Episode Book

EPISODE #6457

Where to Learn LLM Training Without Generic Courses

PUBLISHED	4 November 2026 at 20:09 UTC
DURATION	20:01
LISTEN	myweirdprompts.com/episode/learn-llm-training-curriculum-96137/
HOSTS	Corn, Herman, Desmond, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
BOOK GENERATED	5 November 2026 at 07:03 UTC

SYNOPSIS

Training and fine-tuning language models has become cheap enough to learn from scratch, but high-quality education remains surprisingly fragmented. While open-source lectures like Stanford's CS 336, Karpathy's repos, and Hugging Face offer deep technical grounding, the scarce resource is compute rather than tuition. This guide maps out the four tiers of self-education—from building transformers and writing CUDA kernels to fine-tuning recipes—while exposing the massive shortage of resources for mastering dataset curation.

TRANSCRIPT



Corn

One hundred and sixty-eight hours. That's how long it took OpenAI to train GPT-two in twenty nineteen. This March, an entry on Karpathy's nanochat leaderboard hit the same mark in about an hour and forty minutes.



Herman

Same target, roughly, on far better hardware, and that gap is the whole reason Daniel's question has a real answer now, because the craft has gotten fast enough that a person can actually learn it end to end on a budget.



Corn

Daniel's prompt this time is a map request. He wants the specialised self-education environments, the places where you learn to create models, train them, fine-tune them, and work with datasets, and he's explicitly ruling out the generic learn-anything marketplaces. His framing is "even the big course marketplaces feel a little too generic for this," which I read as: he wants the protein, not the vitamins.



Herman

And the first surprise of the answer is that almost everything worth recommending is free. The lectures, the code, the books' repositories, all free and open. The scarce resource isn't tuition, it's the GPU.



Corn

Which means the real question isn't "which course," it's "who supplies the compute, and was this thing updated in the last year."



Herman

That's the spine of it. There are four shelves, really. The from-scratch tier, where you build a model from nothing. The fine-tuning recipe shops. The practice and community layer, kernels and contribution. And then the data layer, which is... we'll get there. It's the thinnest.



Corn

Then start at the top. Who makes you build it from nothing?



Herman

Stanford. C S three thirty-six, Language Modeling from Scratch, Tatsunori Hashimoto and Percy Liang, ran this spring, nineteen lectures from tokenization all the way through parallelism, scaling laws, inference, evaluation, SFT and RLHF. The lectures are public on YouTube. And the assignments are the syllabus of what "from scratch" actually means: you build a tokenizer, a Transformer and an optimizer and train them. You write FlashAttention Two in Triton and do distributed training. You fit a scaling law. You filter and deduplicate Common Crawl. You run supervised fine-tuning and reinforcement learning on maths reasoning.



Corn

Five assignments that are each somebody's full-time job.



Herman

Pretty much, yes. This is not a MOOC. There's no hand-holding, the prerequisites are strong Python, PyTorch, systems, a prior ML course. You audit it and then you reconstruct it.



Corn

And the compute?



Herman

You buy it. As of March the course page listed a single B two hundred at somewhere between five and seven-fifty an hour. So the lectures are free and the lab is a rental.



Corn

Free tuition, paid gym.



Herman

And there's no scaffolding catching you if you fall behind. I also can't promise that an outsider can run every assignment exactly as an enrolled student does. Nobody checks whether you did the work.



Corn

Which is the actual test, isn't it. If nobody's checking and the workload is brutal, the only people who finish are the ones who'd have finished anywhere.



Herman

That's my read. C S three thirty-six is a filter disguised as a course.



Corn

The on-ramp below it, though. Karpathy.



Herman

Zero to Hero. Eight videos, about fourteen and a half hours, from micrograd, which is backpropagation built out of nothing, through makemore, up to GPT and the GPT tokenizer. Runs on a laptop or free Colab. It's the single best door into the craft, and it's free. But it ends where the interesting parts begin, there's nothing on fine-tuning, nothing on data pipelines, no graded exercises.



Corn

It's a staircase that stops at the second floor.



Herman

The next step up is nanochat, also Karpathy's, and it's the full-stack version: a minimal LLM harness, MIT licensed, about sixty thousand stars on GitHub, covering tokenizer, pretraining, fine-tuning, evaluation and inference in one tight codebase. The README speedrun does the whole thing on an eight-GPU node in about an hour and a half, for around forty-eight dollars, or fifteen on spot pricing.



Corn

You learn by reading a codebase, not by being taught.



Herman

Correct, that's the honest weakness. There are no lessons in nanochat, it's a well-organized machine you take apart. But that leaderboard is the live measure of the field: the time-to-GPT-two record fell from just over three hours in late January to under two hours by mid-March. That's a public leaderboard chasing a target that took a frontier lab a week in twenty nineteen.



Corn

The floor is dropping fast enough that "I could never afford to train a small model" has stopped being true, and I don't think many people have updated their priors.



Herman

The book-shaped version of the same path is Sebastian Raschka's Build a Large Language Model From Scratch. About a hundred and seven thousand stars on the repo. Seven chapters plus appendices, chapter five is pretraining, chapters six and seven are fine-tuning for classification and instruction following, there's a LoRA appendix, and a companion video that runs seventeen hours. The code is free and works at GPT-two scale on a laptop, the Manning book is paid. There's a sequel out on reasoning models.



Corn

That's the one I'd actually hand someone who asks me at a dinner party, because it has the thing the other two lack: a spine you can follow in order without already knowing the shape of the territory.



Herman

You'd hand someone an entire textbook at a dinner party.



Corn

I don't get invited back, Herman. It's a system.



Herman

I've met your dinner parties. The from-scratch tier, then, is C S three thirty-six if you're serious, Karpathy if you're starting, Raschka if you want it bound. Expensive in time and pride, all three. But most people's actual job, and I'd guess this is where Daniel lives day to day, is not building a model from nothing. It's fine-tuning something that already exists. That's where the specialised ecosystem gets rich.



Corn

This is the Hugging Face kingdom.



Herman

Mostly. We've done Hugging Face against Kaggle already, so I'll stick to the teaching. Hugging Face Learn lists twelve free courses, and the LLM Course is the flagship: it moves from Transformers and fine-tuning through the Datasets and Tokenizers libraries, on to advanced fine-tuning, high-quality dataset curation, and reasoning models. It's paced at six to eight hours a week per chapter, which is real pacing, not weekend pacing. Below it there's the smol course, Apache licensed, covering instruction tuning, evaluation, DPO, vision-language models, RL and synthetic data on most local machines. Though the README says it's migrating into Hugging Face Learn, and whether the later chapters of version two all shipped, I'm honestly not sure.



Corn

So the concepts are all laid out. What trips people up is the plumbing.



Herman

The plumbing is exactly what the recipe shops solve, and there are two of them with opposite philosophies. Unsloth is a catalogue of Colab and Kaggle notebooks that follow one pattern: run all, add your dataset, train, deploy. SFT, GRPO, DPO, ORPO, KTO, continued pretraining, vision, audio, embeddings. Smaller models run on the free tiers, and the biggest ones need paid Colab.



Corn

And the price of that is you never open the hood.



Herman

That's the trade, yes. Unsloth teaches by working, not by explaining. Axolotl is the other philosophy: YAML-driven, you write a config file describing the whole run, full fine-tunes or LoRA or QLoRA, all the same preference-tuning methods, reward models too. It's the more production-leaning of the two. Needs an Ampere or newer NVIDIA, or AMD, and it brings no compute of its own.



Corn

Run it versus configure it.



Herman

That's the fork. Learn by running, or learn by configuring. If you want to understand what a fine-tune is doing, Axolotl makes you look at every parameter. If you want a model by Friday, Unsloth gets you there and you can read the internals later.



Corn

Neither of them teaches you why your dataset is bad, though. They both assume the dataset arrived good.



Herman

Hold that thought, because that's the punchline of the whole map. Before we get to it, there's the practice layer. GPU MODE is the big one there: a public lecture series whose numbering has passed a hundred and ten, a kernel leaderboard running on sponsored GPUs, and this year a Blackwell kernel hackathon and an end-to-end model speedrun with AMD, free to take part in. It teaches systems and kernels rather than model training as such, but my bet is that's a scarcer skill. Writing a fast kernel is rarer than running a fine-tune.



Corn

The glamour is in the models and the hard part is in the memory access patterns.



Herman

Below that, Tensara and LeetGPU do in-browser, benchmarked kernel practice in CUDA and Triton, Tensara free and leaderboard-ranked, LeetGPU with a paid tier, though I couldn't confirm current numbers for either. And Deep-ML is a browser platform of machine learning coding problems, over twelve hundred by its own count, implementing algorithms and attention and tokenization yourself, Leetcode-shaped but for actual ML internals.



Corn

I tried one of those kernel puzzle sites once.



Herman

How did that go?



Corn

My fastest submission is still running.



Herman

Right, before the community layer, let's hear from Desmond.

D

Desmond

And now, Desmond's Money Tip. I reuse the same piece of aluminium foil for a full calendar year. It saves me roughly two pounds and forty pence annually, which over a lifetime of forty years, invested at a cautious three percent, amounts to a sum I would rather not say out loud because it is so small.

D

Desmond

The embarrassing detail is that I have a dedicated foil drawer, and I have named the current sheet. His name is Gerald. Money is not a river, it is a slow drip you learn to live with.



Corn

I'm not saying I'm going to start naming foil. I'm saying Gerald has a better savings plan than I do.



Herman

I'm appalled, and I want to move on. The community layer, quickly: EleutherAI is an open-science Discord with recorded reading groups and a five-week beginner research programme called SOAR, and you learn by contributing to real projects, dataset auditing with the Data Provenance Initiative and so on. Cohere Labs runs an open science community, by its own figures about four and a half thousand members across a hundred and fifty countries, with a research scholars programme and free API grants. Both free, both places where the learning is doing actual work on actual projects.



Corn

Which leaves the layer you told me to hold the thought about. Datasets.



Herman

Yes. Here's the honest finding: I couldn't find a dedicated, current, specialised platform for learning dataset work. The closest thing, MIT's Data-Centric AI course, last ran January twenty twenty-four by its own page, seven lectures with Python labs on label errors, class imbalance, curation. Materials still free, no newer run anywhere I can find. The live options I found are C S three thirty-six's Common Crawl assignment and the dataset chapters of the Hugging Face LLM Course. For a field that will not stop saying "it's all about the data," the teaching supply looks like one assignment and some chapters.



Corn

That's not a gap, that's the gap. Every other layer has a thriving ecosystem and the layer everything depends on has a single lecture hall with a nearly three-year-old calendar on the wall.



Herman

And I think the reason is structural. Dataset work is unglamorous, it doesn't benchmark well, and the people who are good at it are employed, not teaching. The knowledge lives inside companies.



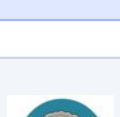
Corn

Which means Daniel's most valuable learning path might be the accidental one: take the C S three thirty-six Common Crawl assignment, fail at it, and learn what the failure teaches. There's no course for that because nobody can grade it.



Herman

Now the other filter, because the map so far has a hazard line drawn around it. Widely recommended is not the same as current. Papers With Code shut down in July twenty twenty-five. The domain now redirects to Hugging Face Trending Papers, which ranks by upvotes and has no benchmark leaderboards at all. Community counts put what was lost at over nine thousand leaderboards and nearly eighty thousand paper-to-code links, and I can't confirm those numbers officially, but the field's reference index is gone.



Corn

And plenty of old "best resources" lists will still point you at it, like directions to a shop that closed last year.



Herman

Karpathy's Eureka Labs was announced in July twenty twenty-four, and its flagship course, L L M one-oh-one N, still does not exist. The repo says so itself, plainly, and it was archived in August twenty twenty-four. More than two years on, no course that anyone can find. Fast AI's Practical Deep Learning for Coders, which is good, is still the twenty twenty-two edition on the site: part one, nine lessons of about ninety minutes, part two, thirty-plus hours up to Stable Diffusion, free on free cloud tiers. Usable today. Not refreshed. Full Stack Deep Learning's latest is twenty twenty-two and a spring twenty twenty-three bootcamp.



Corn

The shelf looks full and half the stock is expired.



Herman

The lesson isn't "don't trust recommendations." It's check the date before the content. A course that hasn't been touched since the pre-reasoning-model era teaches you a field that no longer exists.



Corn

Interesting question though, whether "still the twenty twenty-two edition" disqualifies it. A lot of the fundamentals, backprop, convolutions, the training loop, haven't moved.



Herman

That's fair. Fast AI's foundations are sound and the price is right. But anything touching data scale, inference cost, or post-training is a museum piece. The trick is knowing which half of the course is which, and a beginner can't know that. That's the cruelty of staleness: it punishes exactly the person least equipped to detect it.



Corn

Unless, of course, somebody keeps records.



Hilbert

The specialised ones are better, Herman. No argument there.



Hilbert

But I'll tell you what none of them do. None of them tell you when they've been left to go off. I keep a notebook, a proper one, spiral, and I've been keeping it for years. Every course I start gets a line: what it's called, the date I started, the date I packed it in. A third column says who told me about it, and a fourth column says whether that person had actually done it themselves. I'll let you guess how that fourth column reads.



Herman

Mostly empty?



Hilbert

Almost uniformly. People recommend courses the way they recommend restaurants. They've heard of the place. I count eleven finished against thirty-one started, and I finished the eleven on my own terms, not anybody else's. Here's the part I'll defend, Corn, since you two have been very reasonable today, which I don't love: I will not start a course until I've read its errata page or its issue tracker, all of it, oldest first. If the tracker is full of unanswered questions from two years back, I'm done before I begin. Walked away from a very well-known one on exactly that basis. The questions weren't even hard ones. Nobody was home.



Corn

Eleven out of thirty-one is a better completion rate than online courses usually get.



Hilbert

I don't need encouragement. My cousin Ron does something adjacent to this for a living, data pipelines for a bank, and he says the whole training industry is run by people who have never had a job where the data was wrong. I don't entirely trust Ron. He voted for the wallpaper in his own kitchen. But he's not wrong about that. The fellows maintaining those free courses, the ones that are actually alive, they do it in the evenings, for nothing, and it shows up in the tracker the moment they stop. That's honest work, whatever the dates say.



Herman

The issue tracker as a vital sign. That's... actually a better staleness check than anything we came up with. An abandoned tracker tells you the maintainers left before the syllabus did.



Corn

Ron's wallpaper aside, I'll take that test.



Herman

So take the map as it stands: the from-scratch tier, Karpathy, C S three thirty-six, Raschka, is free tuition and rented compute. The fine-tuning tier, Hugging Face Learn, Unsloth, Axolotl, is rich and practical. Practice and community, GPU MODE, EleutherAI, is where you do real work. The data layer, as far as we could find, is one Stanford assignment and some chapters, and that gap is the finding.



Corn

The one misconception I'd name: people assume the best AI education sits behind a paywall or on a big-brand marketplace, that the serious material is the expensive material.



Herman

It's exactly backwards. Nearly everything serious we named today, the Stanford lectures, Karpathy, Hugging Face Learn, MIT's materials, is free and open. The real bill is compute, and it's measured in dollars an hour, not tuition. The skill you're paying for, whether you pay or not, is knowing which free thing is alive.



Corn

Which leaves an uncomfortable question. If the material is free but the compute isn't, serious AI self-education drifts toward pay-to-play at exactly the moment the craft got cheap enough for anyone to attempt. Forty-eight dollars for a full-stack model is a miracle and still a filter, depending on where you're standing.



Herman

And that staleness problem is structural, it won't age out. The field moves faster than any maintenance cycle, so reading an issue tracker before a syllabus may end up the more durable skill.



Corn

Thanks as always to Hilbert Flumingtop at the desk, producing this whole operation with eleven finished courses behind him and no comment from us on the notebook.



Herman

For more along these lines, there's episode eight, Building Your Own Whisper; episode eleven, How Does Fine Tuning Work Anyway; and episode six, How To Fine Tune Whisper. This has been My Weird Prompts, the human-AI collaboration podcast. If you've got a minute, leave us a review, it helps other people find the show.



Corn

Find everything else at my weird prompts dot com. See you soon, brother.



Herman

See you soon.