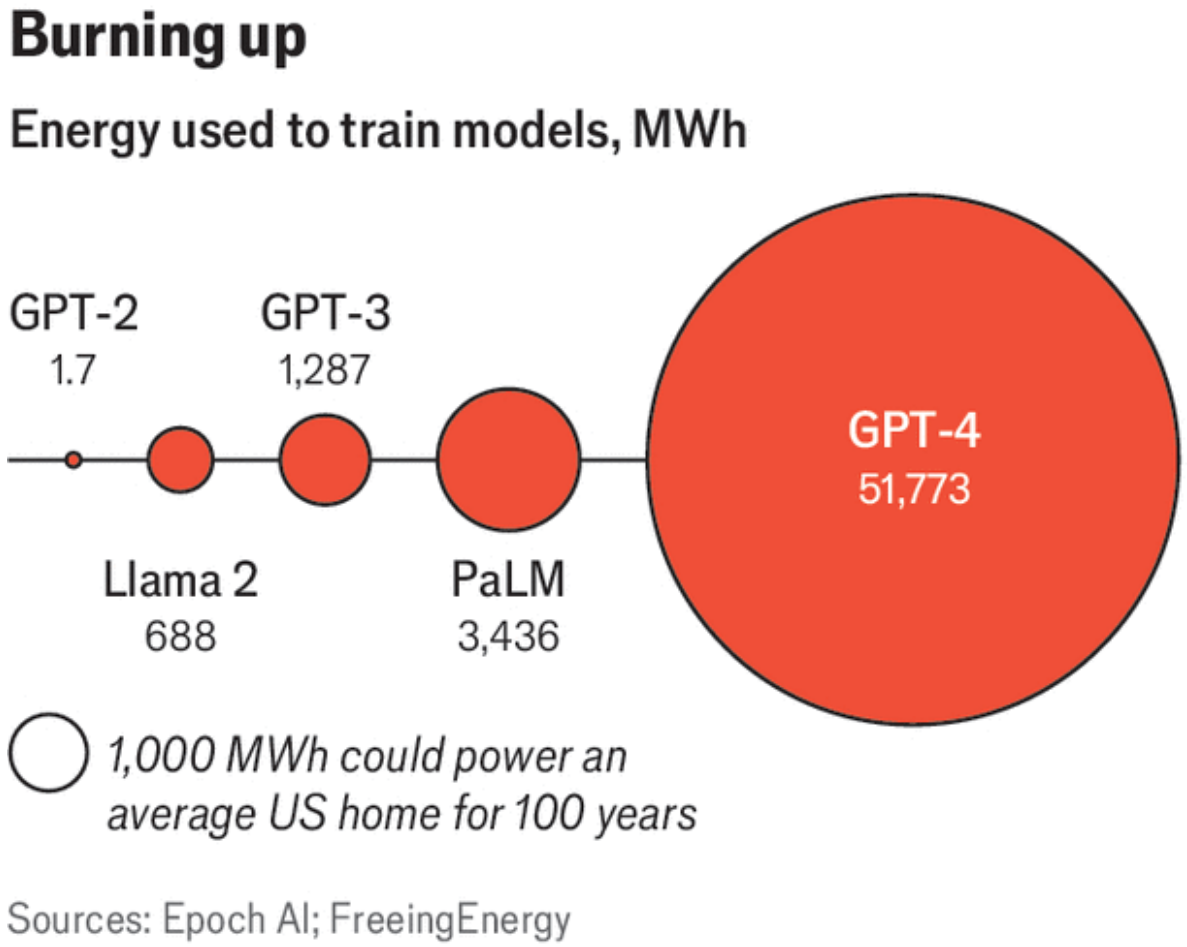
Why Is Extensive Compute Undesirable?

- Time-consuming

- Expensive

- Increased environmental impact from carbon emissions

e.g., A Common Training Situation



Economist, September 21, 2024

e.g., Current Inference Situation

“Google estimates that three-fifths of its total data-center energy use goes on billions of inference queries.”
“On average, a ChatGPT query needs nearly 10 times as much electricity to process as a Google search.”

Economist Sep 21, 2024; https://www.goldmansachs.com/insights/articles/AI-poised-to-drive-160-increase-in-power-demand

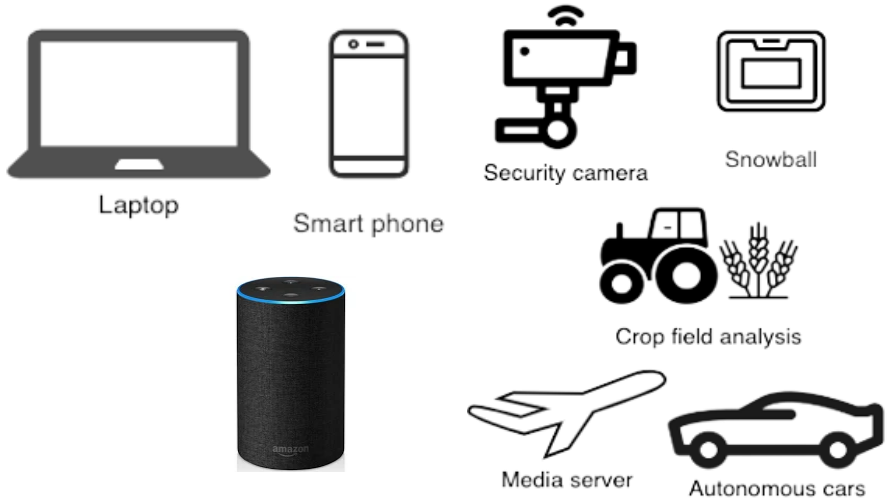
When Is Extensive Compute Unrealistic?

- Learning in resource-constrained settings; e.g., on-device due to privacy concerns and poor/no Internet
- Inference in resource-constrained environments; e.g.,

https://www.ephotozine.com/article/19-things-to-look-out-for-in-a-smartphone-camera--31055

https://en.wikipedia.org/wiki/Wearable_technology

https://www.buzzfeednews.com/article/katienotopoulos/facebook-is-making-camera-glasses-ha-ha-oh-no



https://aws.amazon.com/blogs/machine-learning/demystifying-machine-learning-at-the-edge-through-real-use-cases/

How to develop and use AI models with less compute?

Today's Lecture: A Sampler of HW, Architecture, and Training Tricks

https://beyondmeresustenance.com/mexican-charcuterie-board/

Today’s Topics

- Motivation
- Hardware tricks
- Architectural tricks
- Training tricks
- Programming tutorial

https://breetapp.com/blog/how-to-maximise-your-crypto-investment-portfolio

MAXIMIZE YOUR

RETURN ON INVESTMENT

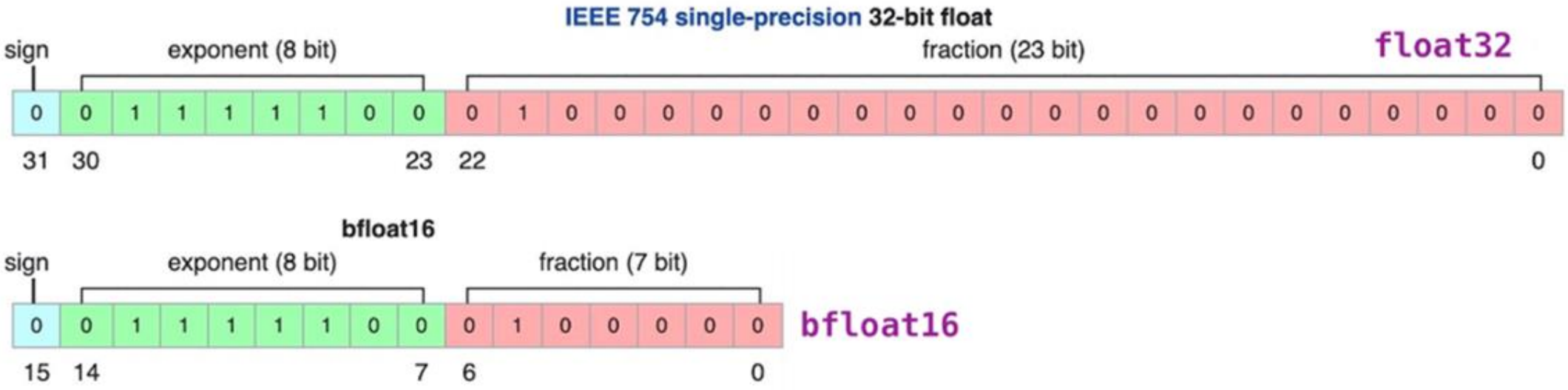
WITH THESE STRATEGIES

Ideas

- Quantization: reduce precision
- Operator fusion: reduce memory read/writes
- Caching: reduce computation
- Distributed optimization: parallelize computation

Quantization: Reduce Precision

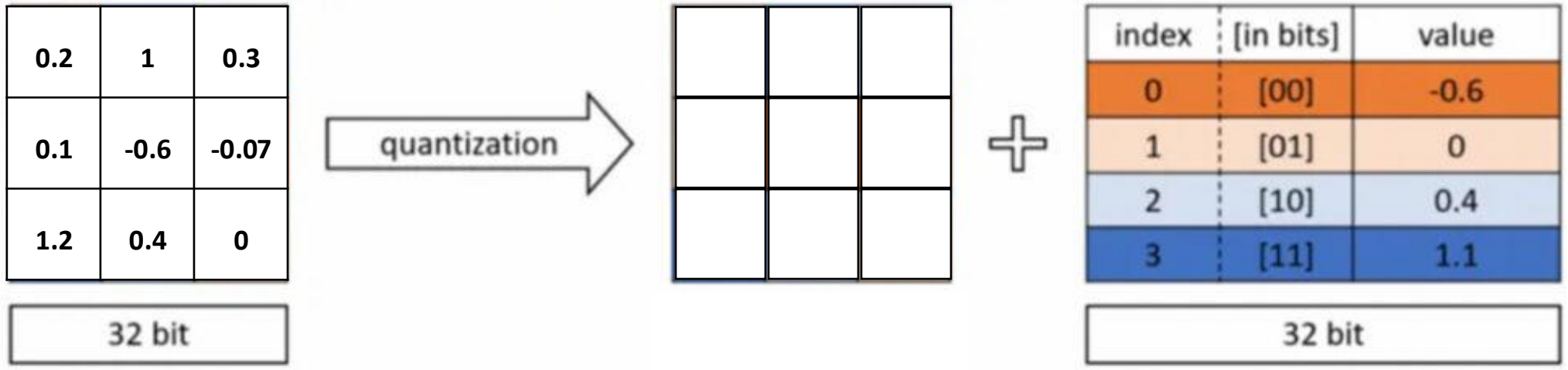
- Which precision and ranges do different data types offer?
  - 32 bits: e.g., ~ $\pm 3.4 \times 10^{38}$ with 2^{32} or ~4 billion values (default PyTorch type)
  - 8 bits: [-127, 127]
  - 4 bits: [-8, 7]



https://medium.com/@dillipprasad60/qlora-explained-a-deep-dive-into-parametric-efficient-fine-tuning-in-large-language-models-llms-c1a4794b1766

Quantization: Reduce Precision

- e.g.,
  - what number of bits are used in this example?
  - what bin value should be used for each of the 9 quantized values?

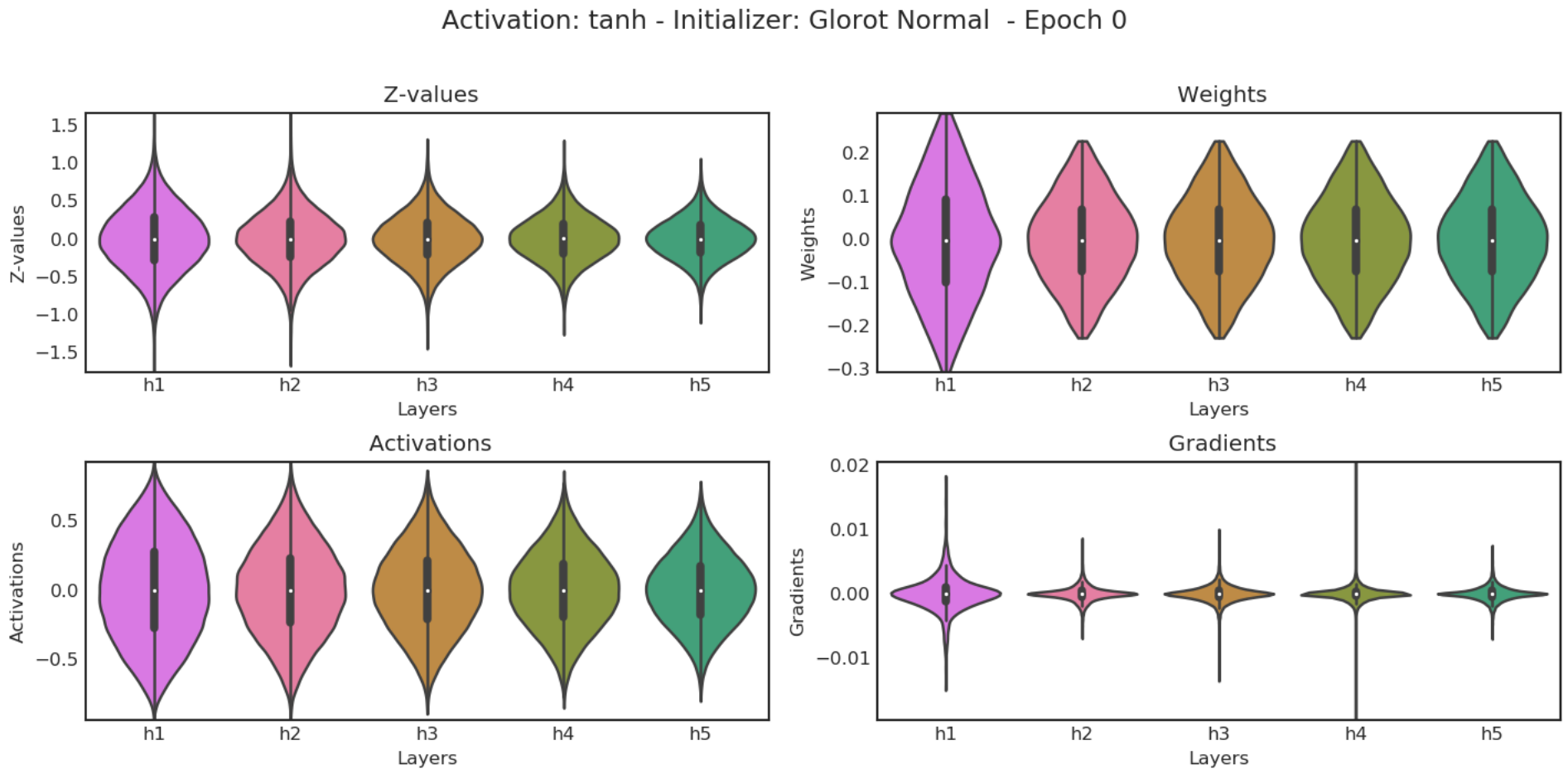


https://xailient.com/blog/4-popular-model-compression-techniques-explained/

Quantization at Training

How much precision is needed for training?

e.g., recall batch normalization causes values to center on 0 and range roughly between -1 and 1



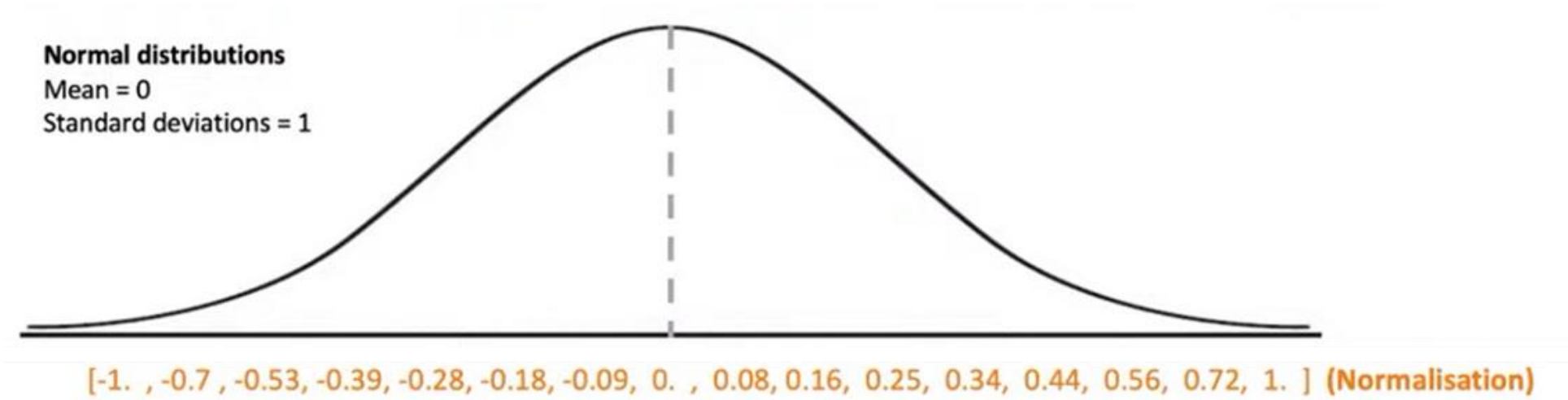
https://towardsdatascience.com/hyper-parameters-in-action-part-ii-weight-initializers-35aee1a28404

Quantization at Training

How much precision is needed for training?

e.g., recall batch normalization causes values to center on 0 and range roughly between -1 and 1

e.g., 4 bits means values mapped to 16 floating point bins (i.e., 2^4 possible values)



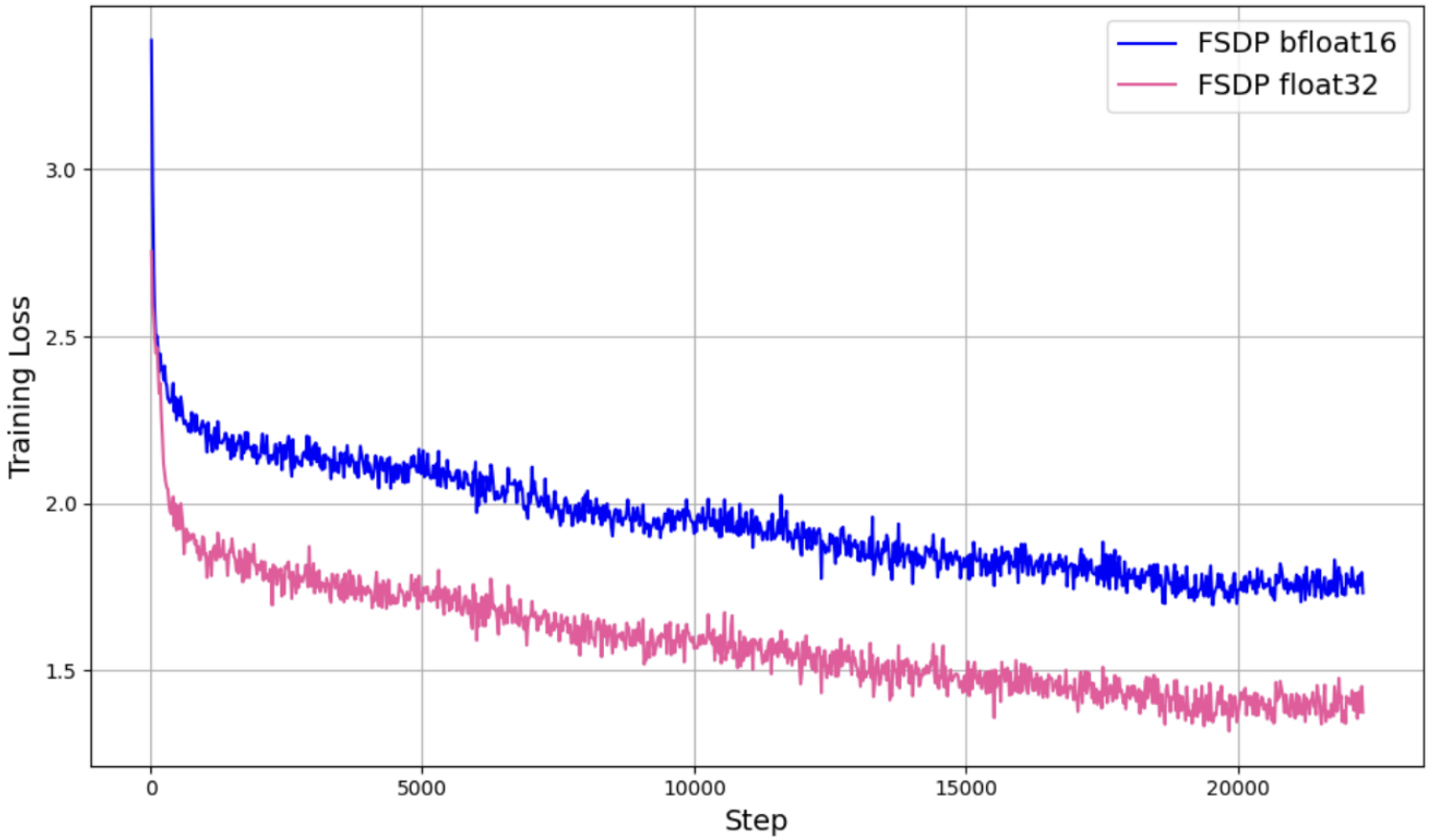
https://medium.com/@dillipprasad60/qlora-explained-a-deep-dive-into-parametric-efficient-fine-tuning-in-large-language-models-llms-c1a4794b1766

Quantization at Training

How much precision is needed for training?

e.g., recall batch normalization causes values to center on 0 and range roughly between -1 and 1

What are risks of using fewer bits?



[Dietke, CVPR25]

Quantization at Training

What are risks and benefits of using fewer bits (e.g., when done with pruning)?

How much precision is needed for training?

e.g., recall batch normalization causes values to center on 0 and range roughly between -1 and 1

	Actual No. Of Parameters/ size	Actual Top-5 Error Rate (%)	Method Type	Para./size after compression	Compression Achieved	Top-5 Error Rate (%) after Compression	Speedup Achieved
AlexNet	61M/240 MB	19.7	Pruning	6.7M	9x	19.67	3x
			Pruning and quantization	6.9 MB	35x	19.7	3x
VGG16	138M/512 MB	10.4	Pruning	10.3M	13x	10.88	5x
			Pruning and quantization	11.3 MB	49x	10.91	3x to 4x

https://xailient.com/blog/4-popular-model-compression-techniques-explained/

Quantization Options

- e.g.,

	A100 80GB PCIe	A100 80GB SXM
FP64 (Exceeds default)	9.7 TFLOPS	
FP64 Tensor Core (Exceeds default)	19.5 TFLOPS	
FP32 (default)	19.5 TFLOPS	
Tensor Float 32 (TF32)	156 TFLOPS	312 TFLOPS*
BFLOAT16 Tensor Core	312 TFLOPS	624 TFLOPS*
FP16 Tensor Core	312 TFLOPS	624 TFLOPS*
INT8 Tensor Core	624 TOPS	1248 TOPS*

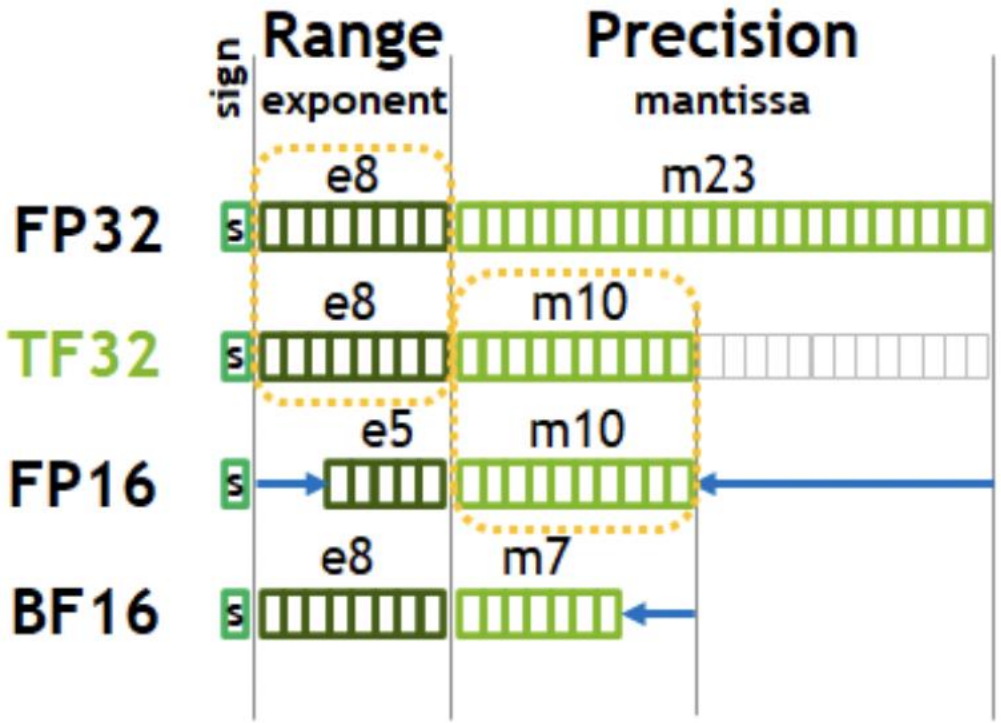
How much is the speed-up?
(without *sparsity)

https://www.nvidia.com/en-us/data-center/a100/

Quantization Options

- e.g.,

FP64
FP64 Tensor Core
FP32
Tensor Float 32 (TF32)
BFLOAT16 Tensor Core
FP16 Tensor Core
INT8 Tensor Core



Greater speed-ups from greater cuts in precision!

https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf

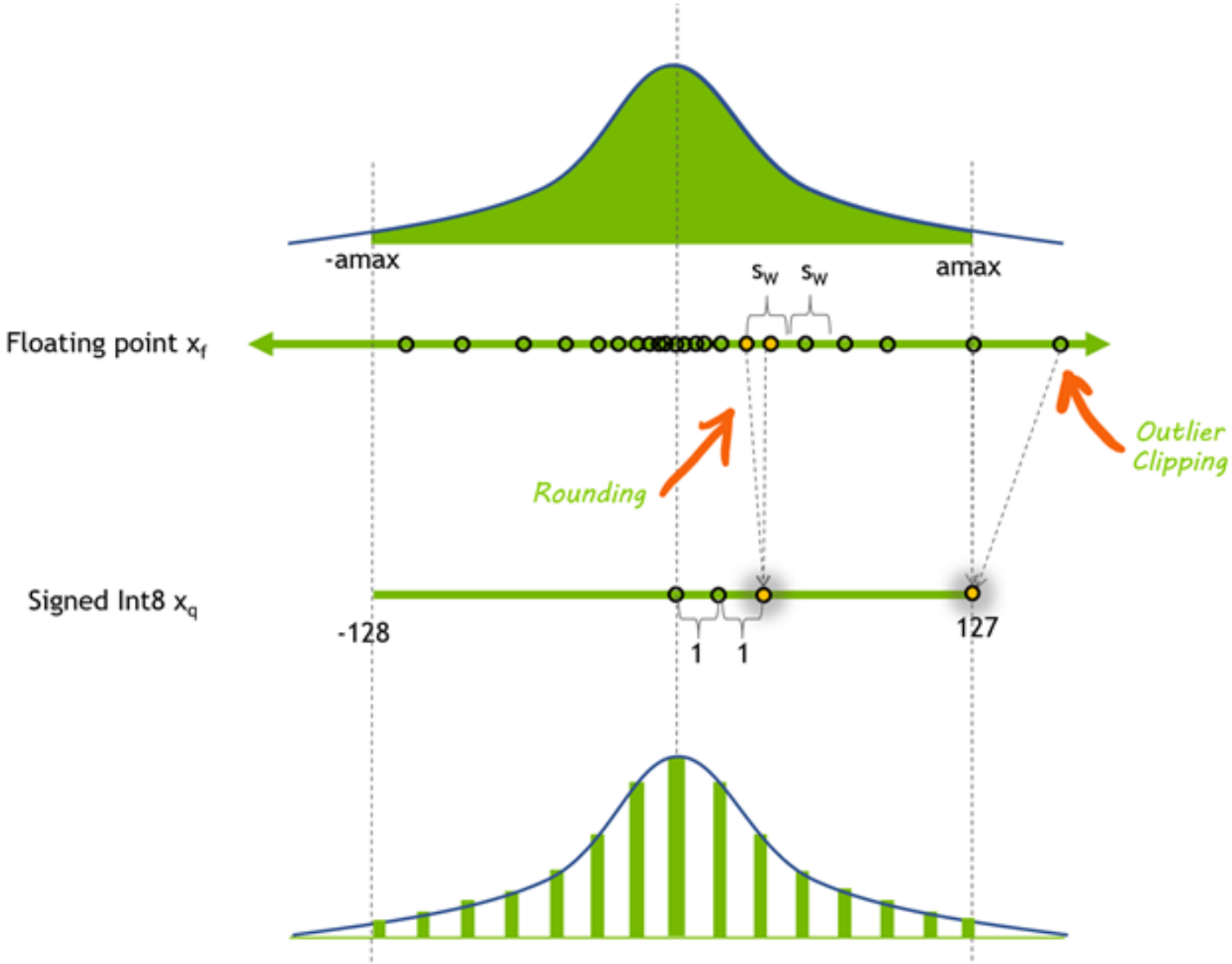
Quantization Options: A100 vs H200

NVIDIA H200 Tensor Core GPU		A100 80GB PCIe
Form Factor	**H200 SXM[1]**	
FP64	34 TFLOPS	9.7 TFLOPS
FP64 Tensor Core	67 TFLOPS	19.5 TFLOPS
FP32	67 TFLOPS	19.5 TFLOPS
TF32 Tensor Core	989 TFLOPS[2]	156 TFLOPS | 312 TFLOPS*
BFLOAT16 Tensor Core	1,979 TFLOPS[2]	312 TFLOPS | 624 TFLOPS*
FP16 Tensor Core	1,979 TFLOPS[2]	312 TFLOPS | 624 TFLOPS*
INT8 Tensor Core	3,958 TFLOPS[2]	624 TOPS | 1248 TOPS*

Use for training?

https://www.fibermall.com/blog/why-gpu-require-hbm.htm; https://www.nvidia.com/en-us/data-center/a100/

Quantization at Inference

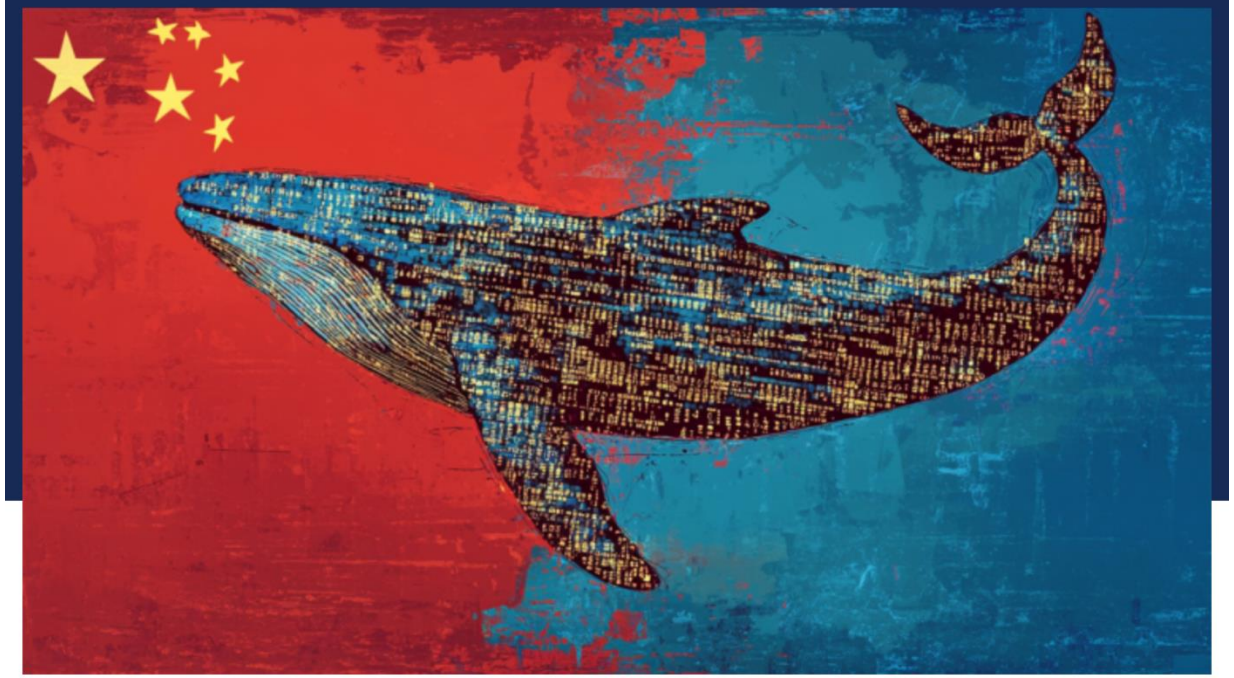


e.g., scale to [-128, 127] with 8 bits

- Outliers set to minimum and maximum

https://developer.nvidia.com/blog/achieving-fp32-accuracy-for-int8-inference-using-quantization-aware-training-with-tensorrt/

DeepSeek-V3 now runs at 20 tokens per second on Mac Studio, and that's a nightmare for OpenAI



Awni Hannun @awnihannun · Mar 24

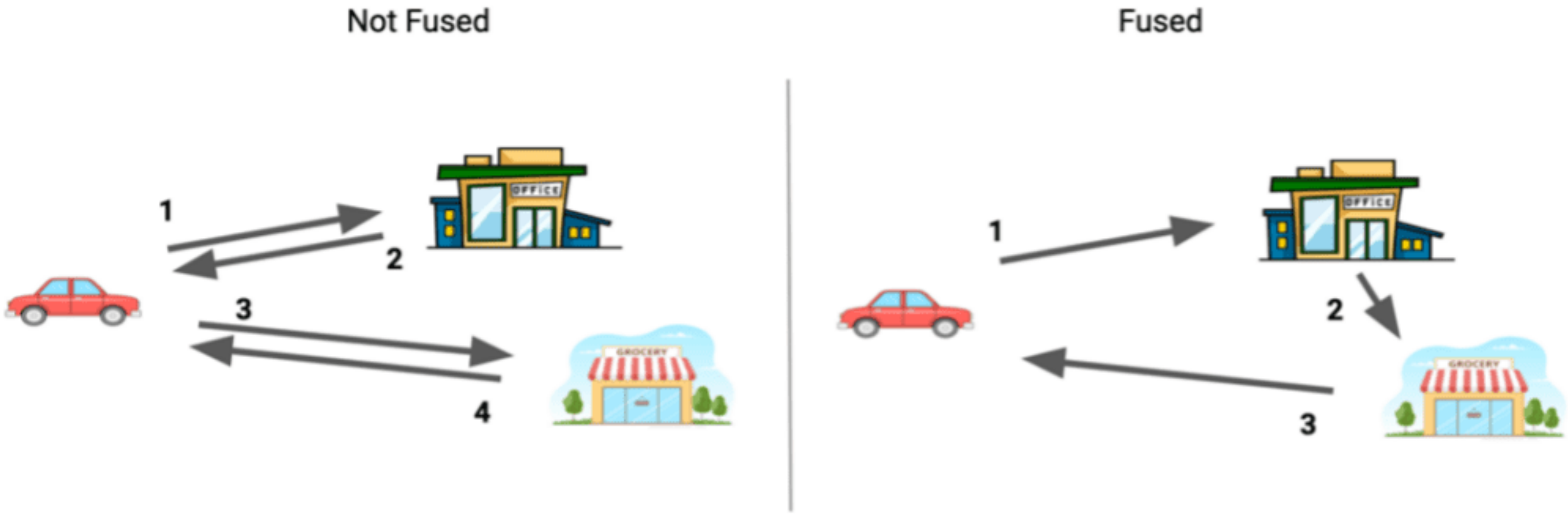
The new Deep Seek V3 0324 in 4-bit runs at > 20 toks/sec on a 512GB M3 Ultra with mlx-lm!

https://venturebeat.com/ai/deepseek-v3-now-runs-at-20-tokens-per-second-on-mac-studio-and-thats-a-nightmare-for-openai/

Ideas

- Quantization: reduce precision
- Operator fusion: reduce memory read/writes
- Caching: reduce computation
- Distributed optimization: parallelize computation

Idea: Fuse Tasks to Reduce Overhead



(e.g., go to school, return home, buy groceries)

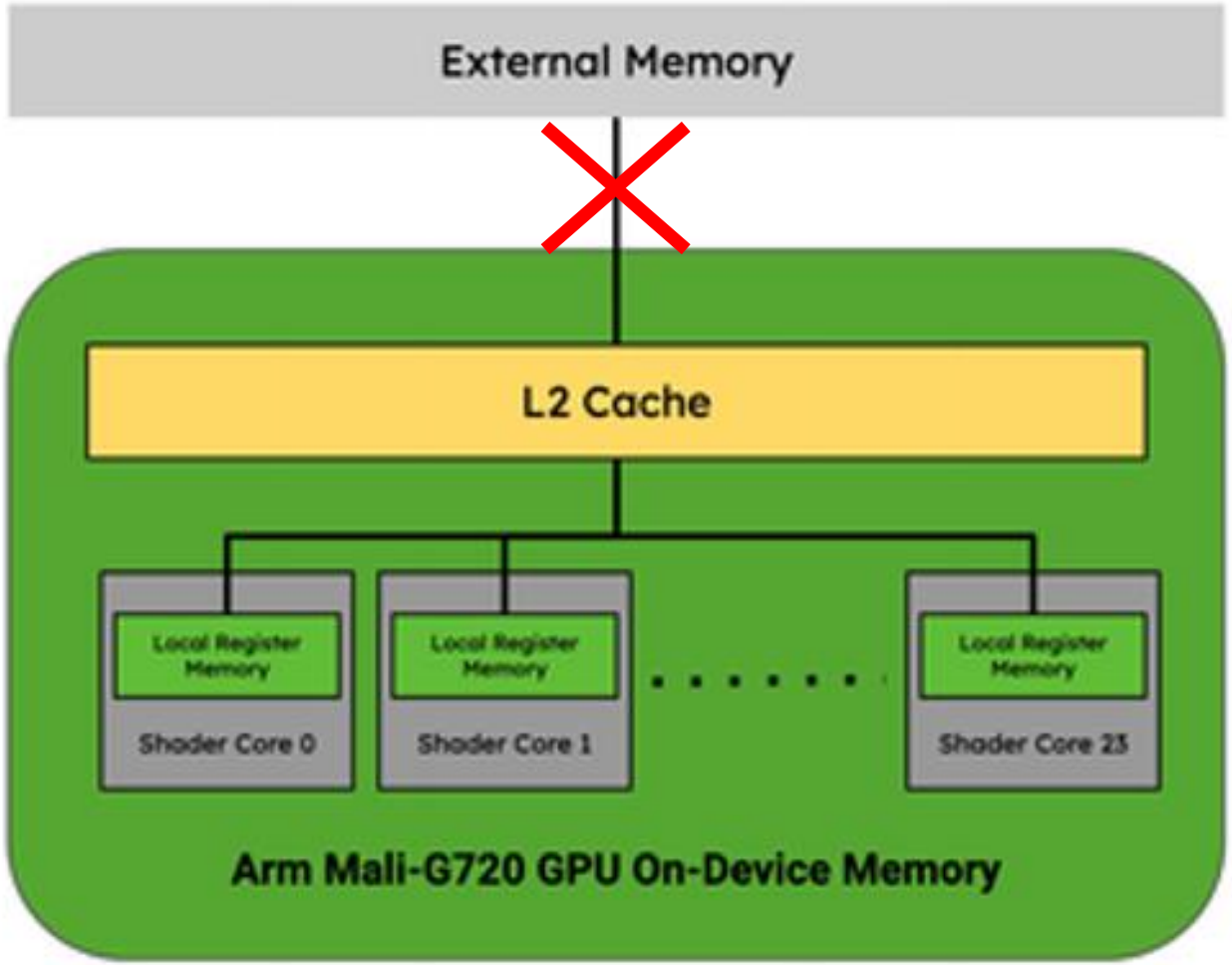
(e.g., go to school and then buy groceries)

https://quadric.io/2023/09/13/how-to-unlock-the-power-of-operator-fusion-to-accelerate-ai/

Approach: Fuse Tasks to Reduce Overhead

Avoid shuttling data back-and-forth to external memory by determining what operations to fuse so relevant information stays in limited memory close to the GPU

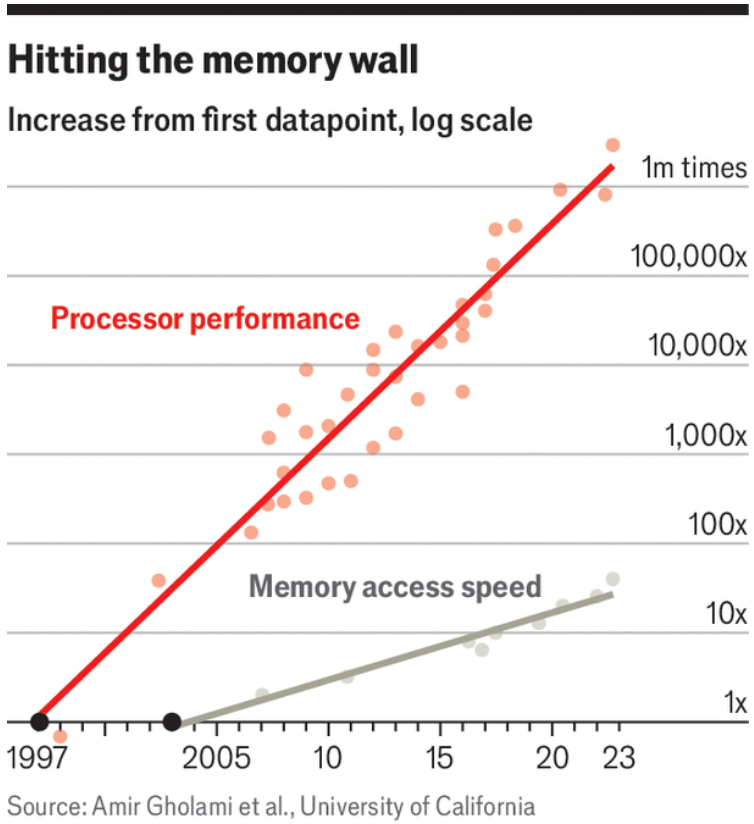
- When would you prefer L2 cache versus Load Register Memory?



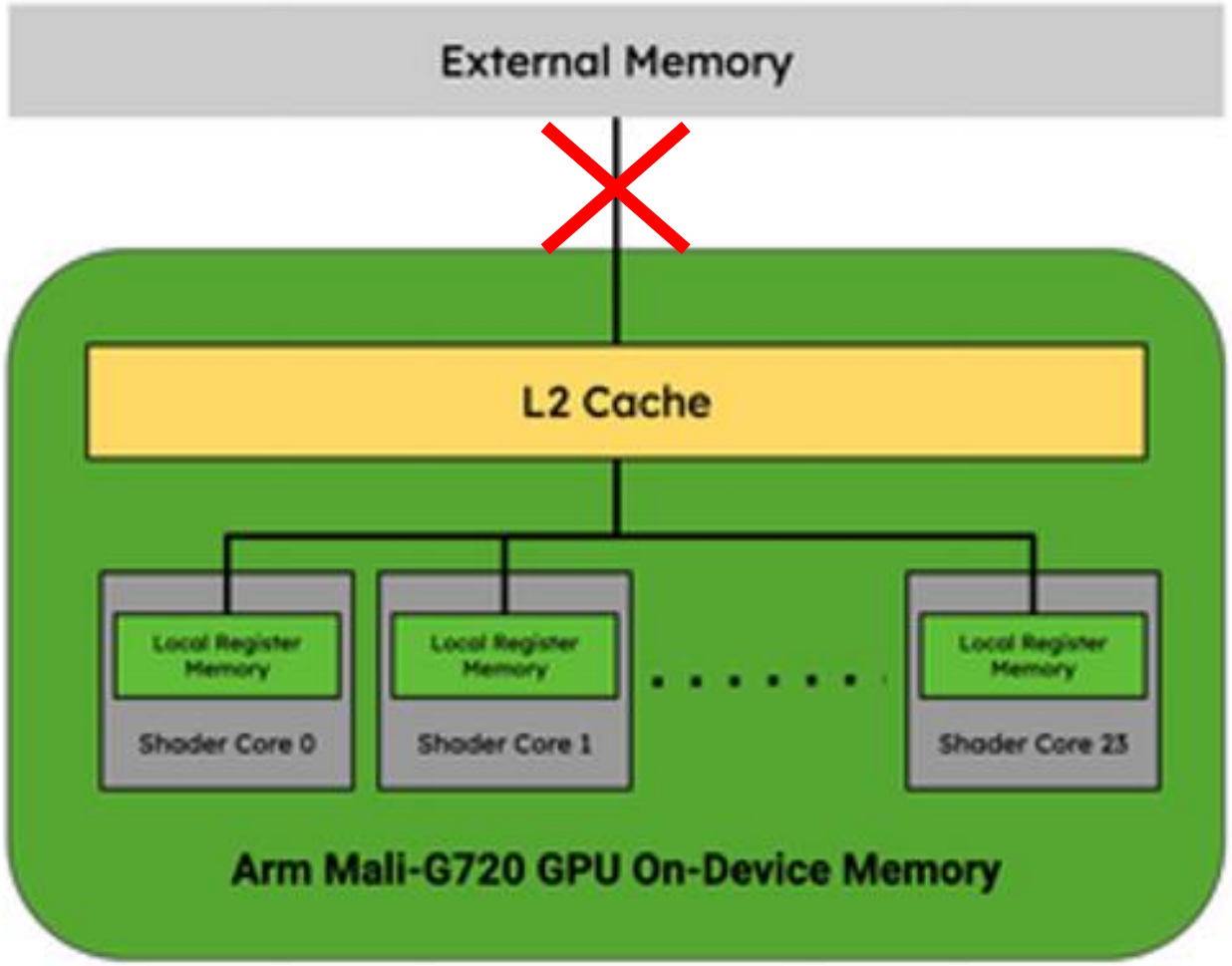
https://quadric.io/2023/09/13/how-to-unlock-the-power-of-operator-fusion-to-accelerate-ai/

Approach: Fuse Tasks to Reduce Overhead

This approach is especially important given HW trends: processor performance (tripling every two years) exceeds that of memory access speed (~half as much)



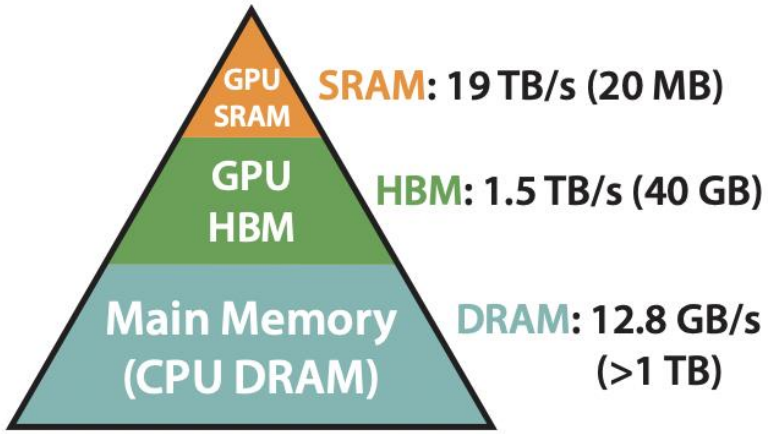
(Economist, Sep 21, 2024)



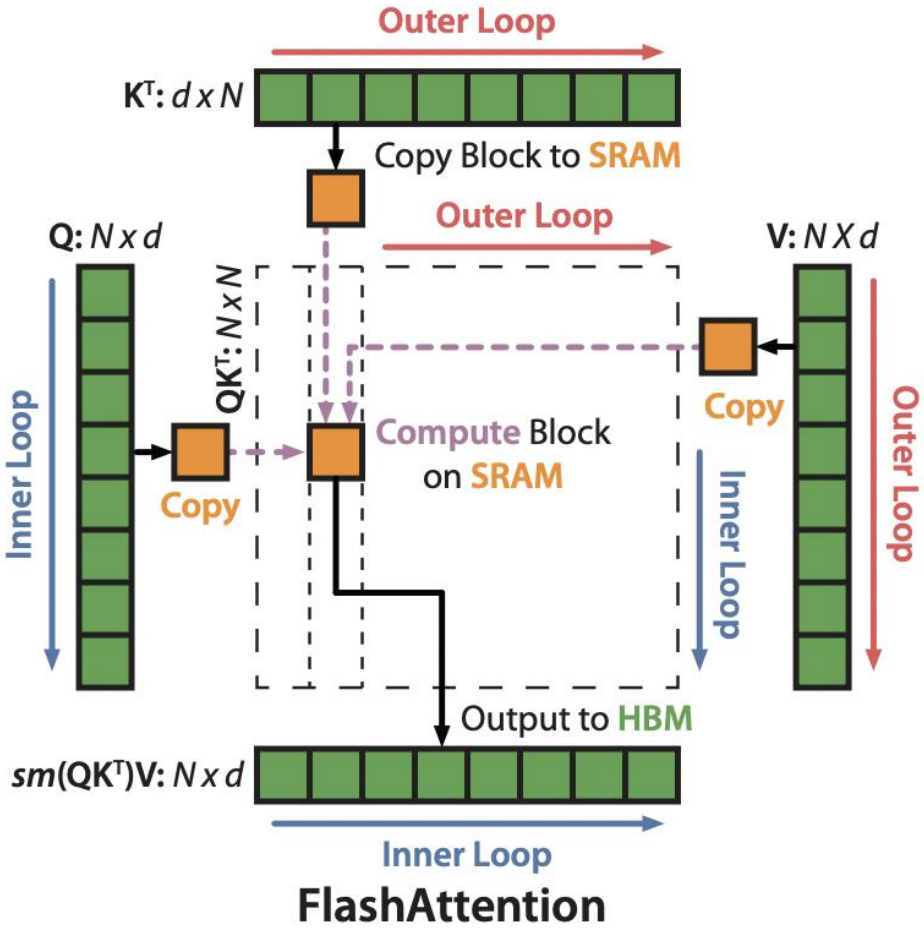
https://quadric.io/2023/09/13/how-to-unlock-the-power-of-operator-fusion-to-accelerate-ai/

FlashAttention: Fuses Attention Operations

Relevant information stays in limited memory close to GPU



Memory Hierarchy with Bandwidth & Memory Size

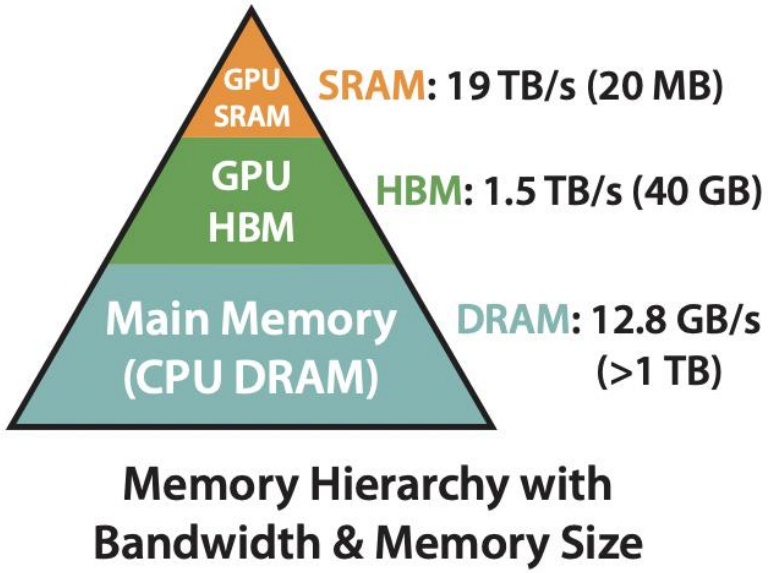


FlashAttention

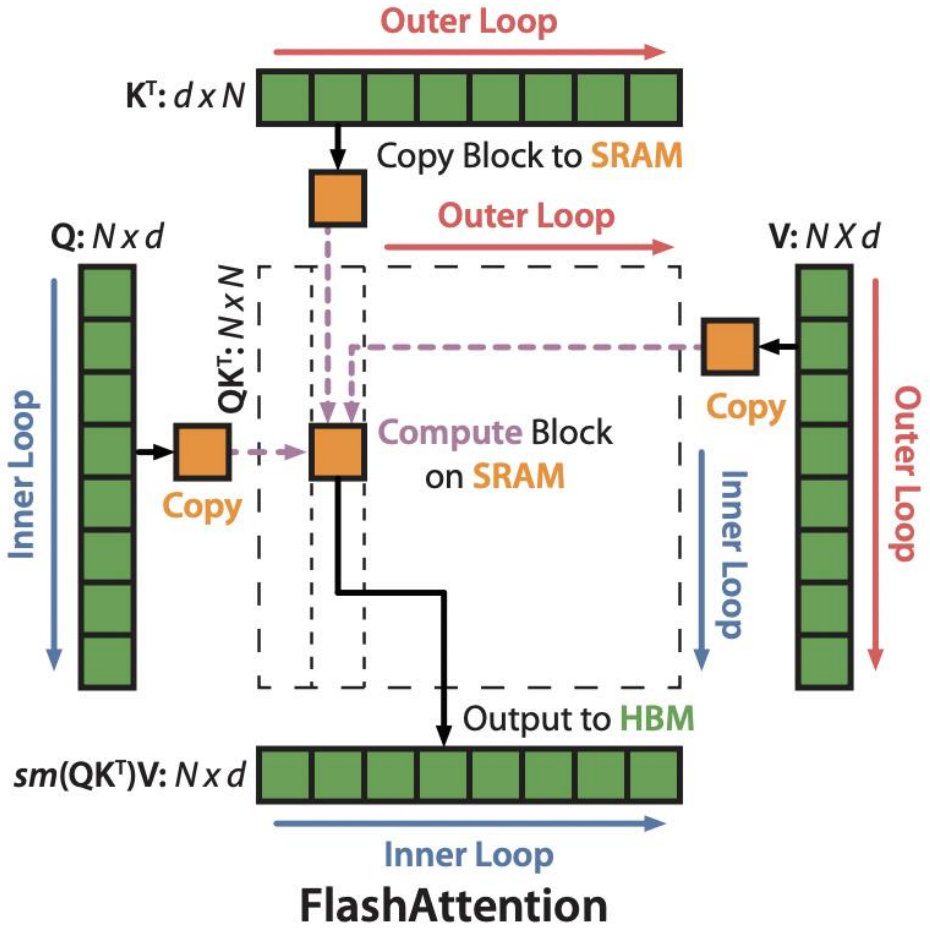
Dao et al. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. Neurips 2022

FlashAttention: Fuses Attention Operations

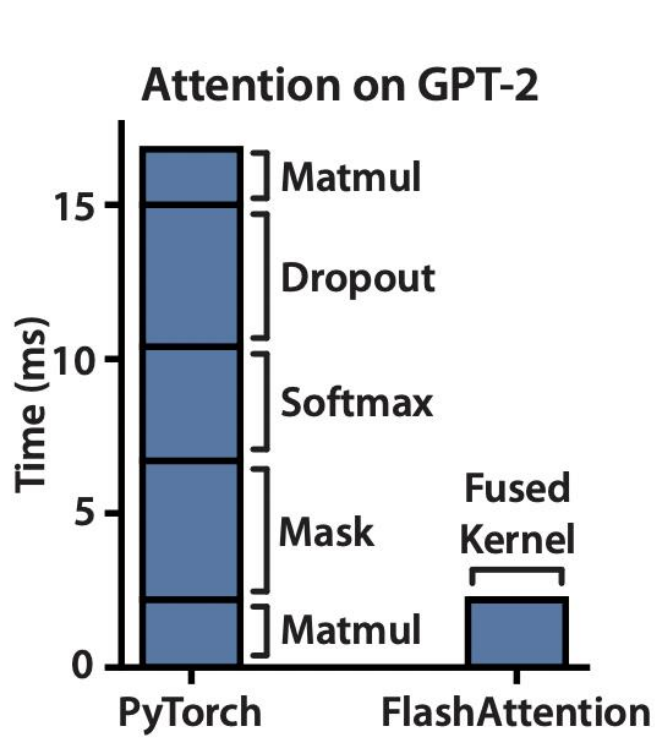
Relevant information stays in limited memory close to GPU



Memory Hierarchy with Bandwidth & Memory Size

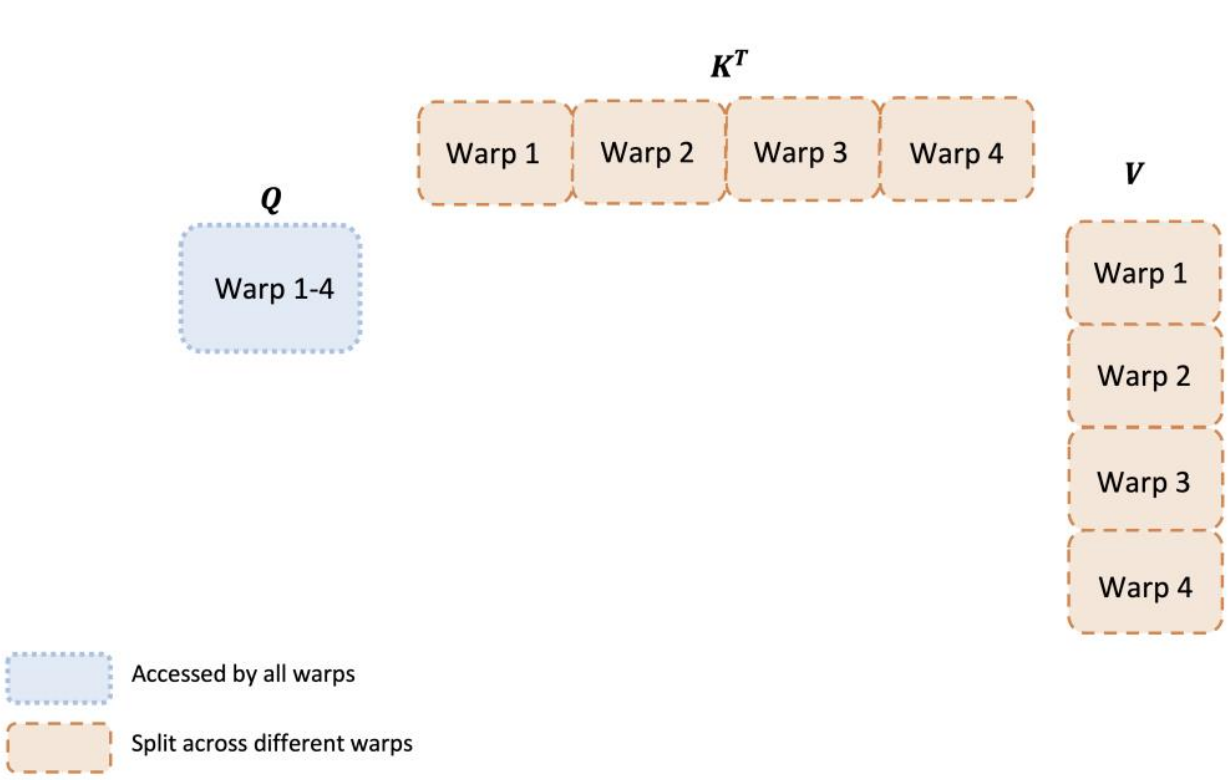


FlashAttention

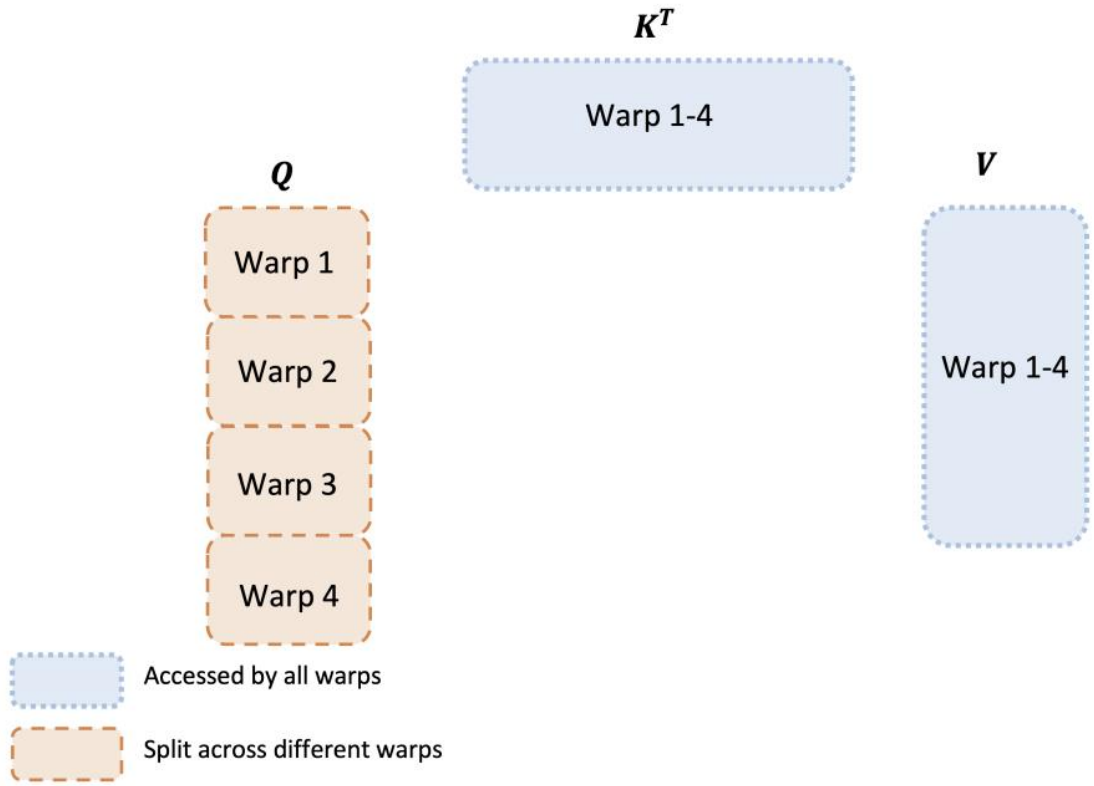


Dao et al. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. Neurips 2022

FlashAttention2



(a) FlashAttention



(b) FlashAttention-2

An estimated 2x speed-up compared to FlashAttention!

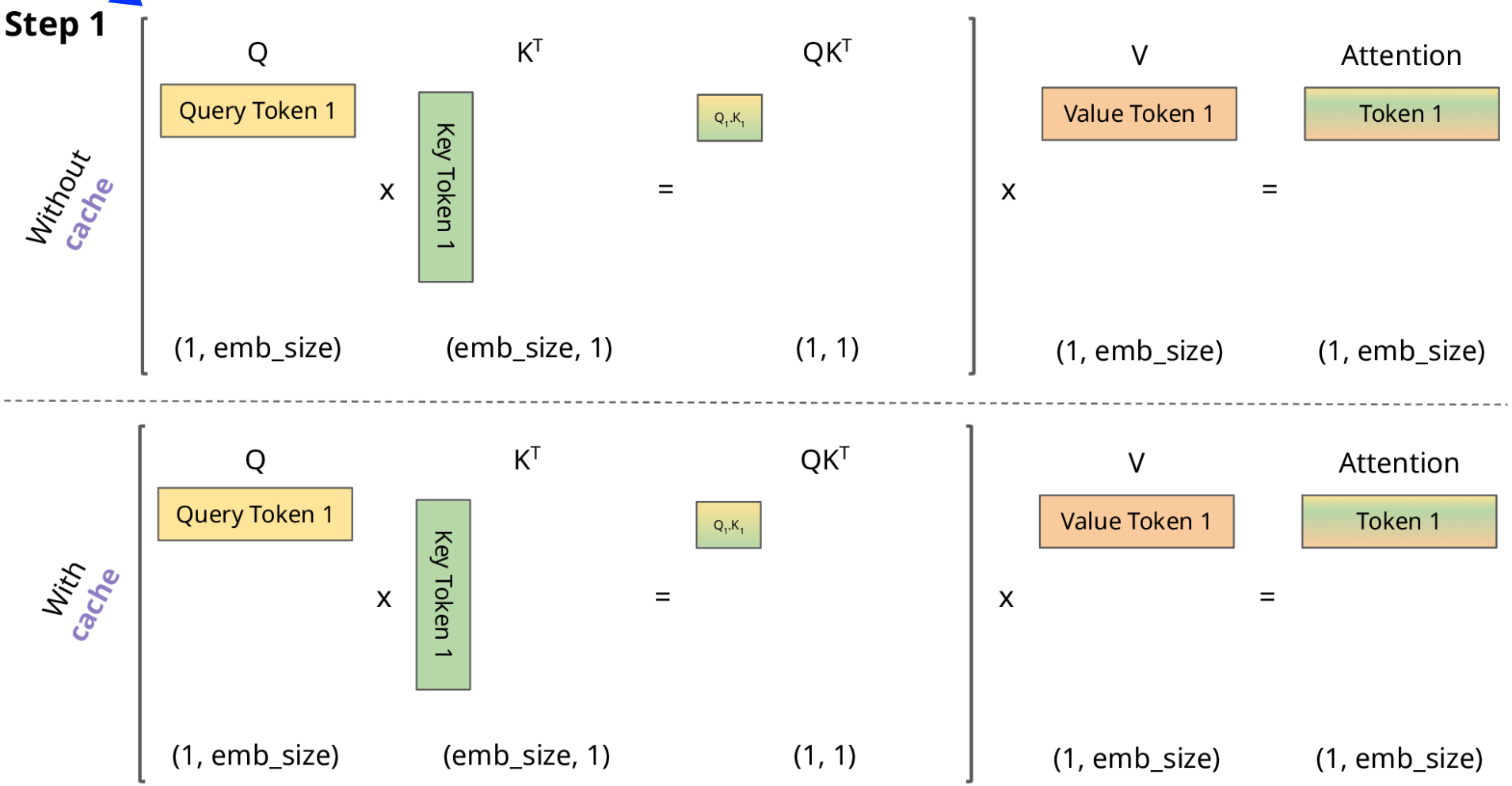
Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. arXiv 2023

Ideas

- Quantization: reduce precision
- Operator fusion: reduce memory read/writes
- Caching: reduce computation
- Distributed optimization: parallelize computation

KV Caching: Store Keys and Values for Attention

Only applies to generative (autoregressive) models



(Operations repeated in future time steps have results stored in cache)

Limitations: uses memory inefficiently since needed cache size regularly changes and reserves lots of cache memory to support large sizes

https://medium.com/@joaolages/kv-caching-explained-276520203249

vLLM: Overcomes Limitations of KV Caching

- Inspiration: virtual memory with paging used in operating systems to limit memory fragmentation by allocating memory “pages” on demand
- Idea: an LLM serving engine using PageAttention, which allocates KV cache into fixed-size blocks (aka, “pages”) that don’t have to be contiguous

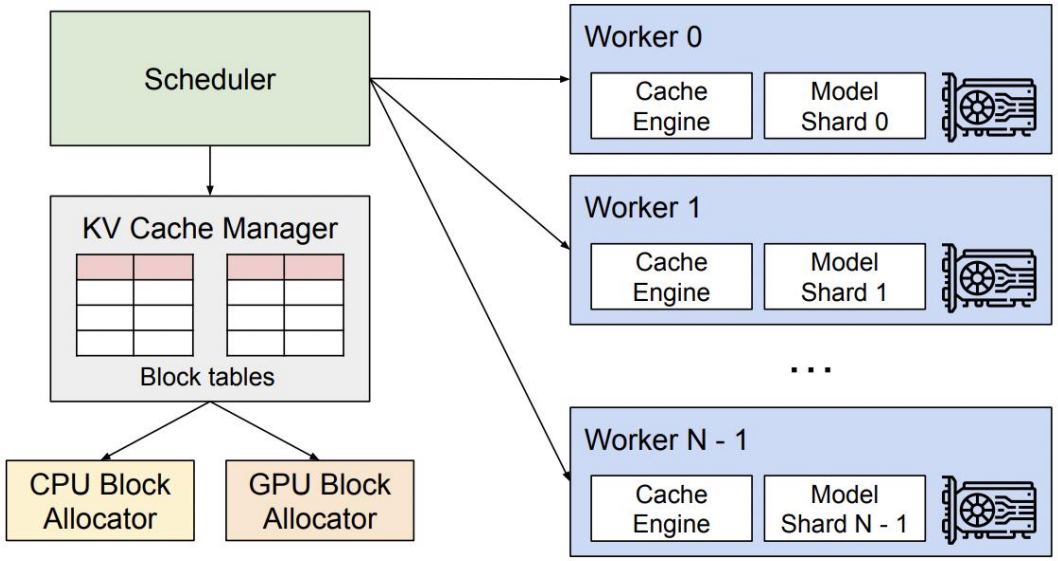


Figure 4. vLLM system overview.

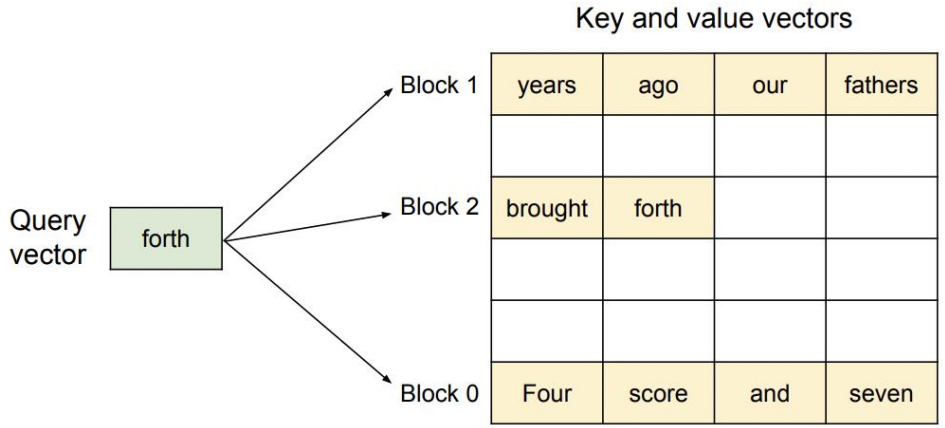


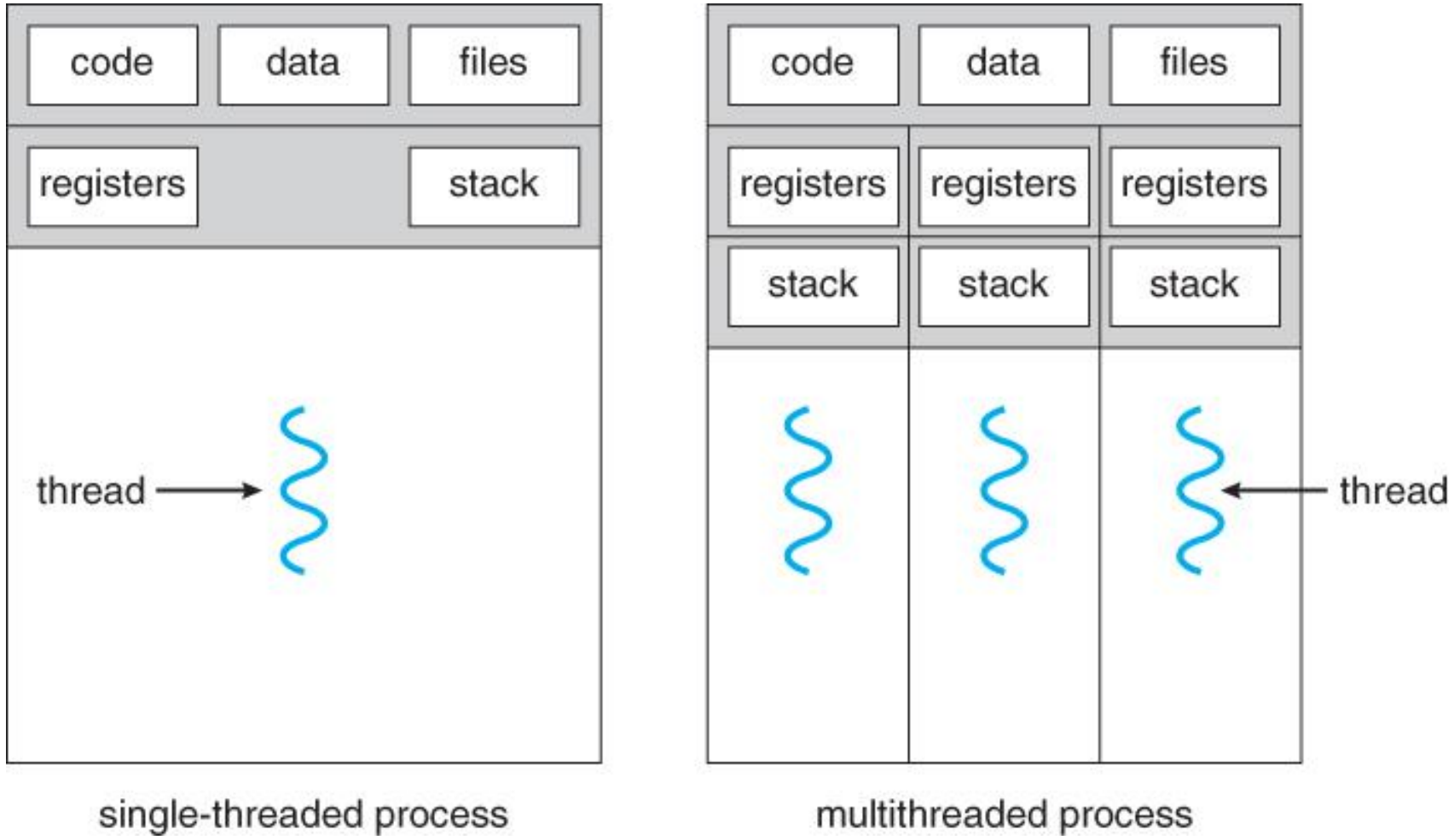
Figure 5. Illustration of the PagedAttention algorithm, where the attention key and values vectors are stored as non-contiguous blocks in the memory.

Kwon et al. Efficient Memory Management for Large Language Model Serving with PagedAttention. SOSP 2023

Ideas

- Quantization: reduce precision
- Operator fusion: reduce memory read/writes
- Caching: reduce computation
- Distributed optimization: parallelize computation

Recall Multi-Thread Programming



https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/4_Threads.html

Distribute Neural Network Computations Across Multiple Devices (e.g., GPUs, CPUs)

- Key questions:
  - What tasks to delegate to each device?
  - What communication to facilitate across devices and how to do this?
- Popular Methods:
  - Distributed Data Parallel (DDP): each "machine" holds copy of model to perform forward and backward passes on different data subsets, with cross-machine communication (i.e., check-pointing) to identify average gradients across machines
    - If you had 4 GPUs, approximately what amount of speed-up might you expect?
  - Distributed Lower-Communication Training (DiLoCo): distributes training to multiple "islands" of a massive cluster (e.g., 100,000 GPUs), with regular checkpointing in each and less checkpointing across them, enabling devices to come from different sources (e.g., locations, companies, countries)
  - Zero Redundancy Optimizer (ZeRO): supports parallelism *without* code changes; partitions model training states (e.g., weights, gradients) across devices, enabling training of very large models (e.g., trillions of parameters)

Summary: How to Get a Greater Return on Your HW Investments

- Quantization: reduce precision
- Operator fusion: reduce memory read/writes
- Caching: reduce computation
- Distributed optimization: parallelize computation

Today’s Topics

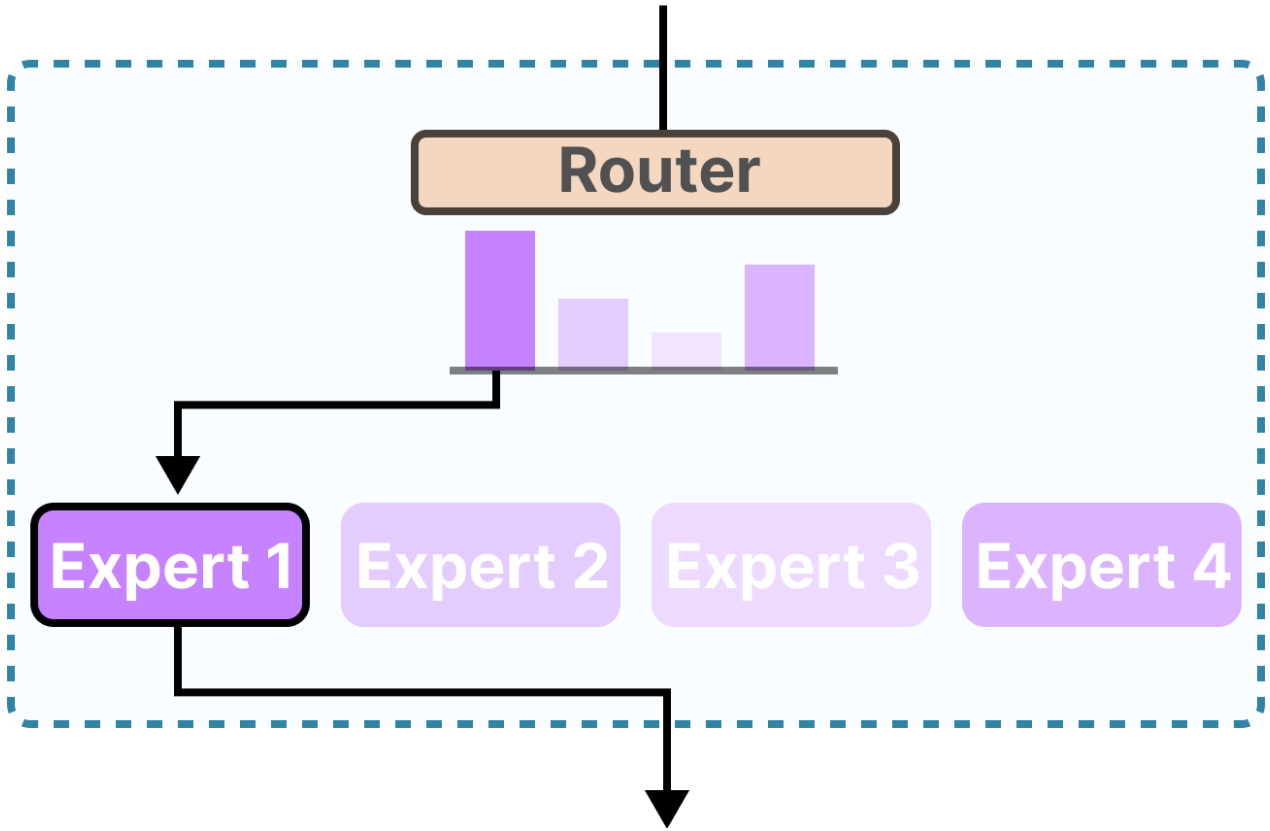
- Motivation
- Hardware tricks
- Architectural tricks
- Training tricks
- Programming tutorial

Ideas

- Mixture of experts: reduce computation
- Multi-token prediction: accelerate computation

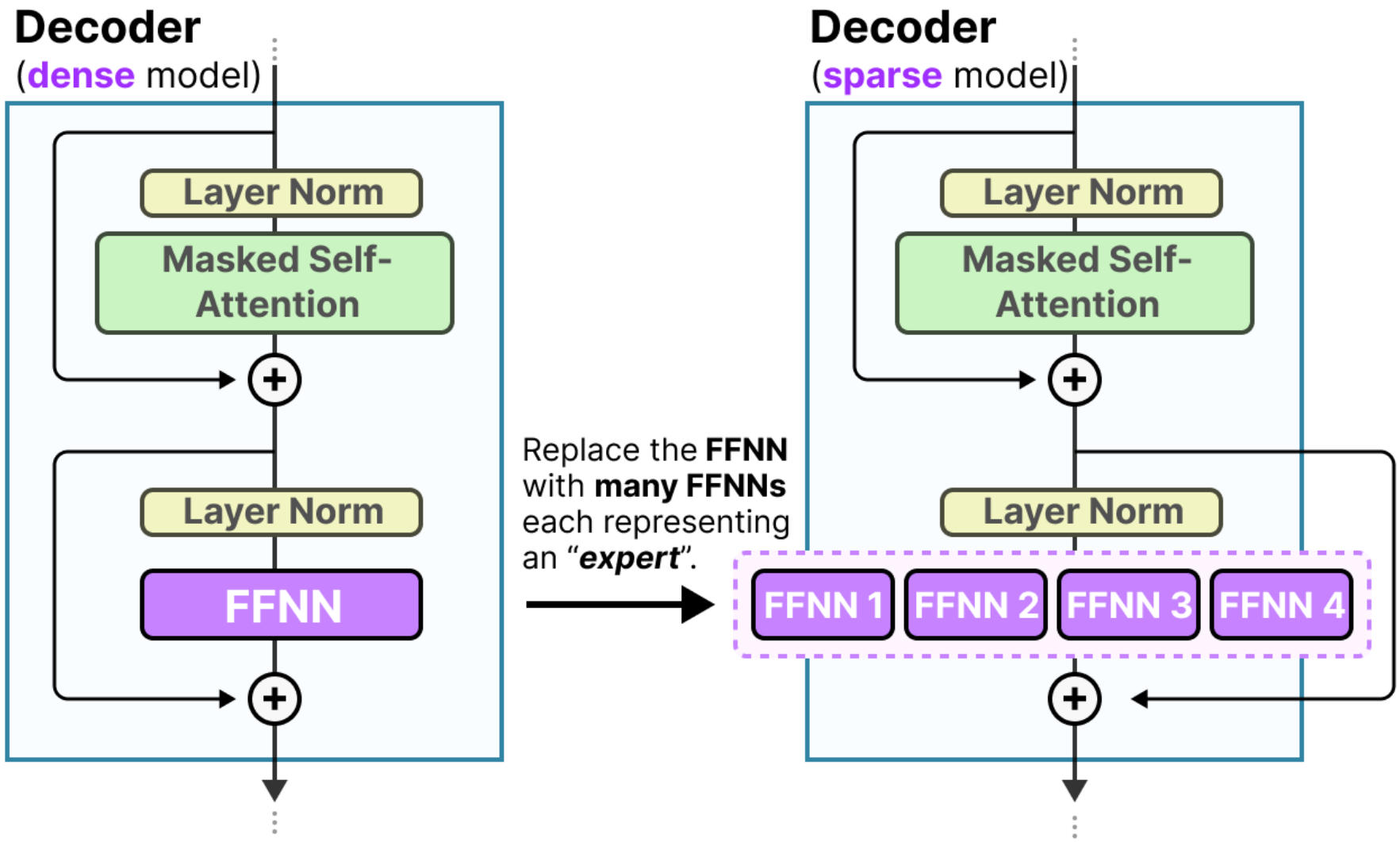
Mixture of Experts

- Can activate only a subset of model parameters at inference time



https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

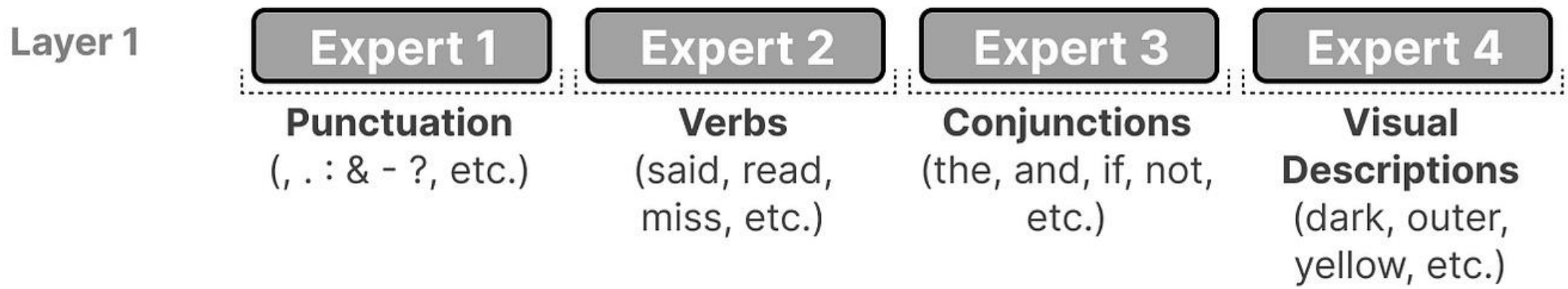
Mixture of Experts: Expert Implementation



https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

Mixture of Experts: Experts

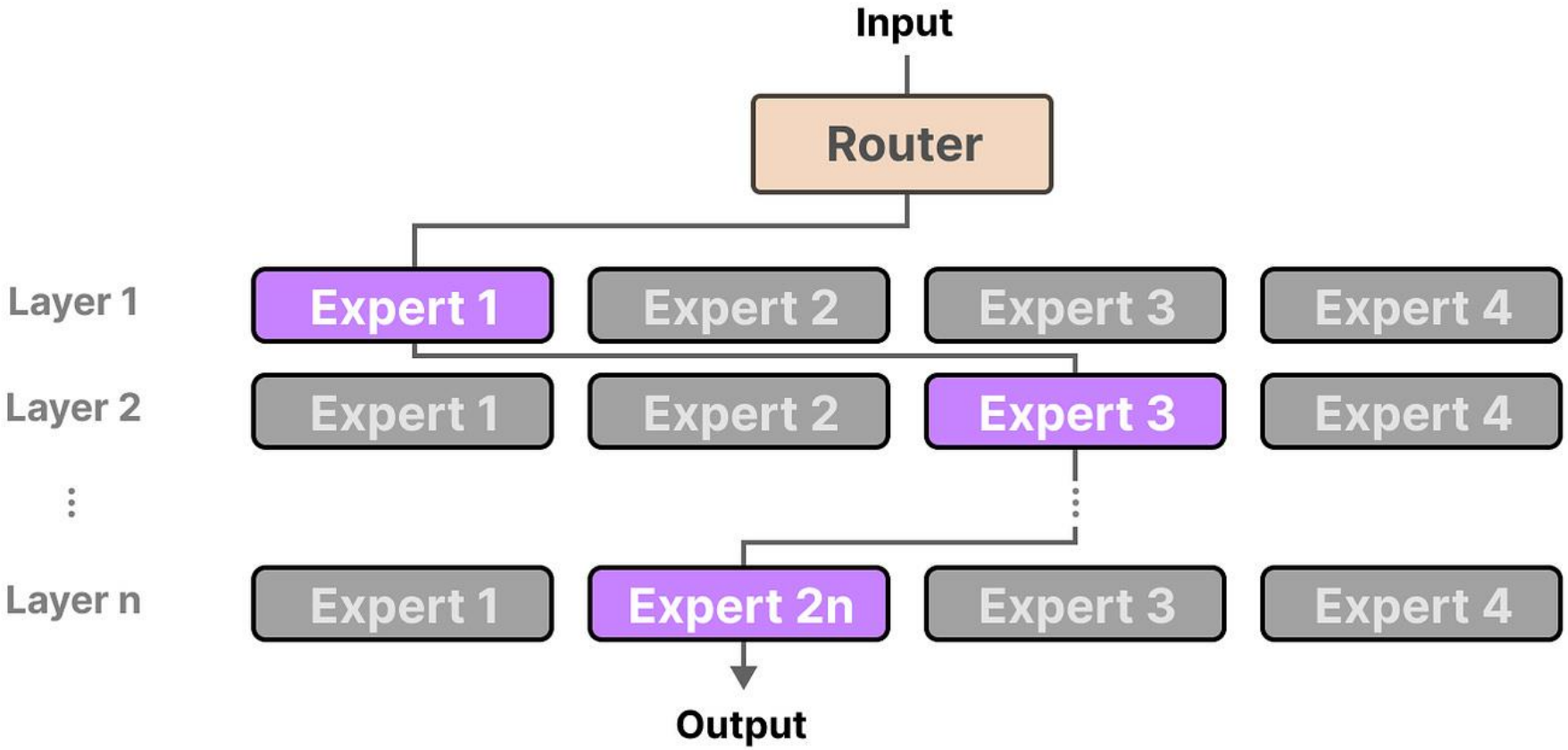
- Focus is on specific tokens, rather than topics (e.g., computer science)



https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

Mixture of Experts: Routers (aka, Gate Network)

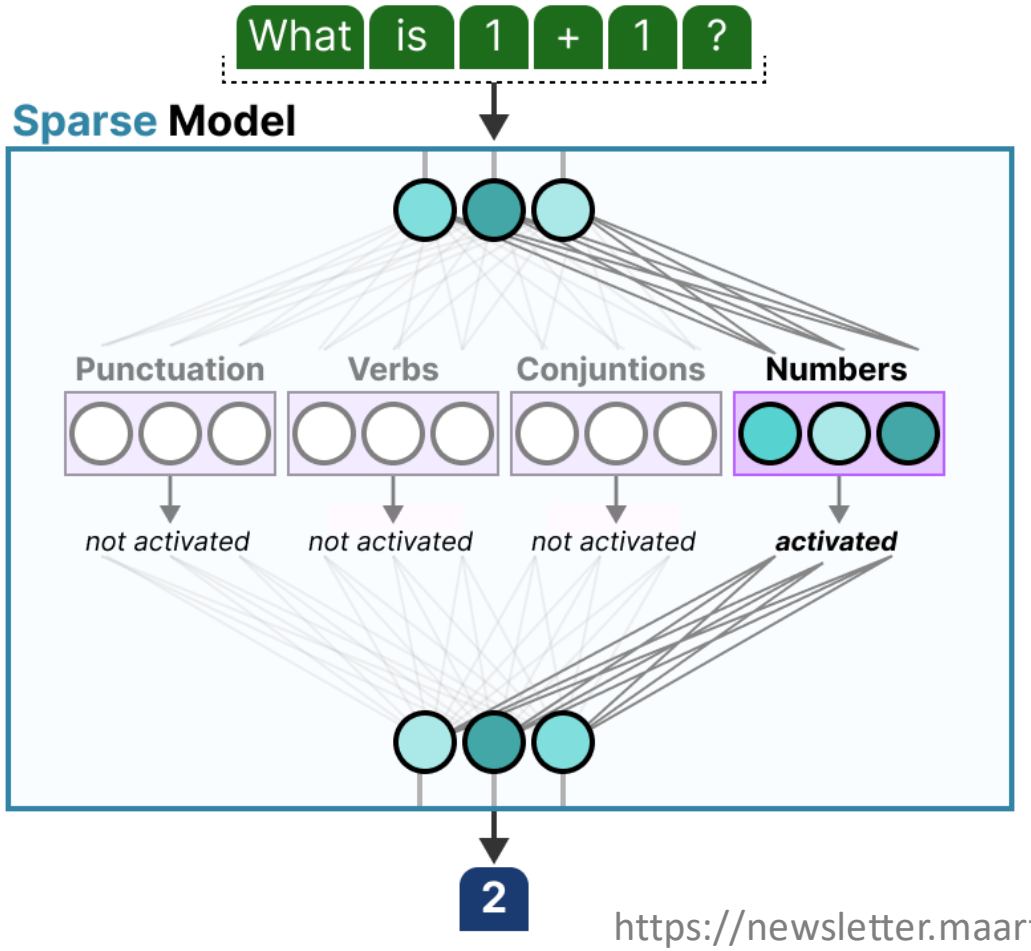
- Given an input, the router selects the best-suited expert(s)



https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

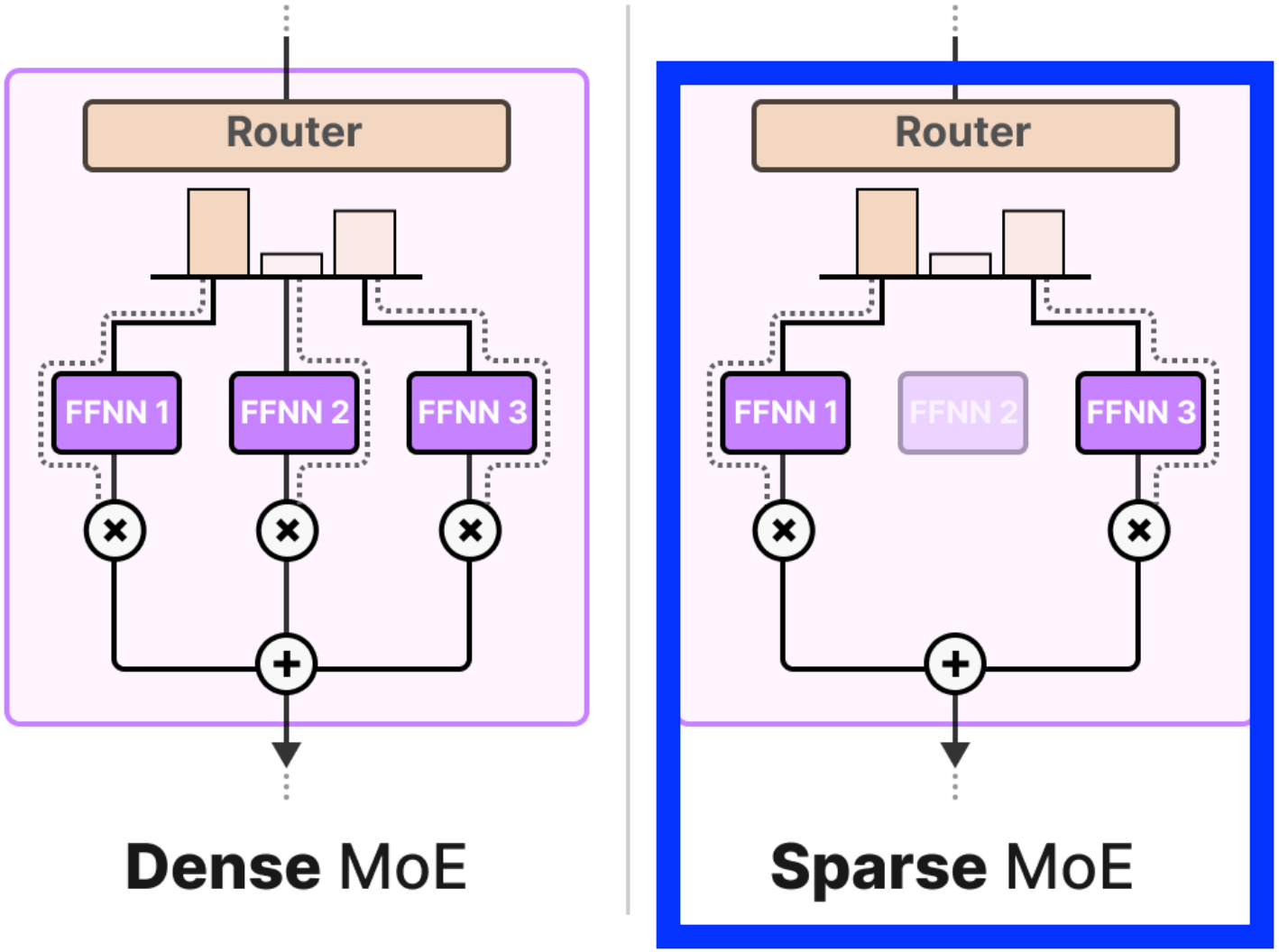
Mixture of Experts: Routers

- Given an input, the router selects the best-suited expert(s); e.g.,



https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

Mixture of Expert: Dense vs Sparse Layer

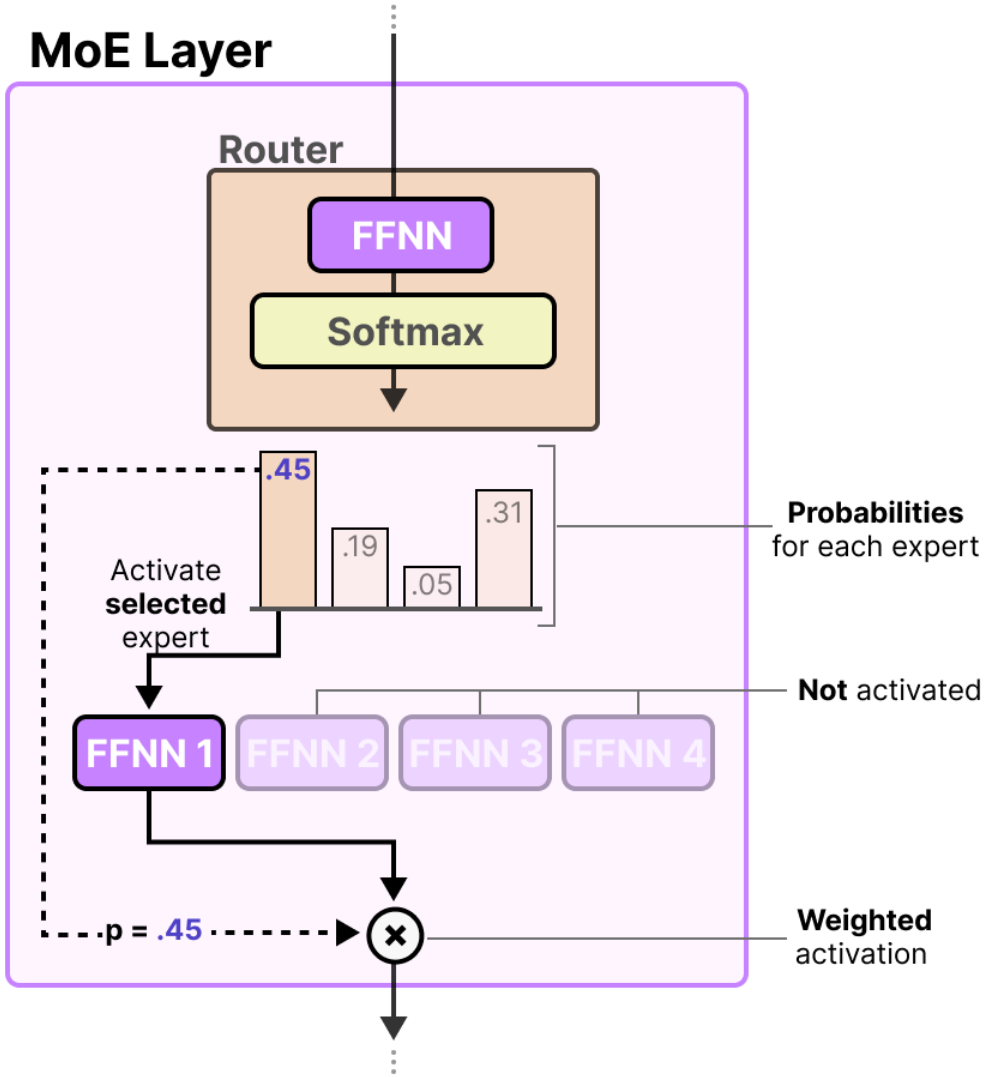


Beneficial for inference time!

Learning challenge: load balancing to apply similar importance to many experts rather than overfitting to few experts

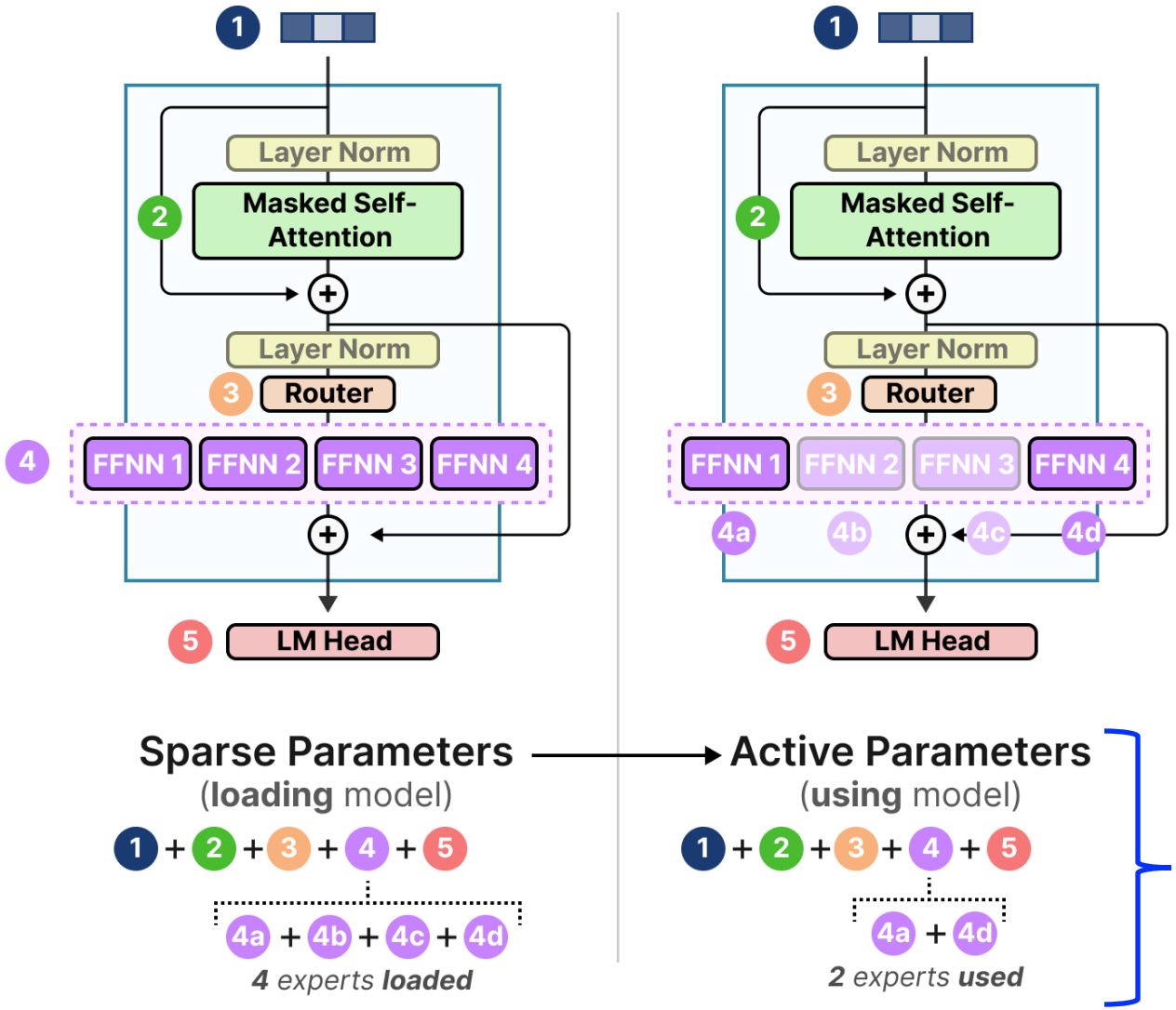
https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

Mixture of Experts: Router Implementation



https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

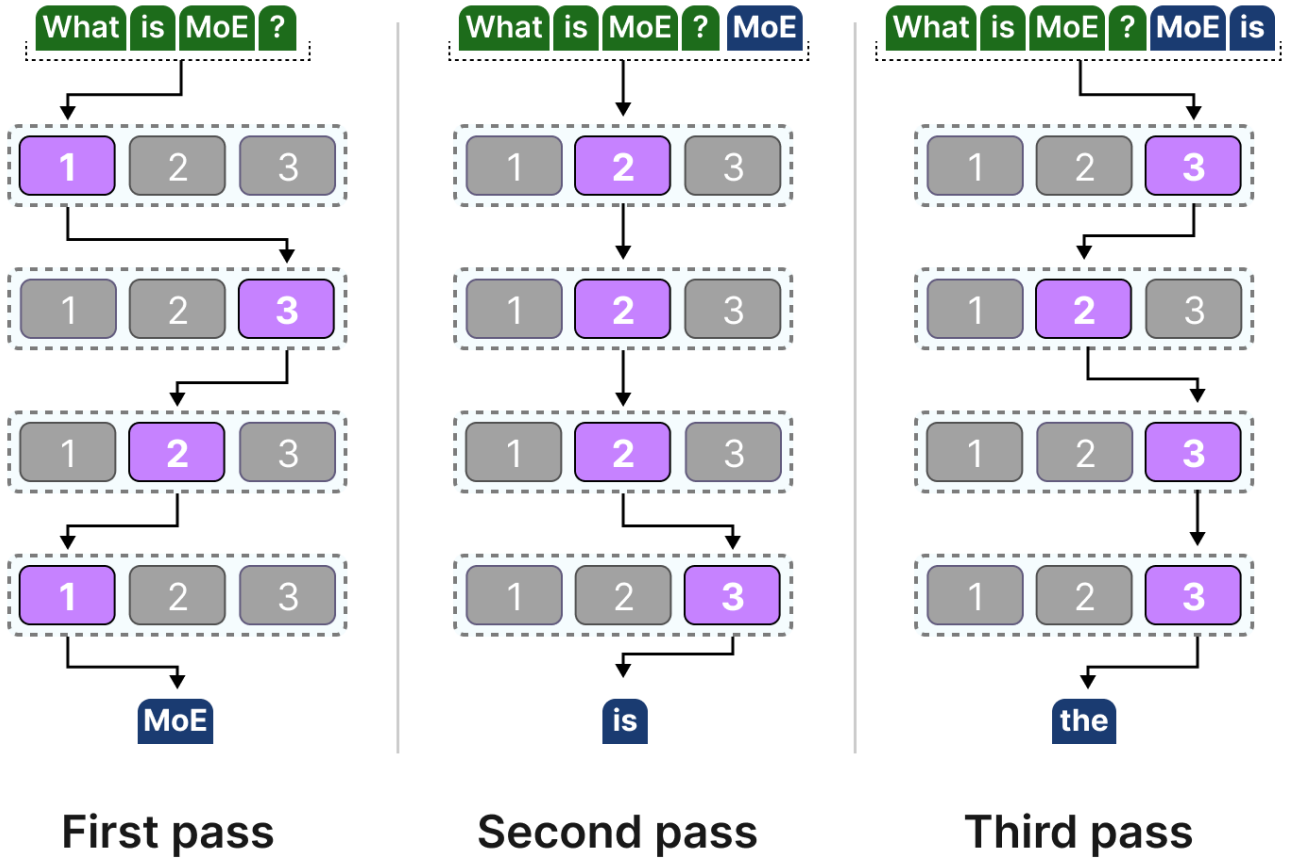
Mixture of Expert: Inference Time Set-Up



https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

Mixture of Experts: Routers

- Paths can differ at different time steps for autoregressive models



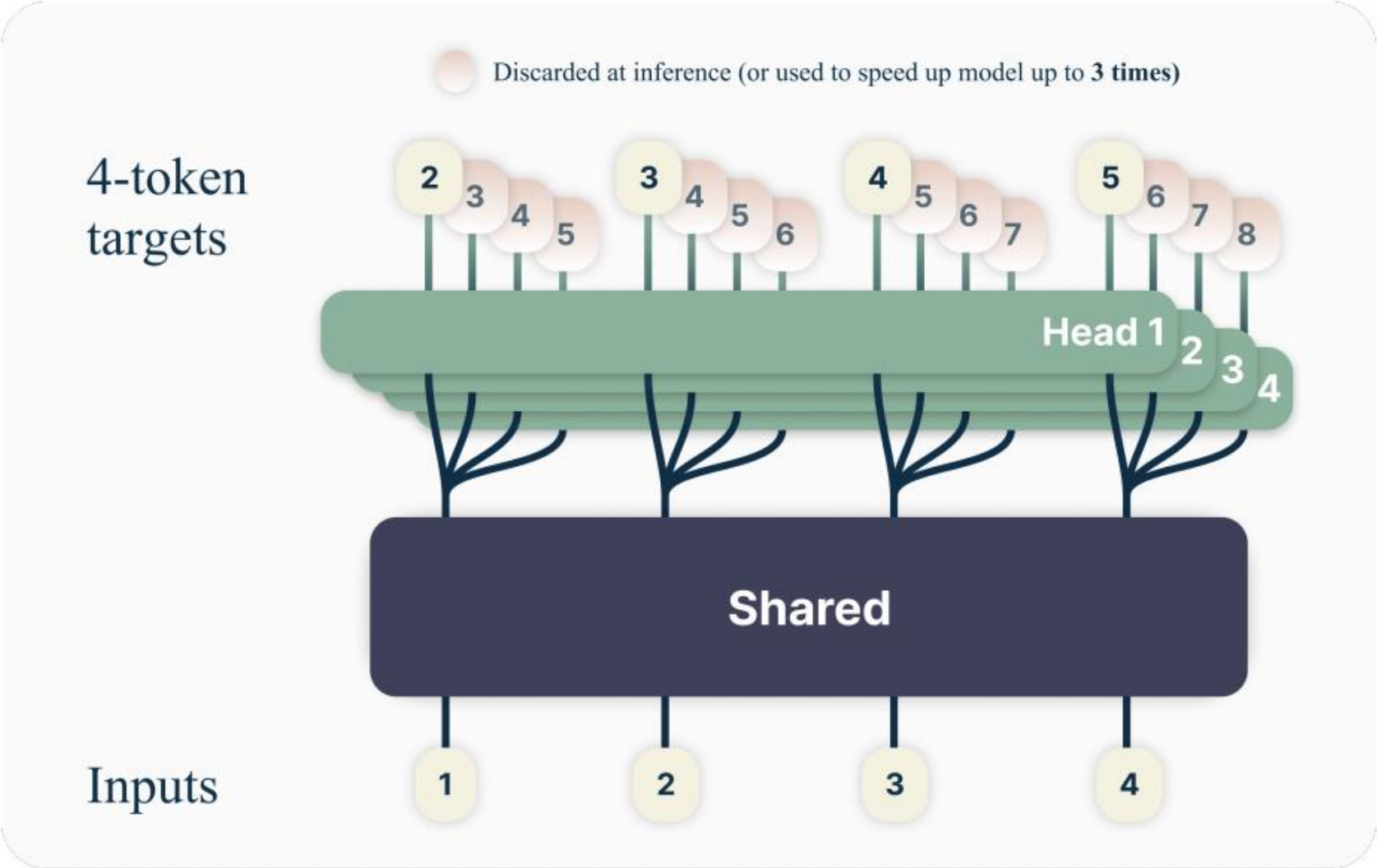
https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

Ideas

- Mixture of experts: reduce computation
- Multi-token prediction: accelerate computation

Multi-Token Prediction

Use *N* output heads to predict *N* tokens at each time step:



Gloeckle et al. Better & Faster Large Language Models via Multi-token Prediction. April 2024

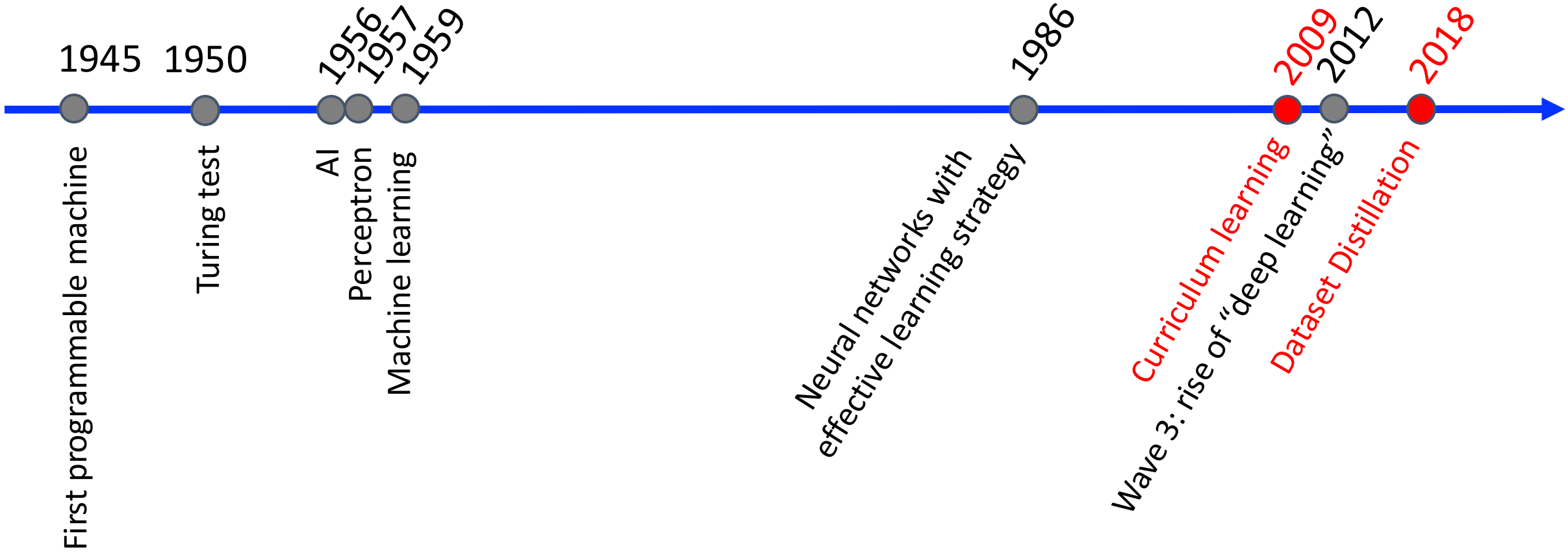
Summary

- Mixture of experts: reduce computation
- Multi-token prediction: accelerate computation

Today’s Topics

- Motivation
- Hardware tricks
- Architectural tricks
- Training tricks
- Programming tutorial

Pioneering Methods: Historical Context

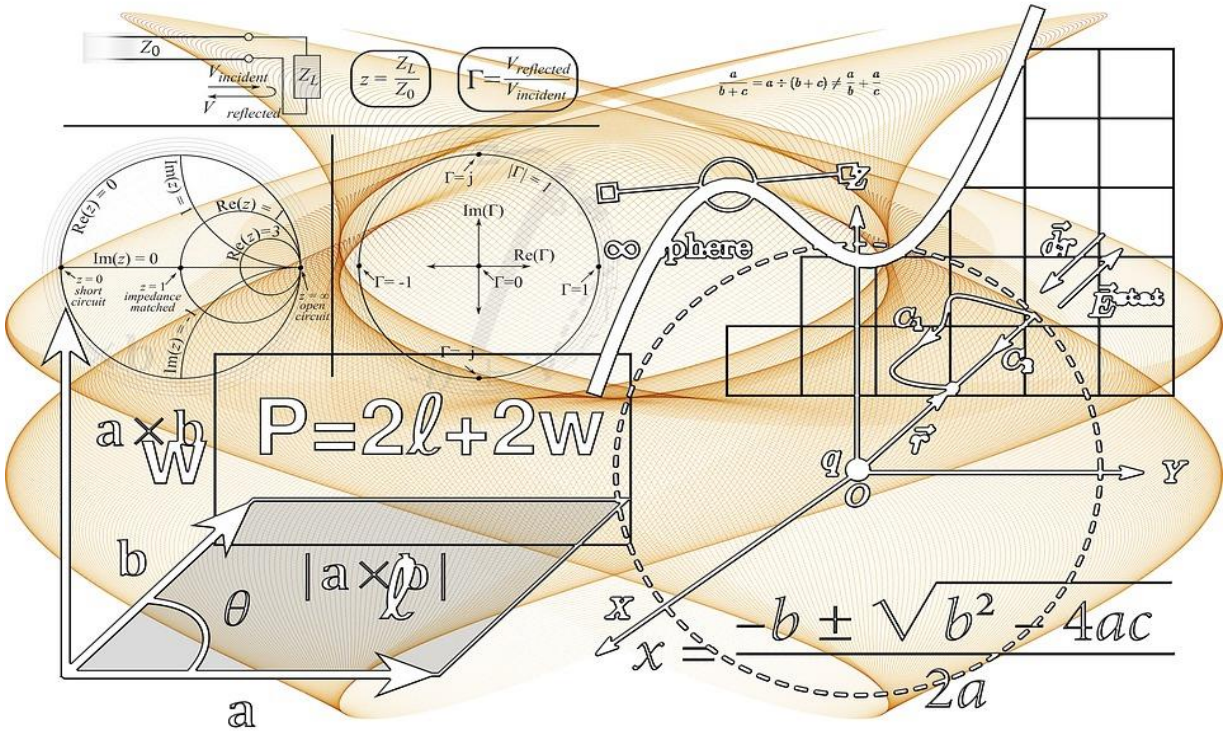


Ideas

- Curriculum learning
- Dataset distillation

Accelerate Learning with a Curriculum

Random Order of Examples



Ordered Examples

Big Book of Math; Dinah Zike

How did you learn math?

Accelerate Learning with a Curriculum

Random Order of Examples

Ordered Examples

How did you learn to read?

Key Questions In Creating a Curriculum

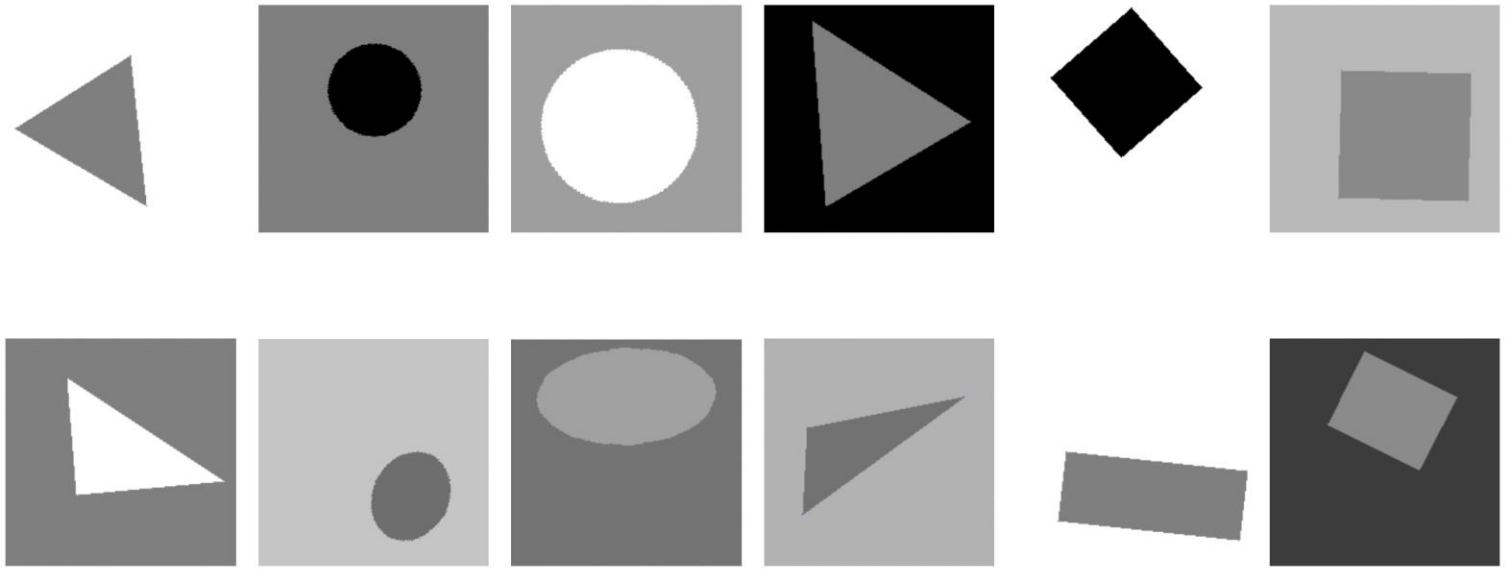
- How many levels to include in the curriculum from easy to hard?
- How to define what is “easier” versus “harder”?

How to Evaluate a Curriculum?

- How good is the model? - generalization performance on test data
- How long did learning take? - training convergence speed

Pioneering Task: Shape Prediction

Classify each shape as rectangle, ellipse, or triangle



Solution: 3-layer neural network

1. Easy (Basic): less shape variability (squares, circles, and equilateral triangles); 10,000 examples
2. Hard (Geom): more shape variability (rectangles, ellipses, and triangles); 10,000 examples

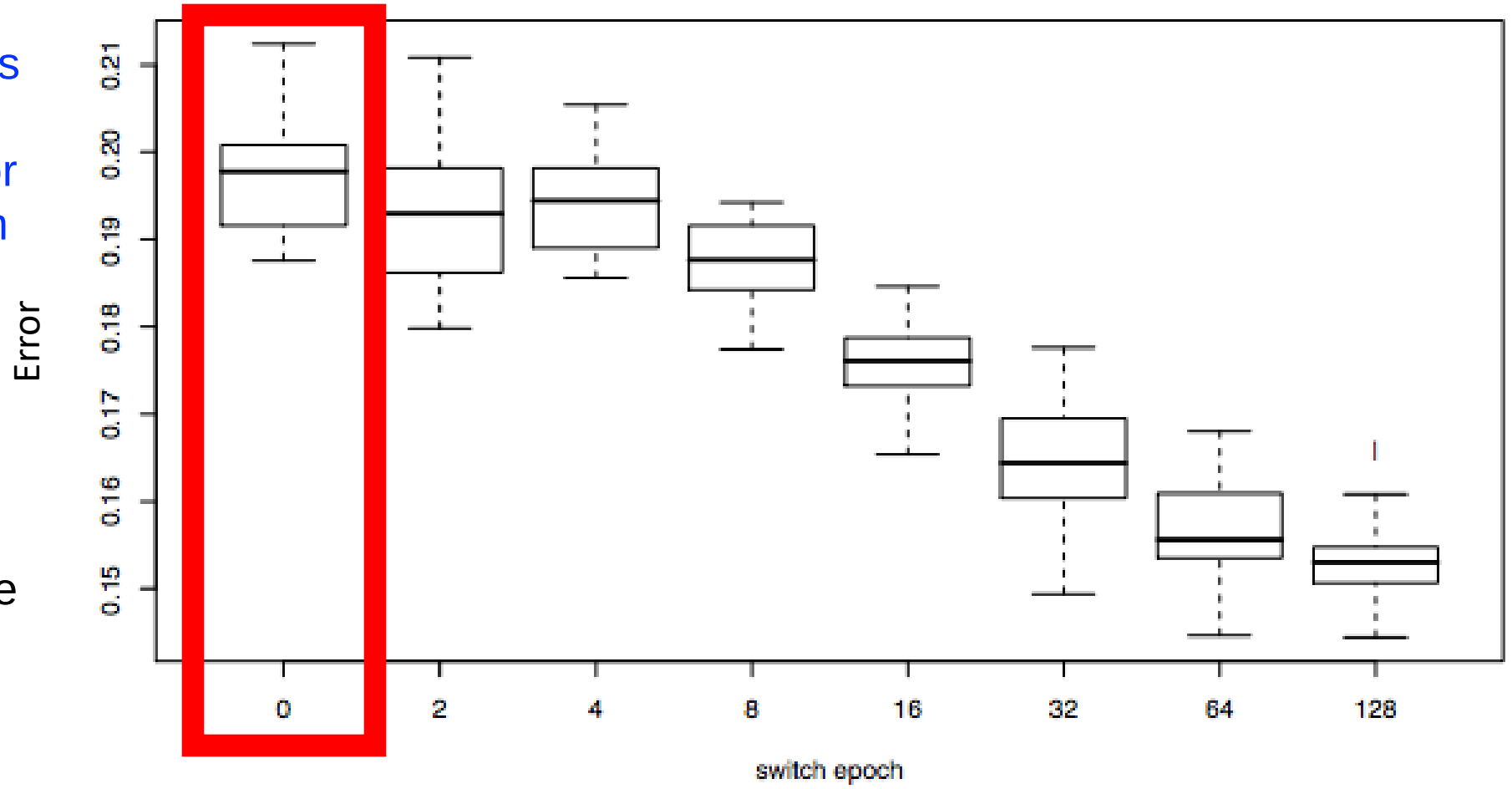
Bengio et al., Curriculum Learning, 2009

Shape Prediction: Curriculum Learning

Results when training on easier examples for *n* epochs and then training on harder examples until 256 epochs or validation error hits minimum (20 random initializations).

What are the benefits of curriculum learning?

How many epochs should we train the model before introducing hard examples?



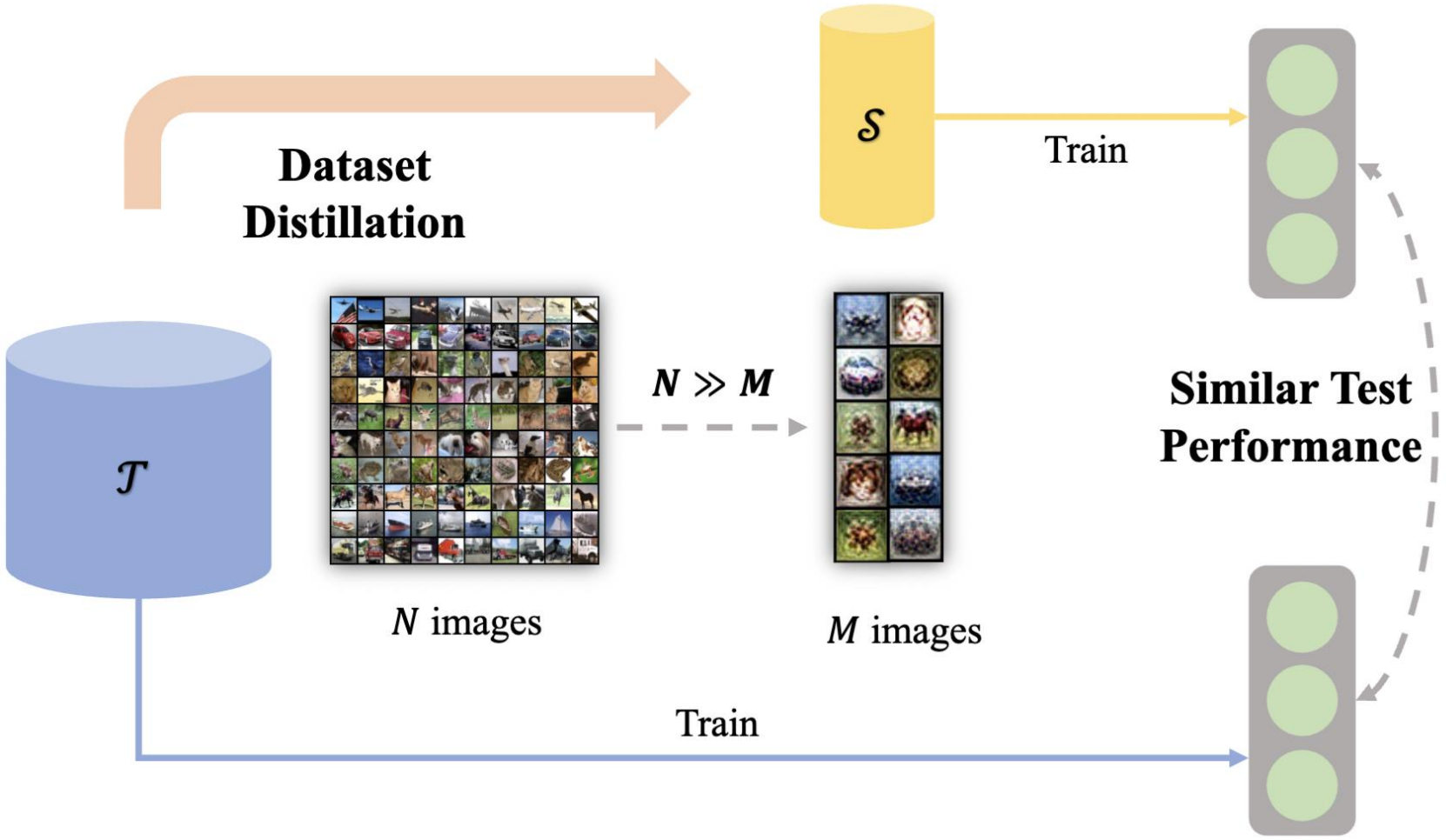
No curriculum

Bengio et al., Curriculum Learning, 2009

Ideas

- Curriculum learning
- Dataset distillation

Distill Large Dataset Into a Small Number of Synthetic Images



Yiu, Lu, and Wang. Dataset Distillation: A Comprehensive Review. PAMI 2023

Pioneering Paper

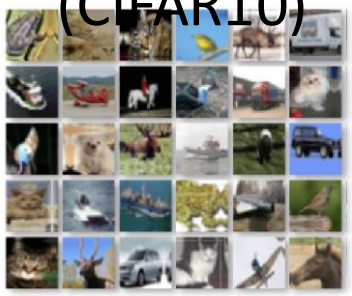
(MNIST)

60K images

distill

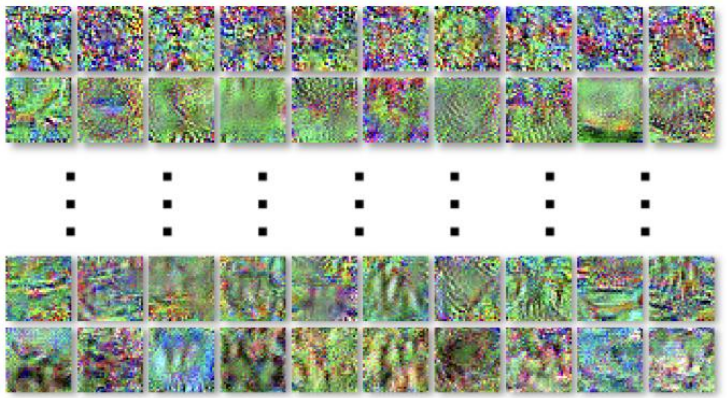
10 images

(CIFAR10)



50K images

distill



100 images

Synthetic training images enable more efficient training (i.e., less training images and so gradient descent steps) while reducing data storage costs and bypassing privacy concerns

Wang et al. Dataset Distillation. arXiv 2018

Typical Approach: Many Optimization Objectives

Algorithm 1: Dataset Distillation Framework

Input: Original dataset $\mathcal{T}$

Output: Synthetic dataset $\mathcal{S}$

Initialize $\mathcal{S}$ ▷ Random, real, or core-set

while *not converge* **do**

  Get a network θ ▷ Random or from some cache

  Update θ and cache it if necessary ▷ Via $\mathcal{S}$ or $\mathcal{T}$, for some steps

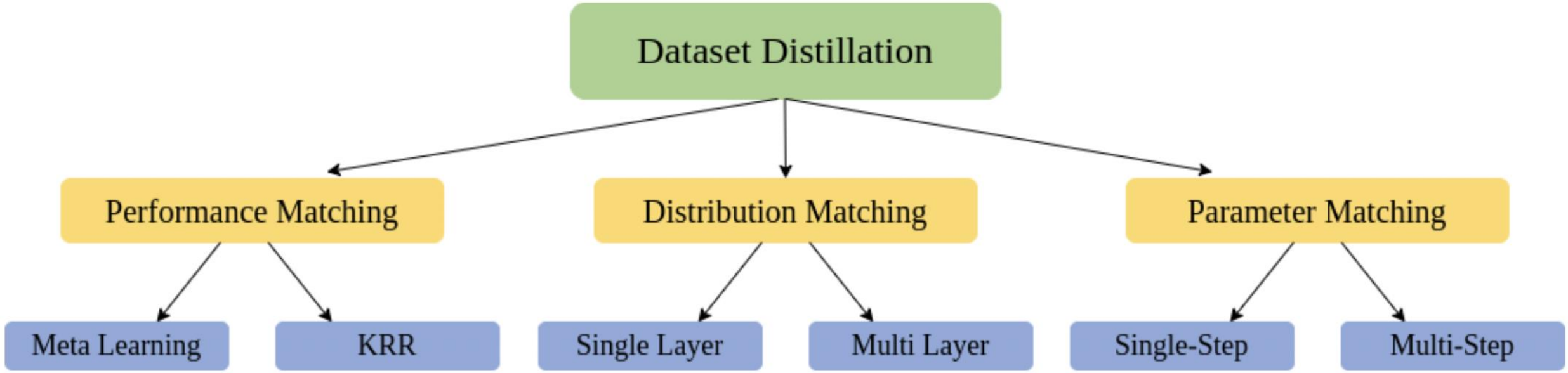
  Update $\mathcal{S}$ via $\mathcal{L}(\mathcal{S}, \mathcal{T})$ ▷ PerM, ParM, DisM, or their variants

end

return $\mathcal{S}$

Yiu, Lu, and Wang. Dataset Distillation: A Comprehensive Review. PAMI 2023

Many Optimization Objectives



e.g., Pioneering 2018 paper "optimize[s] a synthetic dataset such that neural networks trained on it could have the lowest loss on the original dataset"

"key idea… train the same network using synthetic datasets and original datasets for some steps, respectively, and encourage the consistency of their trained neural parameter"

rather than match impacts on training, "obtain synthetic data whose distribution can approximate that of real data"

https://blog.roboflow.com/what-is-dataset-distillation/

Summary

- Curriculum learning
- Dataset distillation

Today’s Topics

- Motivation
- Hardware tricks
- Architectural tricks
- Training tricks
- Programming tutorial

Today’s Topics

- Motivation
- Hardware tricks
- Architectural tricks
- Training tricks
- Programming tutorial


The End