Mistral 7B

Name
박제현
NLP
2024/05/28

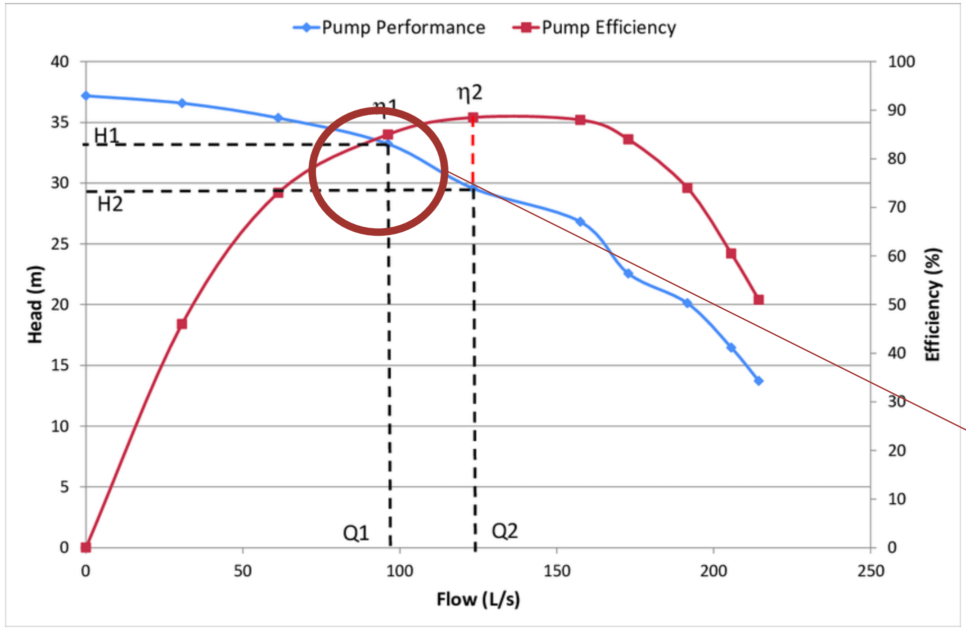
TRAIN AND TEST

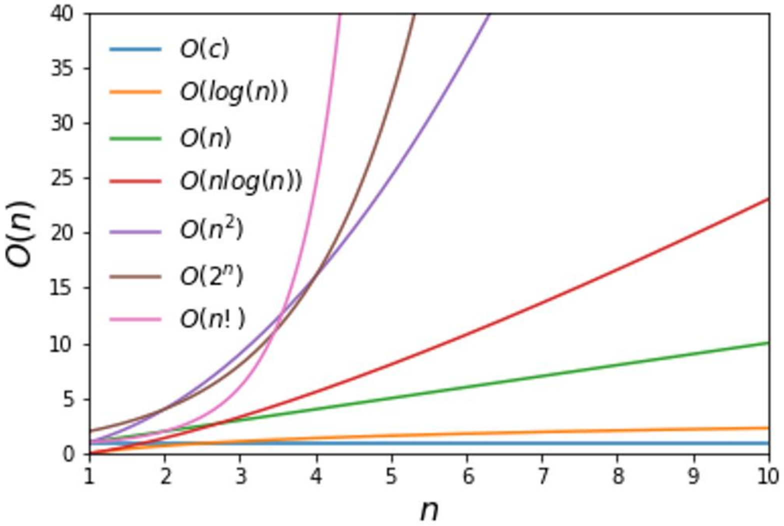
Contents

TRAIN AND TEST

Balancing Performance and Efficiency in NLP Models: Innovations of Mistral 7B

Performance





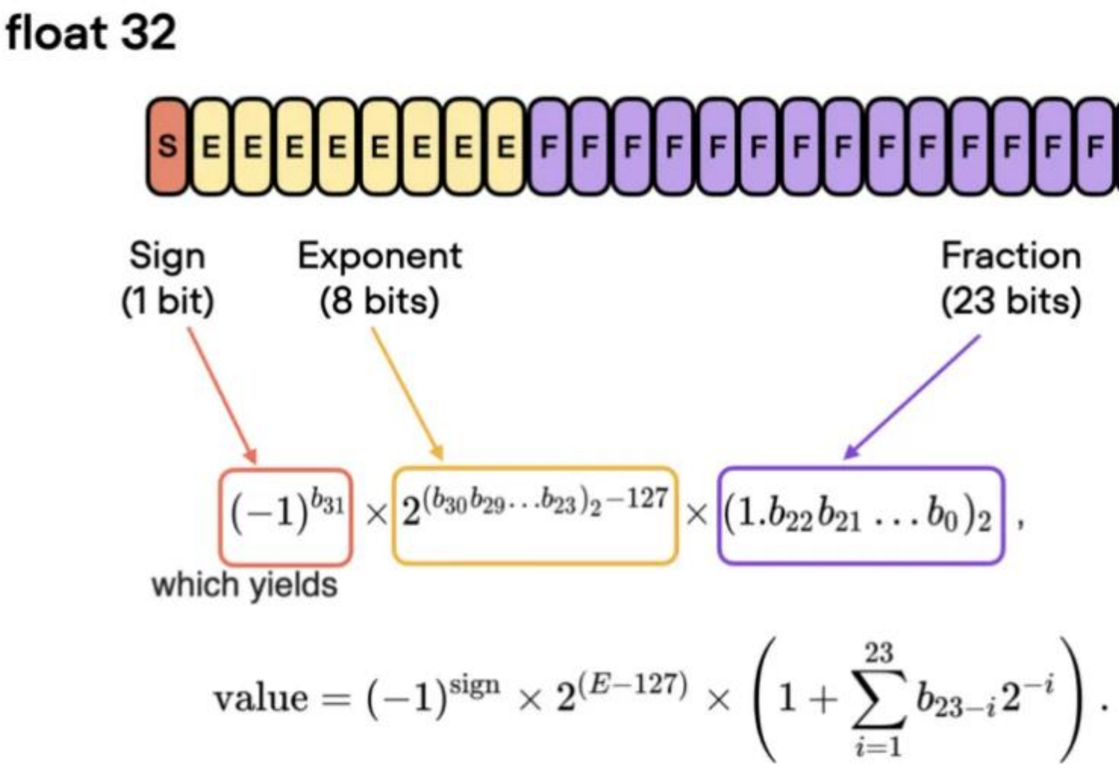
Model Size & Performance
Performance & 1 / Efficiency

- Lower Computational costs!
- Less Inference latency! -> LoRA

TNT TRAIN AND TEST

Recent Appoaches – LLMs Optimization

Floating point



1. 32-bit precision requires twice as much GPU memory as **16-bit precision**, allowing more efficient use of GPU memory.

2. Since operations on **lower precision tensors require less memory**, GPUs can process them more quickly.

Recent Appoaches – LLMs Optimization

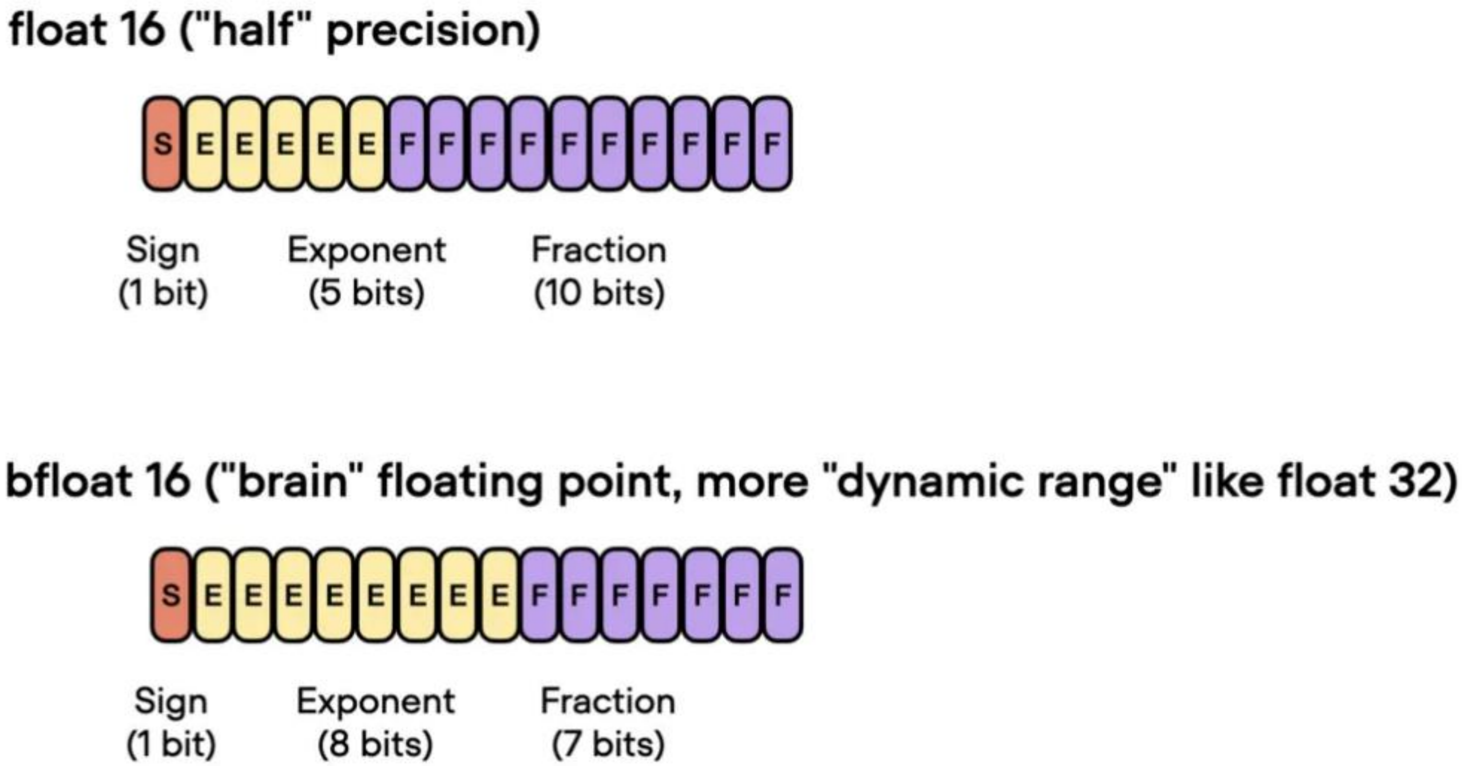
Floating point

Don't transfer all parameters and operations to 16-bit floats. Instead, we **switch between** 32-bit and 16-bit operations during training

TRAIN AND TEST

Recent Appoaches – LLMs Optimization

Floating point

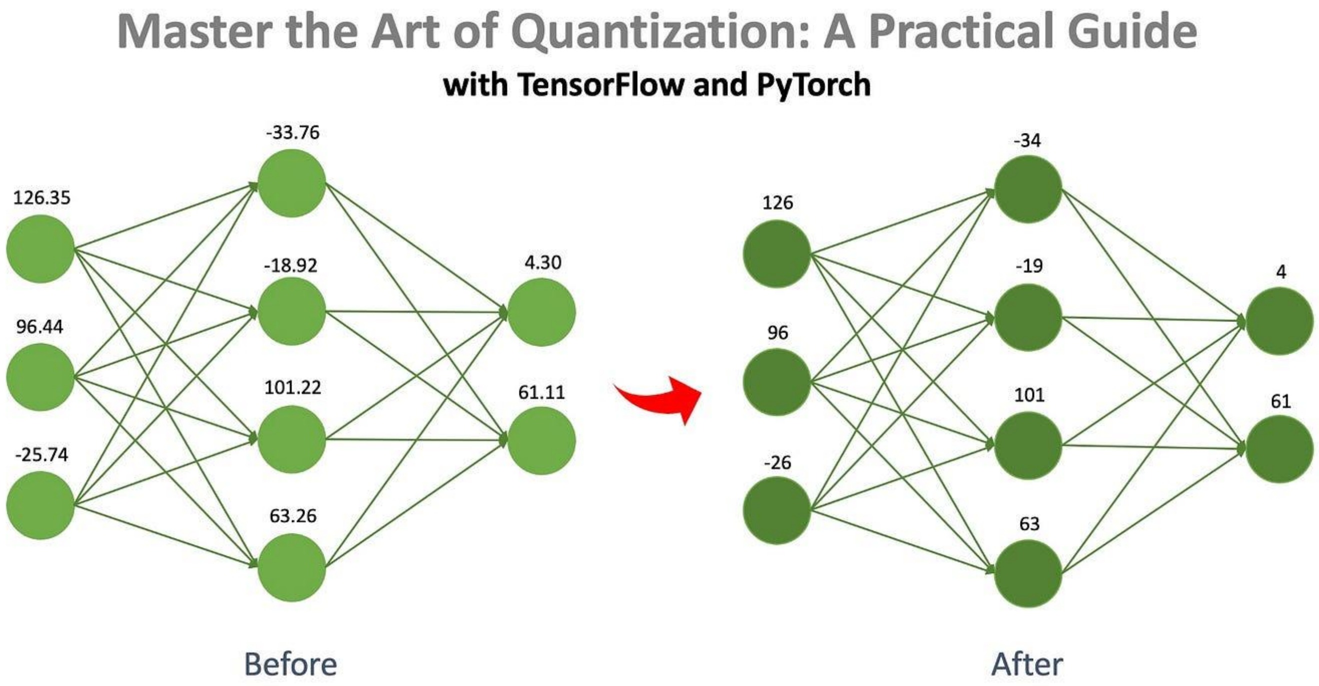


Google developed this format for machine learning and deep learning applications, particularly in their **Tensor Processing Units** (TPUs). While bfloat16 was originally developed for TPUs, this format is now supported by **several NVIDIA GPUs**.

TRAIN AND TEST

Recent Appoaches – LLMs Optimization

Quantization



Quantization converts the model weights from **float32** to **low-bit integer** representations, for example, **8-bit integers** (and, recently, even **4-bit integers**).

TRAIN AND TEST

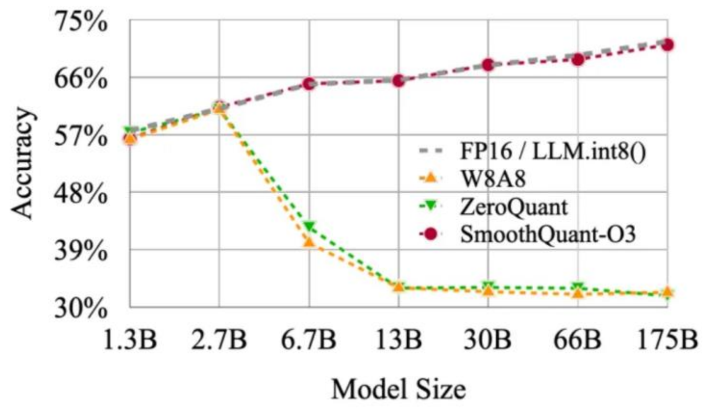
Recent Appoaches – LLMs Optimization

Quantization

1. Post-Training Quantization (PTQ): A model is **first trained to converge, then we convert its weights** to a lower precision without more training. It is usually quite cheap to implement in comparison to training.

2. Quantization-Aware Training (QAT): Quantization is applied **during pre-training or further fine-tuning**. QAT can perform better but requires extra computation resources and access to representative training data

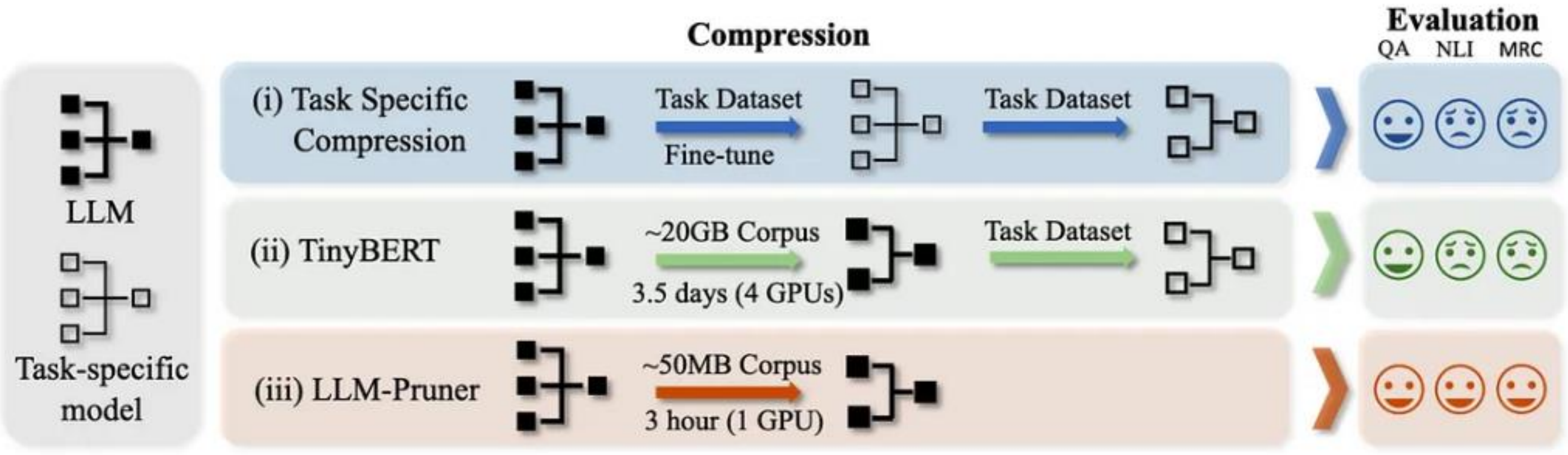
- But Both can lower Performance



TRAIN AND TEST

Recent Appoaches – LLMs Optimization

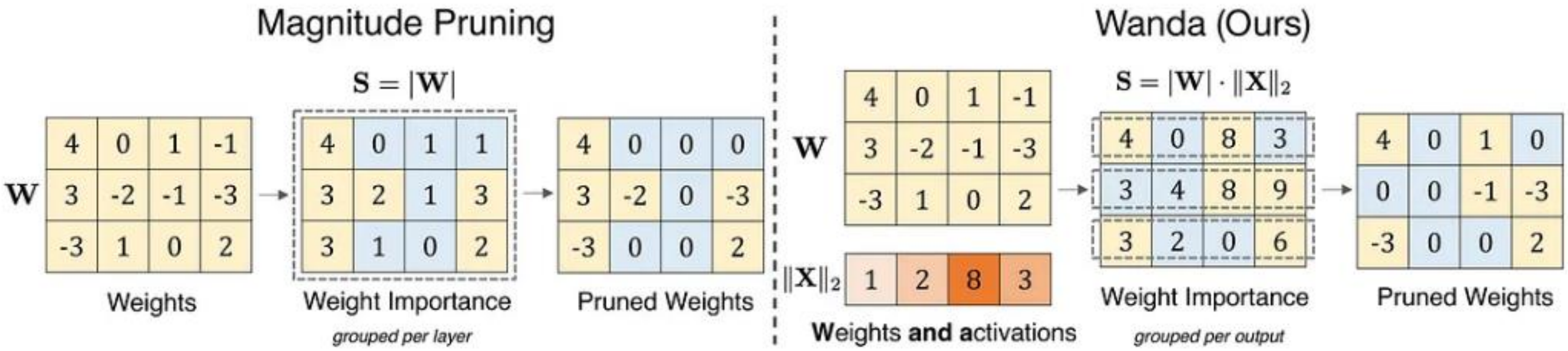
Pruning



Adopts **structural pruning** that selectively removes **non-critical coupled structures** based on **gradient information**, maximally preserving most of the LLM's functionality.

Recent Appoaches – LLMs Optimization

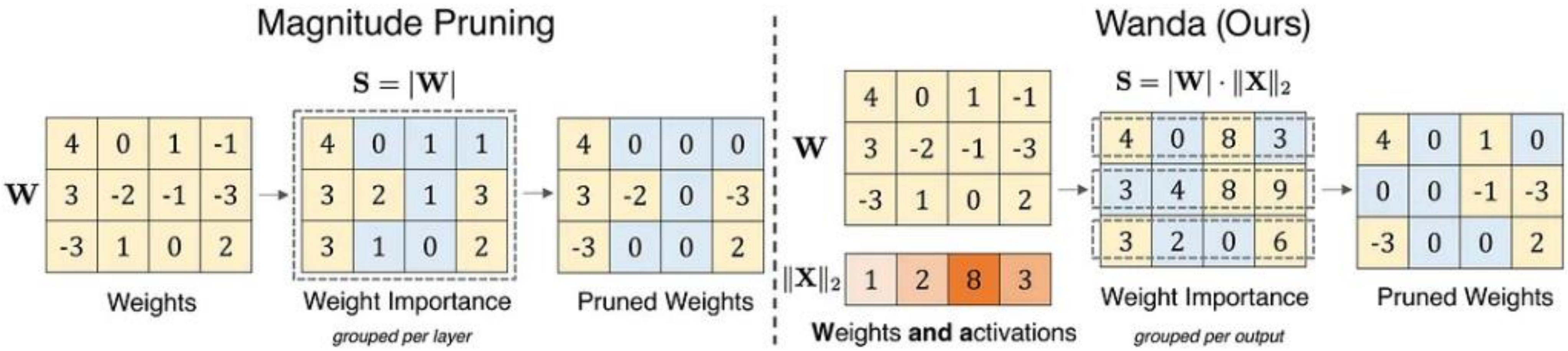
Pruning



Compared to **magnitude pruning** which removes weights **solely based on their magnitudes**, Wanda removes weights on a **per-output basis** by the **product of weight magnitudes** and **input activation norms**.

Balancing Performance and Efficiency in NLP Models: Innovations of Mistral 7B

Pruning



Compared to **magnitude pruning** which removes weights **solely based on their magnitudes**, Wanda removes weights on a **per-output basis** by the **product of weight magnitudes** and **input activation norms**.

TRAIN AND TEST

Technical Innovations of Mistral 7B

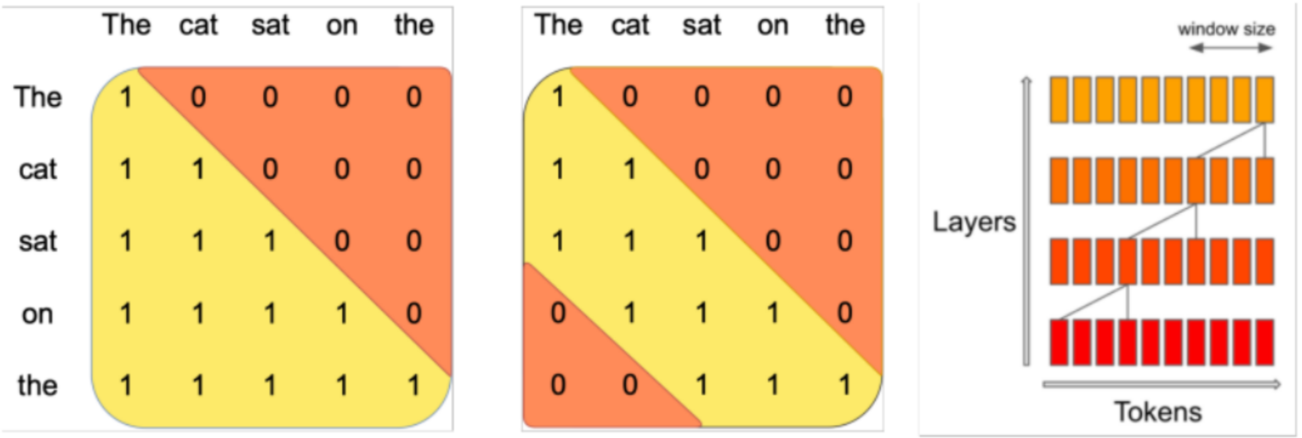
Fine-tuning

Grouped-Query Attention (GQA):
- Acceleration of inference speed.
- Reduction of memory requirements and increase in batch size.

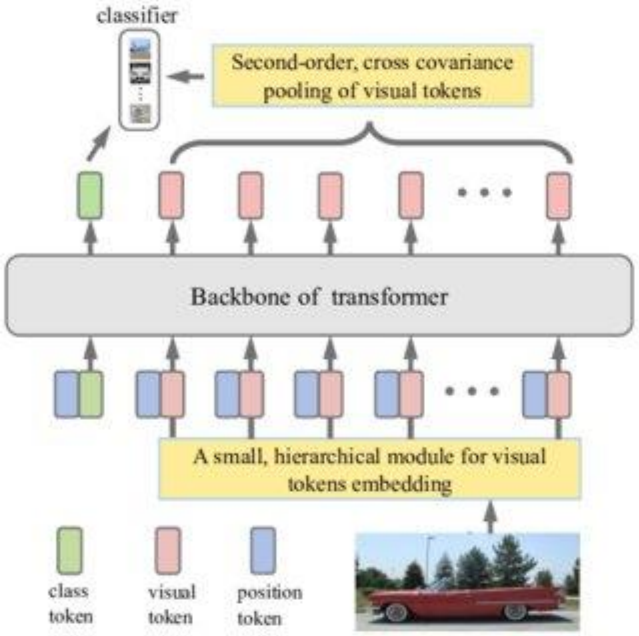
Sliding Window Attention (SWA):
- Improved efficiency in handling long sequences.
- Reduction of computational costs.

Sliding Window Attention

Sliding Window Attention

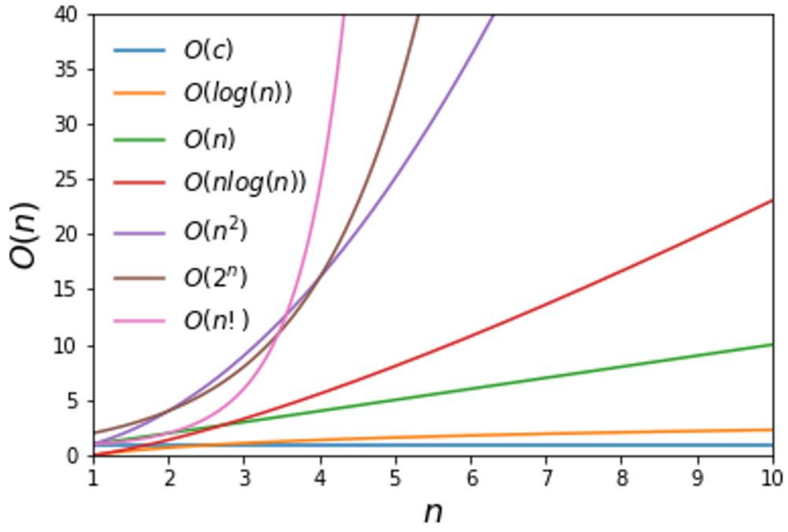


Vanilla Attention Sliding Window Attention Effective Context Length



Sliding Window Attention

Sliding Window Attention



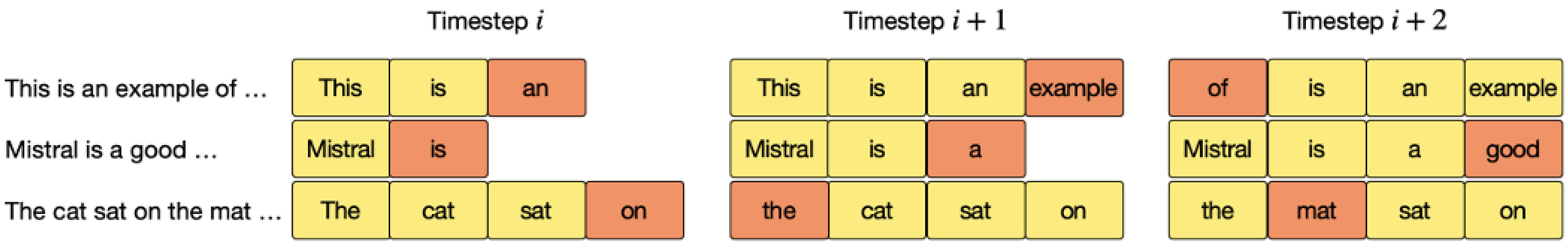
Vanilla Attention:
Computational complexity: **O(n^2)**, where n is the sequence length.
Memory usage: Increases **linearly** with the number of tokens.
Doubling the sequence length **quadruples the computation**.

Sliding Window Attention (SWA):
Computational complexity: **O(n * W)**, where W is the window size.
Memory usage: Increases **linearly** with W.
The computational load is **constrained by the window size**, even as the sequence length increases.

TRAIN AND TEST

Rolling Buffer Cache

Rolling Buffer Cache



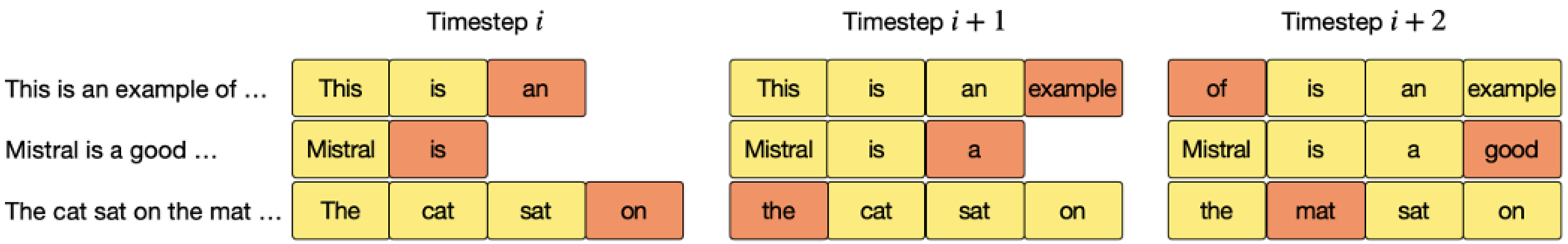
The attention span is **limited to a fixed size** W.
This means each token will only attend to W **preceding tokens**.

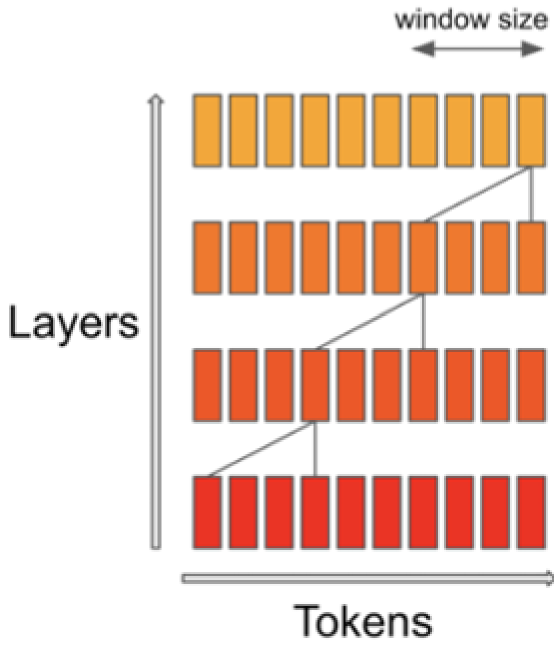
The cache is organized as a rolling buffer with a fixed size W.
Keys and values for each timestep i are stored in the cache position i **mod** W. (**i = 0, 1, 2, 3, 4 . . .**)

TRAIN AND TEST

Rolling Buffer Cache

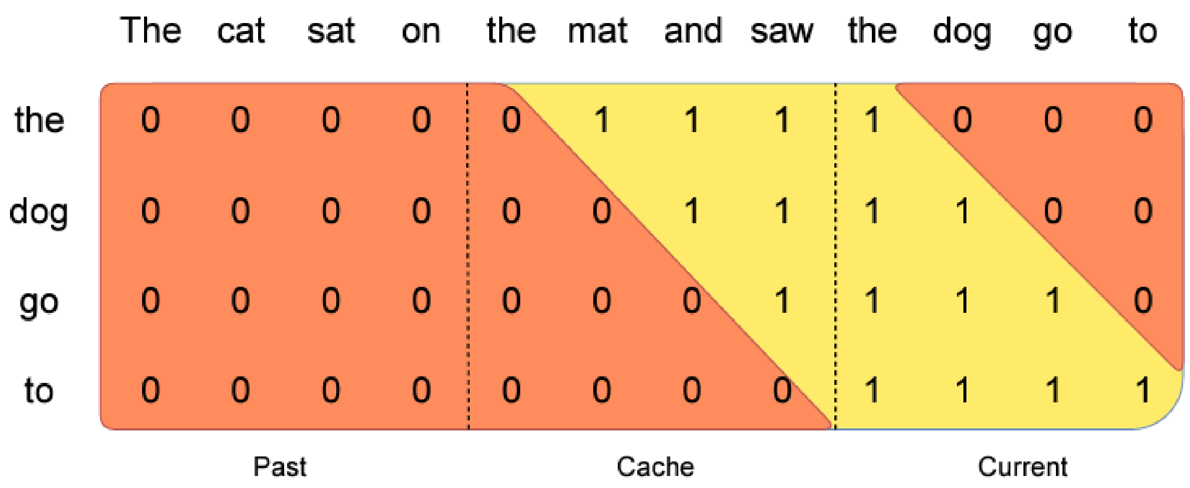
Rolling Buffer Cache





Pre-fill and Chunking

Pre-fill and Chunking

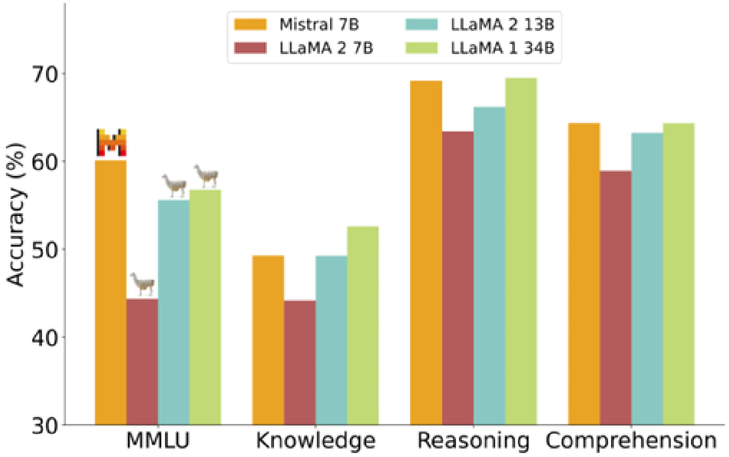


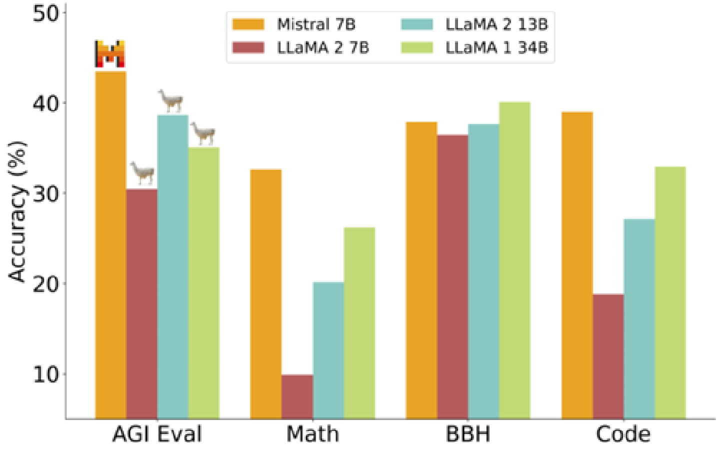
the	mat	and	saw	the	dog	go	to
the	1	1	1	1	dog	go	to
the	mat	and	saw	the	dog	go	to
the	mat	1	1	1	1	go	to
the	mat	and	saw	the	dog	go	to
the	mat	and	1	1	1	1	to
the	mat	and	saw	the	dog	go	to
the	mat	and	saw	1	1	1	1

TRAIN AND TEST

Main resilts & Analysis

Main resilts & Analysis

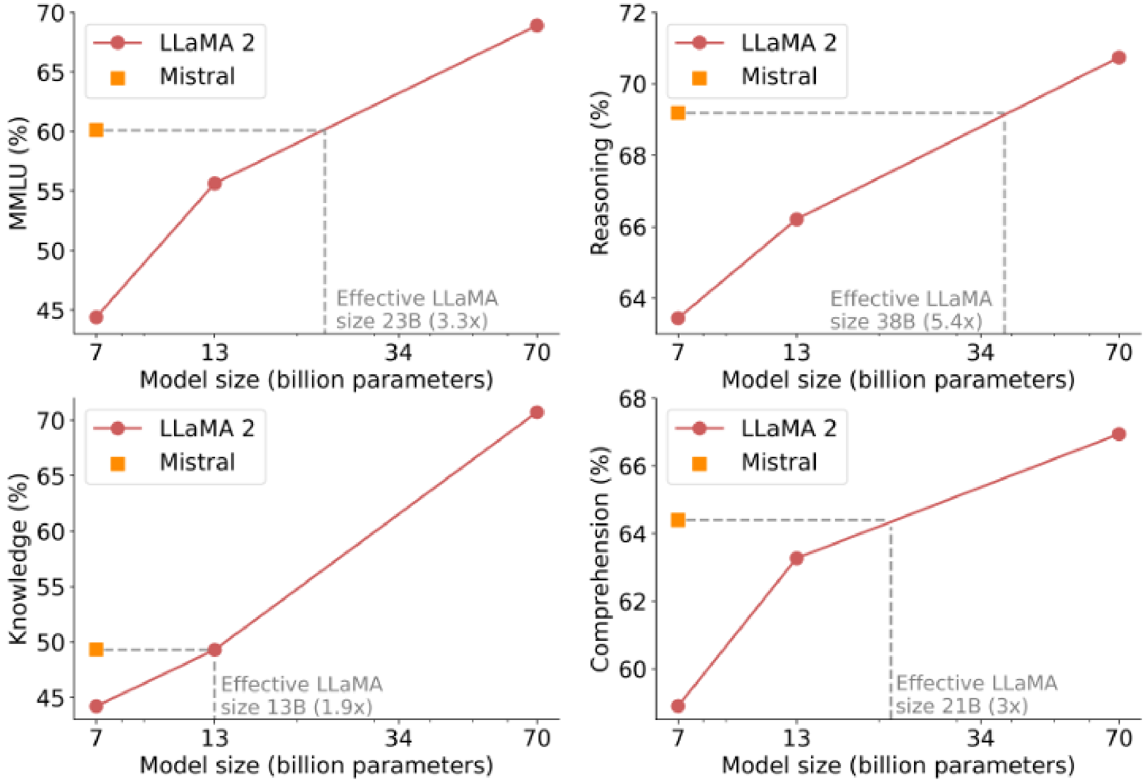




Model	Modality	MMLU	HellaSwag	WinoG	PIQA	Arc-e	Arc-c	NQ	TriviaQA	HumanEval	MBPP	MATH	GSM8K
LLaMA 2 7B	Pretrained	44.4%	77.1%	69.5%	77.9%	68.7%	43.2%	24.7%	63.8%	11.6%	26.1%	3.9%	16.0%
LLaMA 2 13B	Pretrained	55.6%	**80.7%**	72.9%	80.8%	75.2%	48.8%	**29.0%**	**69.6%**	18.9%	35.4%	6.0%	34.3%
Code-Llama 7B	Finetuned	36.9%	62.9%	62.3%	72.8%	59.4%	34.5%	11.0%	34.9%	**31.1%**	**52.5%**	5.2%	20.8%
Mistral 7B	Pretrained	**60.1%**	**81.3%**	**75.3%**	**83.0%**	**80.0%**	**55.5%**	**28.8%**	**69.9%**	**30.5%**	47.5%	**13.1%**	**52.2%**

Main resilts & Analysis

Main resilts & Analysis



TRAIN AND TEST

TNT

TRAIN AND TEST