Fine-Tuning and Instruction Tuning of Llama 3.1 8B for Astronomy

Milad Nourizade

7 January 2025

Project Overview

- Objective:
  - Adapt a pre-trained LLM to the domain of astronomy using continued pretraining and instruction fine-tuning.
- Constrains:
  - Should fit into 15 GB of RAM (free Google Colaboratory T4 GPU)

Approach Overview

- At each step:
  - 4-bit quantization
  - LoRA fine-tuning
  - Merge LoRA adapters + model

Why Llama 3.1 8B?

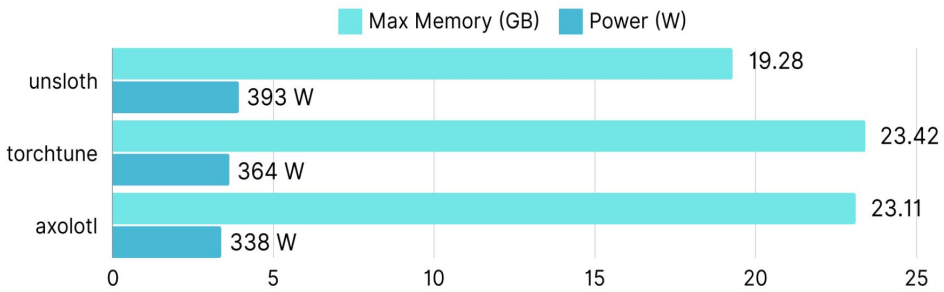
- Strong general capabilities
- Superior astronomical recall (score = 73.7%)
- Resource efficient

Model	Score
Claude-3.5-Sonnet	**85.0**
LLaMA-3.1-405B	83.8
LLaMA-3.1-70B	80.1
Mixtral-8x22B-v0.1	77.7
Gemma-2-27B	75.3
LLaMA-3.2-11B	74.9
LLaMA-3.1-8B	73.7

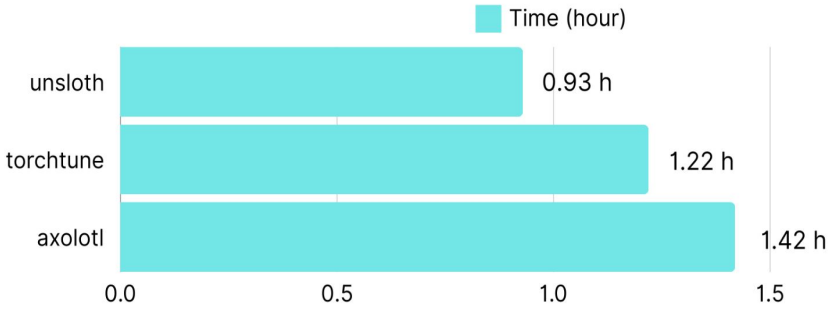
“AstroMLab 1: Who Wins Astronomy Jeopardy!?” (July 2024)

Unsloth

- 2.2x faster, uses 70% less VRAM
- Currently only supports one-GPU fine tuning
- Limited available models
- Supports only NVIDIA GPUs since 2018+.
- Compatibility with Hugging Face, PEFT and, BNB



*tested on llama 3 8B on RTX 4090
source: wandb.ai

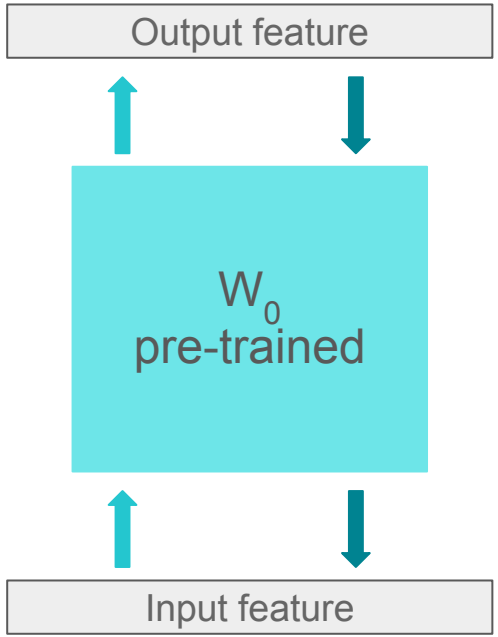


*tested on llama 3 8B on RTX 4090
source: wandb.ai

“Trainer performance comparison: torchtune vs. axolotl vs. Unsloth” (June 2024)

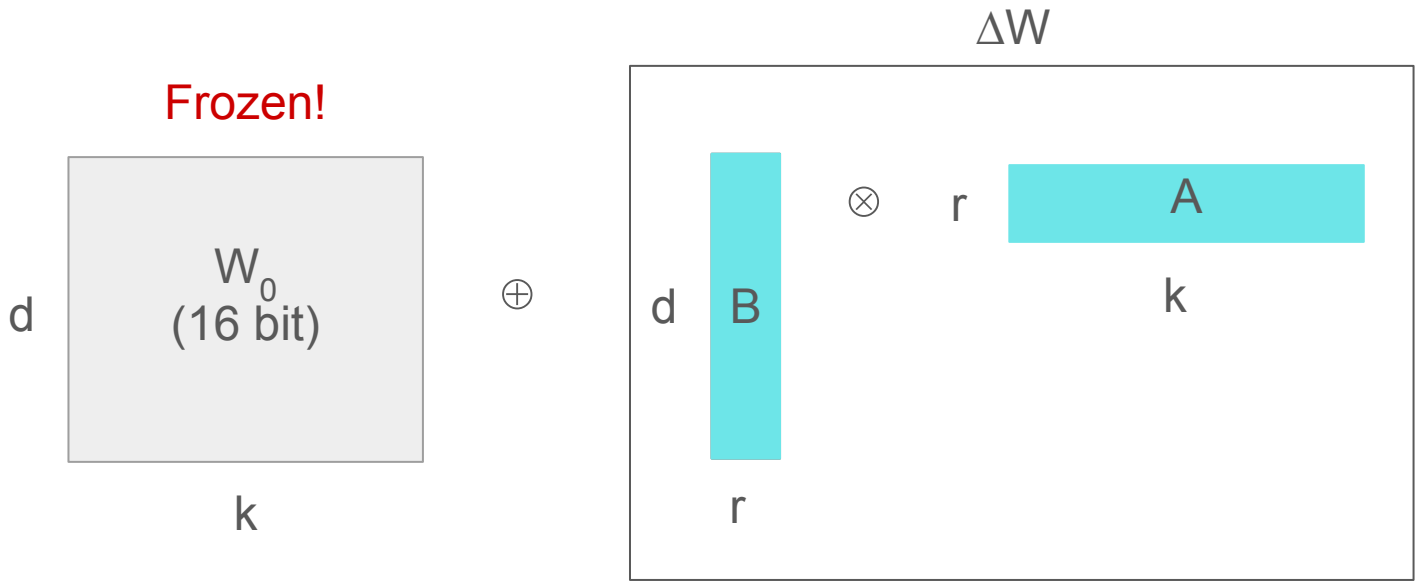
Full Fine-tuning

- Adam optimizer
  - For each parameter: 2 bytes
    - Gradient: 2 bytes
    - Momentum term: 4 bytes
    - Adaptive term: 4 bytes
    - Parameter copy: 4 bytes
- Total Memory= Parameter + Gradient +Optimizer
- Llama 3.1 8B ≈ 128 GB VRAM



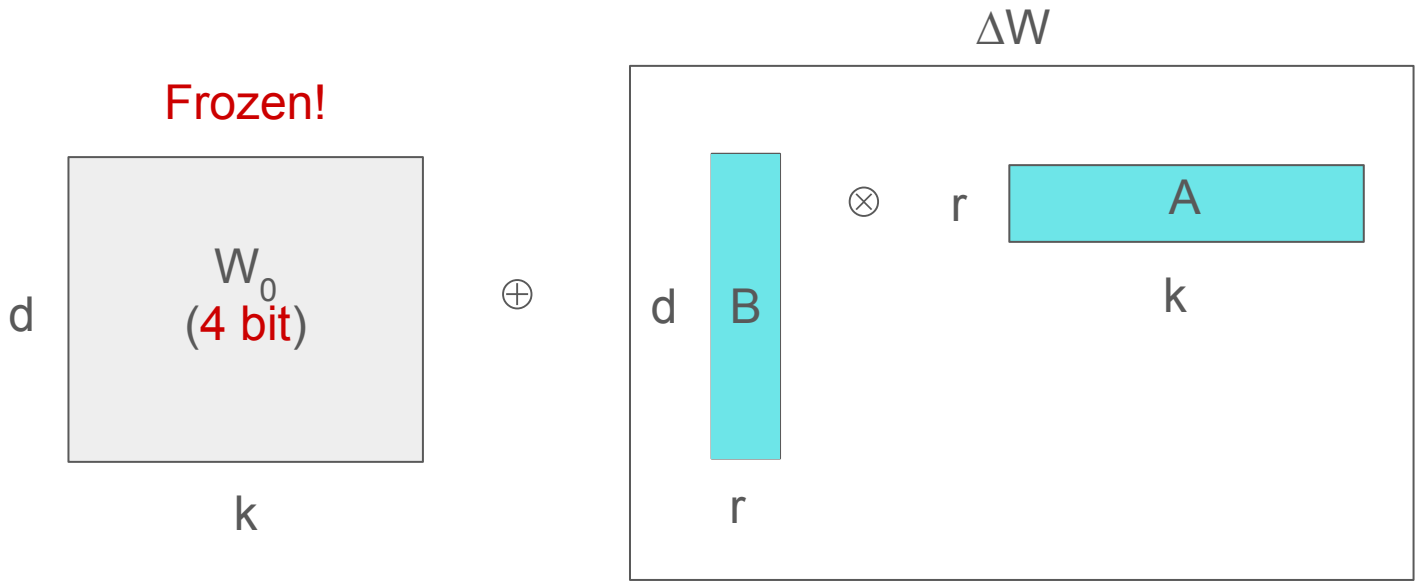
LoRA Fine Tuning

- Adds trainable low rank “adapters”
- $W_0 + \Delta W = W_0 + BA$
- r : Hyper parameter
- A, B (16 bit)



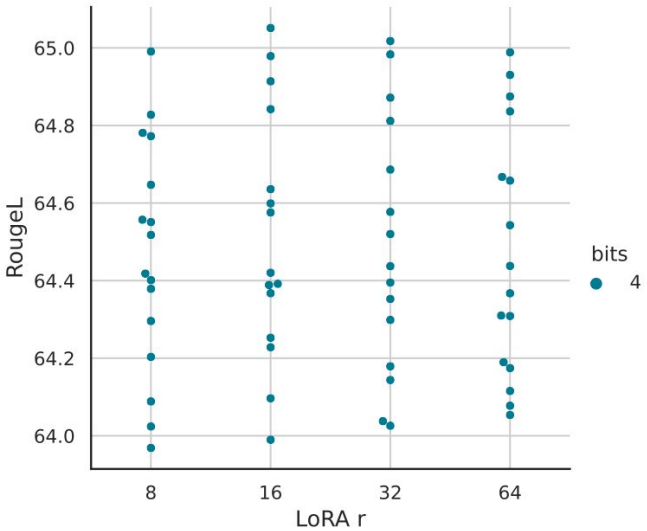
Quantized LoRA Fine Tuning

- $W_0 + \Delta W = W_0 + BA$
- r : Hyper parameter
- A, B (16 bit)



(Q)LoRA Fine Tuning

- Rank (r)
  - Lower rank (e.g., 8 or 16) → specific task
  - Higher rank (e.g., 32, 64) → teaching new concept
- Alpha (α)
  - Common starting point → Set α twice the r or equal
- Target modules
  - Targeting all linear layers increase the performance



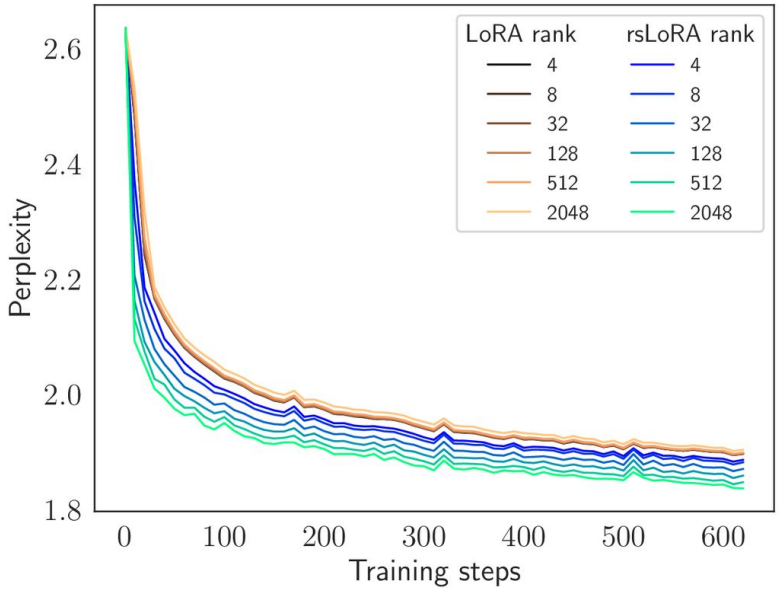
“QLORA: Efficient Finetuning of Quantized LLMs” (May 2023)

(Q)LoRA Fine Tuning

- Continued pre-training
  - Embed_tokens
    - Embedding domain-specific terms
  - Lm_head: producing probabilities for next token prediction
    - Adapting output distribution
    - Improving token generation
- Instruct fine-tuning
  - Attention block: (q_proj, k_proj, v_proj, o_proj)
  - Feedforward network: (gate_proj, up_proj, down_proj).

Rank-Stabilized LoRA (rsLoRA)

- LoRA performs well with lower rank (e.g., 4 to 32)
- BUT higher rank will not improve performance
- $h = W_0 x + (\alpha/r) BA x$
- (α/r) slows learning outside the lower rank regim
- New scaling factor = α /√r



Fine-tuning curves of LoRA vs rsLoRA for Llama 7B on a random subset of the instruction tuning dataset OpenOrca

"A Rank Stabilization Scaling Factor for Fine-Tuning with LoRA" (Nov 2023)

Evaluation

- Traditional evaluation metrics are not suitable!
- Perplexity
  - Measures next token prediction performance.
  - ↓ perplexity ↑ next token prediction accuracy
- Domain- specific benchmarks:
  - MMLU:Astronomy
  - AstroBench
- General benchmarks:
  - IF-EVAL, BBH , MATH , GPQA , MUSR, and MMLU-PRO

“AstroMLab 1: Who Wins Astronomy Jeopardy!?” (July 2024)

Results

- Continued pre-training
  - Excluded “Embed_tokens”
  - Unstable training and validation loss
  - Highest possible rank 64 was ineffective
- Instruction fine-tuning
  - Converging training and validation loss