Why fine-tuning takes longer than the FLOPs say: predicting job time on shared GPU clusters

25 November 2026 · 8 min read · blogs.iictp.org/predicting-fine-tuning-time-catboost-flop-estimate-gpu-cluster-explained

Short answer: a Hugging Face post by a team at Dharma AI, a preview of an upcoming paper, compares the usual FLOP-based estimate of fine-tuning time with learned predictors on 272 real supervised fine-tuning jobs from a shared H200 pool. The authors report that the FLOP estimate explains little (R^2 0.31, mean absolute error 12.41 hours), while a CatBoost model that sees only pre-dispatch information reaches R^2 0.98 and a mean error of 1.71 hours in 5-fold cross-validation. The full paper had not yet been posted when the blog was published.

TL;DR The short version

- A Hugging Face post from Dharma AI compares FLOP-based estimates of fine-tuning time with learned predictors on 272 real jobs from a shared H200 pool.
- FLOPs describe only the training loop, while setup, downloads, evaluation and checkpoints make jobs with the same compute finish hours apart.
- Results come from one organisation's cluster and the full paper is not yet out; log configuration and wall-clock time for every run to build your own predictor.

Read online, listen to it, or change the text size: https://blogs.iictp.org/predicting-fine-tuning-time-catboost-flop-estimate-gpu-cluster-explained

What is the problem?

Cluster owners must decide which jobs run on which GPUs and in what order, and the plan is only as good as duration estimates. The common back-of-the-envelope estimate divides 6 × parameters × tokens by GPUs, peak FLOPS and an assumed model-FLOPs utilisation (MFU).

How it works

- **Compute-only baseline.** The FLOP estimate describes the optimisation loop but not environment setup, model download, tokenisation, periodic evaluation, checkpoint uploads, or contention for network and storage.
- **Learned predictor.** Duration is treated as a regression problem using only information known before dispatch: hyperparameters, dataset sizes, model identifiers and GPU allocation.
- **Derived features.** Examples include evaluation samples, number of checkpoint saves, parameters saved and per-GPU token volume. The FLOP estimate with MFU fixed at 0.45 is itself a feature.
- **Models compared.** Linear Regression, Ridge, XGBoost and CatBoost; the final CatBoost model uses 24 features.

What is new compared with earlier approaches

The point is that overheads are configuration-dependent and do not scale neatly with workload, so two jobs with the same nominal compute can finish hours apart. A learned model captures those interactions from past runs.

Results reported by the source

- Data: 272 production SFT jobs, about 7,382 GPU-hours, on 1, 2 or 4 H200 GPUs from a pool of 8 to 32 across one to four nodes, with real contention.
- Durations from 1.6 to 172.4 hours; mean 23.43 h, median 15.29 h; models up to 15 billion parameters, LoRA and full fine-tuning.
- FLOP baseline: R^2 0.31, MAE 12.41 h. CatBoost: R^2 0.98, MAE 1.71 h, median error 0.51 h in 5-fold cross-validation.

Limitations and open questions

- A single organisation’s cluster and software stack (Hugging Face Transformers and Unsloth); results may not transfer to another.
- Only supervised fine-tuning up to 15B parameters; no pre-training or reinforcement learning.
- The blog is a preview; the paper has not been posted, so methods and splits cannot be checked yet. Sources were read on 25 November 2026. Products, rules and standards change, so check the linked originals before you rely on any detail.

How to build a similar predictor

1. Log job configuration and actual wall-clock time for every run.
2. Add features for evaluation cadence, checkpoint policy and per-GPU load.
3. Keep the FLOP estimate as a baseline feature.
4. Validate by cross-validation across different jobs, not random rows from the same job family.

What it means for engineers and platform teams

Better estimates reduce idle GPUs and slipping queues. The lesson also applies to cost estimates for fine-tuning: include evaluation and checkpoint overhead before promising a delivery date. See Hugging Face on Google Cloud Marketplace billing for another cost-side topic.

Frequently asked questions

- **Why is the FLOP estimate off?** It covers the optimisation loop but not setup, evaluation, checkpoint upload and contention.
- **How accurate is the learned model?** R^2 0.98 and mean absolute error 1.71 hours in 5-fold cross-validation, per the authors.
- **What data was used?** 272 production SFT jobs, about 7,382 GPU-hours on H200s.
- **Is there a paper?** The post says the full version will be on arXiv soon.

Related reading

- Hugging Face on Google Cloud Marketplace
- Harness adapters and agent fine-tuning
- Local LLM Practitioner certification

Related credential: Local LLM Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. Duarte B, Pimenta de Freitas Cardoso G, Carvalho IF, Beleza B, Lima F, Dharma AI. Why your fine-tuning job takes longer than the FLOPs say. Hugging Face, 18 November 2026 (preview of an upcoming paper). huggingface.co/blog/Dharma-AI/predicting-fine-tuning-time

Published by IICTP, an independent technical credential body, on 25 November 2026. The current version of this post is at https://blogs.iictp.org/predicting-fine-tuning-time-catboost-flop-estimate-gpu-cluster-explained. More guides: blogs.iictp.org.