Episode Book

EPISODE #6223

How Many Epochs Does Fine-Tuning Actually Need?

PUBLISHED	23 October 2026 at 12:34 UTC
DURATION	20:20
LISTEN	myweirdprompts.com/episode/fine-tuning-epoch-count-guide-05662/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	anthropic/claude-sonnet-5.5
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.20
BOOK GENERATED	24 October 2026 at 07:26 UTC

SYNOPSIS

Almost nobody picks an epoch count in advance — they set a ceiling, evaluate often, keep the best checkpoint, and the epoch count is whatever that turns out to be. This episode works through two concrete cases: a Whisper speech fine-tune scored on word error rate, and an instruction-tuned language model using SFT or LoRA. Along the way: why steps and tokens beat epochs as a unit, why validation loss and word error rate can tell opposite stories, why the best LoRA learning rate runs about ten times the full fine-tuning rate, and what the position of your best checkpoint tells you about what to change next.

TRANSCRIPT



Corn

How many times should a model read the same data before it starts learning the wrong things? And the one that bothers me more: how would you ever know a longer run would have been better? That's Daniel's question this time, and it follows straight on from the fine-tuning run we've been doing. Overfitting is the obvious thing to avoid, but picking the number of epochs still feels like black magic to him. How many does it take, how do you benchmark it, and how do you tell whether more training will actually buy accuracy. He wants it run through two cases: a speech recognition fine-tune scored on word error rate, and an instruction-tuned language model, SFT or LoRA.



Herman

The reframe comes first, because it removes about half the mystery. Almost nobody picks an epoch count in advance. They set a ceiling, a generous one, they evaluate often along the way, they keep the best checkpoint, and the epoch count is whatever that checkpoint turns out to have been. After the fact.



Corn

So the black magic is a receipt. You find out what you bought once you've paid.



Herman

Fair, but the receipt is cheap. Evaluating costs a fraction of training. And the other half of the reframe is that epochs are a poor unit to begin with. We've been through what a pass actually does to the weights, and through overfitting being a slope and not a cliff, so I'll skip both.



Corn

Good. I'd have napped through them again.



Herman

Here's the unit problem. The Hugging Face walkthrough for fine-tuning Whisper runs five thousand steps at a batch size of sixteen. Their table works out to about four hundred and ten steps per epoch, so that run is around twelve epochs. The blog never says that, it's my arithmetic from the table. Now take LIMA, the paper that fine-tuned a sixty-five billion



Herman (continued)

parameter model on one thousand examples. Fifteen epochs, batch of thirty-two. That's roughly four hundred and seventy steps in total.



Corn

Hold on. The whole LIMA run is about one epoch of the Whisper run?



Herman

One and a bit, yes. Fifteen epochs sounds reckless and twelve sounds reckless, and they're wildly different amounts of training. So when somebody says three epochs is plenty, the first question is how many examples. Steps and tokens seen are the honest measure.



Corn

Three epochs on a hundred examples is a few dozen steps. Three epochs on a million is a different weekend.



Herman

And the second knob matters even more than the first: the learning rate. If it's wrong, an epoch sweep tells you nothing, because you're measuring how fast it goes wrong, not how long it takes to go right. For Whisper, the fine-tuning rate people settle on is around ten to the minus five. One Whisper fine-tuning paper says it started from roughly forty times below the pre-training rate, on advice from one of the original Whisper authors. The pre-training figure in the Whisper paper works out closer to twenty times, so I'd say twenty to forty and not lean on the ratio.



Corn

Twenty to forty times gentler than the original training. It's a fine-tune, not a renovation.



Herman

LoRA has its own quirk. Thinking Machines ran a large comparison and found the best LoRA learning rate was about ten times the best full fine-tuning rate, across a dozen-odd Llama and Qwen models. That was a single-epoch regime, so I wouldn't promise it carries over to a multi-epoch speech run. But the mistake it points at is real: someone ports a full fine-tune rate onto an adapter and wonders why nothing moves.



Corn

Daniel's speech case, then. Yesterday's point was that an epoch measures exposure, not achievement, so the scoring has to happen on audio the model has never heard.



Herman

Fresh audio, ideally different speakers and different microphones, scored on word error rate. One practical detail people skip: normalise the text before scoring, casing and punctuation and numerals written out one way, or you end up measuring formatting habits instead of listening. And the Whisper blog turns out to be a free worked example. Evaluation every thousand steps, so five points on the curve, keeping the best on word error rate. The pretrained model sits at sixty-three and a half percent. The best checkpoint lands at thirty-two point oh one, at step four thousand.



Corn

And step five thousand?



Herman

Thirty-two point one oh. Basically the same. Now the part that made me stop. Training loss ends up around zero point zero zero zero two, so the model has nearly memorised the training set. And validation loss climbs, from about zero point three oh seven at step one thousand to zero point four six eight at step five thousand. Worse at every single checkpoint.



Corn

While the word error rate kept improving.



Herman

Until step four thousand, yes. The blog never comments on it, it just reports the table.



Corn

Two held-out curves telling opposite stories. I'd believe the one I'm being paid for.



Herman

Same, but the reason is worth having. Validation loss is cross-entropy, and cross-entropy punishes overconfidence. A model can get a few examples confidently wrong and the loss jumps, even though the word it actually outputs is right more often than before. So loss starts rising from overconfidence well before anything you'd call memorisation. Word error rate only looks at the final transcript.



Corn

Does it always run that direction? Loss up, metric fine?



Herman

No. People have reported the reverse on small sets, word error rate rising while validation loss keeps falling. I'm going by reports there, not a clean study, so hold it loosely. That's why you can't swap one signal for the other. Choose by the number you'll be judged by, and for speech that's word error rate on audio that looks like deployment.



Corn

Then the held-out set is doing a lot of the work.



Herman

It is, and its size sets your resolution. Thirty-two point oh one against thirty-two point one oh is a tenth of a point. On a few hours of test audio that's noise. Calling step four thousand better than step five thousand would be fooling yourself. What you can say is that the curve flattened.



Corn

Flattened, and then you pay for another thousand steps anyway.



Herman

Which is the interesting part for Daniel. There's a diagnostic hiding in where the best checkpoint sits. If the best one is the last one you saved, your ceiling was too low, because the curve was still going when you cut it off. If it's the very first evaluation, the learning rate is probably too hot, or the data is tiny and you're past it already. Best in the middle, with a flat or rising stretch afterwards, is a healthy run.



Corn

So the position of the winner tells you what to change.



Herman

It does. And here's something I'd add that the blog doesn't, and it's my instinct, not a result. Keep a second held-out set that isn't your target domain, just ordinary speech. Whisper is reported to forget things when you push it hard on a narrow set. If the target error flattens and the general one starts climbing, that's your stopping point even though the number you were chasing looks fine. Corn, if you were setting up this run, would you stop at the first flattening, or let it go to the ceiling and take the best?



Corn

Ceiling. I don't stop early on a curve that wiggles. Wiggles come back.



Herman

I lean the same way, and there's a reason beyond noise. Training curves aren't always monotone. Test error can fall, rise, and fall again as training continues. That's a result from image classifiers with noisy labels, and I wouldn't expect it in a typical fine-tune, but it's why I like a ceiling plus a patience window and not a stop on the first dip.



Corn

We have a caller on this, and he wants to go back to the very start, which is fair, since the whole question is how many of these passes to run.
Doug: Hi, sorry, I have been listening for a while now and nodding along, but I think I missed step one. What actually is an epoch? Is it like a season of the year, or like a shift at work, does one take about an hour? And when you say fine-tuning, is that like tuning a radio, do you just turn the knob until the talking gets clear?



Herman

An epoch is reading your whole cookbook once through, and fine-tuning is taking a cook who already knows how to cook and teaching them your grandmother's recipes, so no, no knob, and it takes however long the cookbook takes.



Corn

That cost him something. Daniel's second case, the instruction-tuned model. Where does the epoch count come from there?



Herman

Convention, mostly, and I'd call it convention, not a proven optimum. Guides like Unsloth's say one to three epochs for LoRA, with diminishing returns after three, and a LoRA learning rate that starts around two times ten to the minus four. I haven't seen a controlled sweep of win-rate against epochs that proves one to three is right. It's what people converged on because it mostly works.



Corn

And the stopping signal is the loss again.



Herman

The same guide says that if training loss falls below about zero point two, suspect overfitting. It's a useful warning light. But LIMA is the better test of it. They trained fifteen epochs, and held-out perplexity on a couple of thousand Stack Exchange examples went up as training went on. At the same time the generations got better. The authors say outright that perplexity did not correlate with generation quality.



Corn

The thermometer says fever and the patient is walking around.



Herman

Retired pediatrician, so I'll allow it. What LIMA actually did was pick a checkpoint between epoch five and epoch ten, by hand, using a dev set of fifty examples and reading the outputs.



Corn

Fifty. That's a small dev set to hang a model on.



Herman

For a thousand-example project, I think it's fine, because the thing you're deciding is coarse. For your own run the idea scales: sample outputs at every checkpoint, run the task the model is for, and score it. Format compliance, correct answers, a judge model if you have to, and your own eyes on a handful. That's the instruction-tuned equivalent of word error rate. Training loss is nearly useless for deciding, because it always goes down. Validation loss is a smoke alarm. The task score is the verdict.



Corn

Give me a concrete run, then. Two thousand examples, say.



Herman

Batch of sixteen, so about a hundred and twenty-five steps an epoch. Set the ceiling at five epochs, around six hundred and twenty steps, evaluate every half epoch, so you get ten points. Warmup five to ten percent of steps, linear or cosine decay, a bit of weight decay, LoRA dropout around zero point one if you're seeing a gap open. LIMA used a fancier dropout ramp across layers, but for an adapter the simple one is the usual advice.



Corn

And if the best checkpoint is at epoch two?



Herman

Then you learned the epoch count is two, for this data and this rate. Next time you have similar data, start with a ceiling near that and don't pay for the tail.



Corn

What about the hosted route? Daniel's probably seen the buttons. I once pressed auto on something and felt very responsible.



Herman

OpenAI's fine-tuning API lets you set epochs, batch size and learning rate multiplier to auto, and the documentation doesn't say how it picks. I haven't seen a formula anywhere. What they do give you is an epoch checkpoint saved at the end of each epoch, the last three kept, precisely so you can roll back if the model improves early and then starts memorising. Same philosophy as everything else. Save, score, pick.



Corn

Then what Daniel really asked. Say I've got a decent checkpoint at epoch three. How do I know epoch six would have been better, without paying for it?



Herman

You mostly can't, and the reason is subtle. Train longer is not the same run extended. Learning rates decay on a schedule towards a planned end point, so your three-epoch run spent its whole schedule annealing down to nothing at step three. A six-epoch run anneals more slowly and ends somewhere different. If you compare the epoch-three checkpoint of a six-epoch run, which still has a high learning rate, against the finished three-epoch run, that isn't a fair test.



Corn

So the fair test is two full runs. One short, one at double, each with its own decay.



Herman

Each complete, yes. A cosine schedule stretched well past its real step count is reported to leave you under-trained, and matching schedule length to run length gave the best result at each budget. I only have that from a secondary description, but the logic holds without it. Then you have two numbers on the same held-out set, and the question is whether doubling the cost beat the noise.



Corn

Doubling the cost to buy a tenth of a point is a poor deal.



Herman

Often it is. Thinking Machines saw loss falling roughly in a straight line against the log of the number of steps. If that holds, every doubling buys about the same small improvement, never a growing one. That tells you what to expect from run two: modest and predictable, not a breakthrough. And run to run noise from the random seed alone can swamp a tenth of a point, so you'd want the gap bigger than that before you believe it.



Corn

Then the cheap trick is the pilot. Small slice of the data, find the learning rate, then scale.



Herman

Ten to twenty percent of the data, a handful of learning rates, a handful of evaluation points. Pick the rate, then do the full run with a ceiling. It sorts out the rate problem before you spend on epochs.



Corn

Hang on, though. If I'm training on the whole pile, is it dishonest to look at the held-out score fifty times and keep the best? Isn't that overfitting to the held-out set?



Herman

A little, yes, and it's a fair worry. Keeping the best of ten checkpoints on a small dev set gives you a slightly flattering number. With a few thousand held-out items it's mild. With fifty it's not, and that's why serious people keep a third set that picks nothing, which they score once at the end.



Corn

Daniel's got a lot of audio of his own voice lying about. A third pile costs him nothing.



Herman

It costs him a little care, because it has to be audio the run never touched in any way. Not used for training, not used for picking. And here's where I'd ask you something, Corn. Say his voice data is ten hours, and the held-out part is one hour. Does that one hour mean anything?



Corn

It means something about his voice on that day. Not about speech in general. If he's deploying on himself, fine. If he's selling it to someone with a cold and a bad microphone, no.



Herman

That's the line. The held-out set defines what accuracy means, so it should be the world you'll actually be in. A set that's easier than deployment makes every extra epoch look like progress.



Corn

Everything improves if you grade it on the homework.



Herman

There's a smaller version on the language model side, too. Instruction tuning for long enough can dent what the base model could already do, and I'm going on indirect evidence for that, nothing epoch-specific. So I'd keep a handful of prompts the base model handled well and check them at every checkpoint. If those start slipping while your task score rises, you've found the trade.



Corn

Which is the second exam again.



Herman

Same exam, different subject.



Corn

I'd still like the pre-training numbers people keep waving around at this question.



Herman

Then there's the pre-training evidence, which people quote at the epoch question. Muennighoff and colleagues ran models up to nine billion parameters on repeated data. Up to about four epochs was nearly as good as fresh data, a half a percent difference in validation loss on the largest one they compared.



Corn

Four is free. I'll take it.



Herman

Past that it decays. They fit a half-life of repetition around sixteen epochs, and at forty-four the loss rises midway and the run diverges.



Corn

So a thousand-example fine-tune at fifteen epochs shouldn't survive.



Herman

It does, and that's the caveat. That work is pre-training on web text, from scratch. A small fine-tune at a low learning rate with the best checkpoint kept tolerates much more repetition. LIMA at fifteen, the Whisper blog at twelve. Different regime, different rules. What they share is the habit of not trusting the last checkpoint.



Corn

I notice you aren't giving Daniel one number.



Herman

I'd rather give him a procedure than a number I can't defend. If he wants a starting ceiling, for a few thousand SFT examples I'd set around five epochs and see where the best checkpoint lands. For speech on a modest set, twelve is what the Whisper recipe used, and it'll show where the curve flattens. Learning rate first, ceiling second, eval five to ten times a run, task metric picks, patience stops.



Corn

A ceiling of five, written down so I can pretend I always knew.



Herman

A receipt.



Hilbert

Sounds right to me. Bernard Oyelaran has told me my grout is coming Thursday for forty-one weeks, and I wanted it in writing, so I'm training a Whisper on his voicemails. Eleven hundred of them. Herman, you said ceiling a lot, so I set mine at fifteen.



Herman

Fifteen on a speech model is high, but with a low learning rate and a checkpoint every epoch it's workable.



Hilbert

Epoch seven, the model stopped hearing days. Everything Bernard said came out as pallet of grout Thursday. Including my own voice. As a test I pasted that into his order form and he honoured it. Eleven pallets.



Corn

Eleven pallets is a lot of Thursdays.



Hilbert

Bernard holds the record for longest voicemail with no verb in it. Seven minutes forty seconds. That one's in the training set. Corn, you said to believe the number you're being paid for. Bernard gets paid on Fridays. I've started ringing him on Fridays.



Herman

Then the checkpoint to ship is from before it got that confident. Epoch four, epoch five.



Hilbert

Epoch four says Thursday on the Thursday calls. The invoice came to two thousand three hundred and thirty-two dollars. Bernard rang to say Thursday.



Corn

Epoch four was already sitting on the disk. That's the whole argument for saving one every epoch.



Herman

It changes Daniel's budget question too. Saving checkpoints is nearly free, so the extra epochs you didn't need cost compute and nothing else. The expensive mistake is a run you can't step back through. And when a run won't improve, I'd check the learning rate before I raise the ceiling.



Corn

So the forward-looking part is smaller than a number. Make the run something you can rewind, and let the best checkpoint tell you the epoch count on its way out the door.



Herman

Thanks to our producer, Hilbert Flumingtop, for getting this one to the finish. This has been My Weird Prompts, and you can find every episode at my weird prompts dot com.



Corn

See you tomorrow.