GO UNDERNEATH

Running Models Yourself

Local and self-hosted LLMs, with the arithmetic instead of the marketing.

First edition, September 2026

Addaly

Series editor: Dr Mustafa Mun

Draft, open for academic review

GO UNDERNEATH

Running Models Yourself

Local and self-hosted LLMs, with the arithmetic instead of the marketing.

Series editor: Dr Mustafa Mun

Addaly

First edition, September 2026 · Draft, open for academic review

Running Models Yourself

First edition, September 2026, build 62a26075.

Written by AI models to a written standard, with Dr Mustafa Mun as series editor. Exam keys checked blind by a second model: 3,665 of 3,666 agreed. Academic review invited.

The manuscript was drafted with the assistance of a large language model and was edited and published under his name. That is stated plainly here rather than left to be discovered, because a reader is entitled to know how a book they are trusting was made.

How to cite: *Mun, M. (2026). Running Models Yourself. Addaly. addaly.com/learn/running-models-yourself.*

This book is the printed form of the course of the same name at addaly.com/learn/running-models-yourself. The website is the living version and is corrected first. Where the two differ, the website is right.

Free to read, free to download, free to print, and free to teach from. There is no paid edition and there never will be.

Every diagram in it is drawn from data by code, not generated as a picture, so the numbers on the page are the numbers in the argument. The answers to every check, test and examination question are at the back.

9 modules · 82 lessons · 28 figures · 9 case studies · 61,342 words · about 12 hours

Brief contents

1

MODULE 1

Deciding to run it yourself

Decide whether a stated task belongs on your own hardware — by auditioning the exact open weights through a hosted copy, naming the constraint that forces local rather than assuming one, reading a licence well enough to say what you may ship, and stating the capability ceiling the weights impose regardless of the machine.

Lesson 1 · 6 min

Why run a model yourself, and when not to

THE ONE THING TO KEEP

The weights set the quality ceiling; hardware only decides whether you can reach it and how fast.

Almost everything written about local models is either an advertisement or an install guide. Start with the question those skip: should you do this at all?

What running it yourself actually buys

The data never leaves. Clinical notes, legal discovery, an employer's source code, other people's private messages. No processing agreement, no sub-processor list, no question about training on your inputs. For some work this is not a preference, it is the condition of doing the work at all.

No rate limits, no queue, no account. You can hammer it for six hours. Nobody suspends you at 2am mid-batch.

It works with no network. On a plane, on a ship, in a lab with no outbound access, in a country where the API is blocked or the payment method is not accepted. This last one is not a small case.

Nobody deprecates it. Hosted models get retired with a few months' notice, and the replacement behaves differently. A GGUF file on your disk behaves identically in five years. If you have tuned prompts around a specific model's quirks, that stability is worth real money.

Cost, but only under conditions. Local can be dramatically cheaper or dramatically more expensive than an API depending on one variable. Lesson 10 does that arithmetic properly. Do not assume it yet.

The honest reasons not to

The quality gap is real and it is large. An 8B model at 4-bit is a capable 2023-class assistant. It summarises, classifies, extracts, rewrites, and writes short functions well. It does not do long multi-step reasoning, does not hold a large codebase in its head, and calls tools less reliably. No amount of hardware changes this — the weights set the ceiling.

Your time is the largest line item. An afternoon spent debugging a chat template costs more than a year of light API use at any professional rate.

Speed, especially before the first token. On a CPU, generation might be tolerable while processing a 2,000-token prompt takes 30–60 seconds. That delay, not tokens per second, is what makes people give up.

You are the operations team. When it stops working at 3am, that is your problem.

A test that decides it in an hour

Do not start by buying hardware. Separate two questions that people fuse together: *is this model good enough for my task?* and *can I run it?*

Take twenty real prompts from your actual work. Run them against a hosted copy of the exact open weights you would run locally — most open models are available through several providers for a few cents. Grade the outputs yourself.

If the quality is unacceptable there, it will be unacceptable locally, and a bigger GPU will not fix it. If it is acceptable, you now know the target and can shop for the cheapest way to run those specific weights.

This costs about a dollar and it is the single most useful thing in this course.

What you will need to be able to do

The rest of these lessons assume you want to answer four questions without asking anyone:

1. Given any model on Hugging Face, does it fit in my memory, and at what context length?
2. What did quantisation cost me, and where will I notice?
3. Which serving engine wins for my pattern of use, and by what measure of "wins"?
4. Is this cheaper than the API for my volume?

Those are arithmetic and engineering questions with real answers. Almost nothing else in this field is.

BEFORE YOU MOVE ON

Someone runs Llama 3.1 8B locally and finds it noticeably worse than the same model served by a hosted provider. They conclude they need a bigger GPU. What is the best response?

A. Same weights have the same ceiling, so check quantisation level, chat template, context truncation and sampler settings before buying anything.
B. A larger GPU gives the model more capacity to work with, so quality does improve with more VRAM.
C. Providers serve a larger private model behind the public name, so a local copy can never match it.
D. Quantisation always costs a large share of quality, so a local model is inherently the weaker one.

The answer and why: p. 443

Lesson 2 · 9 min

What "open" means, and what it does not

THE ONE THING TO KEEP

"Open" splits into open weights, open source and open data — most famous models are only the first, and a base model's licence and use policy follow every fine-tune and quant made from it.

Three different things get called open

The word is doing too much work. Separate it into three claims, because only one of them is usually true.

Open weights. The trained parameters are downloadable. You can run them, quantise them, fine-tune them, and inspect every number. This is the common case and it is what this course is about.

Open source. The licence meets the Open Source Definition — no restriction on field of use, no restriction on who may use it, no conditions on downstream users. Apache-2.0 and MIT models qualify. Most of the famous ones do not.

Open data. The training corpus is published, so the model can be reproduced from scratch. This is rare. AI2's OLMo and the Hugging Face SmolLM family publish theirs; almost nobody else does. Without it you cannot audit what the model was trained on, which matters for contamination, for bias claims, and for any question about copyright.

A model can be open weights and closed source and closed data at the same time, and most are.

The licences you will meet

- **Apache-2.0 / MIT.** Genuinely permissive. Commercial use, redistribution, modification, no user cap. Qwen, Mistral's smaller releases, Falcon, OLMo, DeepSeek's open releases, Phi. If you want to stop thinking

about licences, stay here.

- **Llama Community Licence.** Meta's own terms, not open source. Commercial use is allowed, but there is a monthly-active-user threshold above which you must ask Meta for a separate licence, an acceptable use policy attached, and a naming requirement — derivative models must carry "Llama" in the name and the attribution notice must be reproduced.
- **Gemma Terms of Use.** Google's. Commercial use allowed, no user cap, but a use policy that binds you *and* everyone you distribute to, and Google reserves the right to require you to stop using a model remotely if it is being used in breach.
- **Non-commercial licences** (CC-BY-NC and friends). Research only. Several strong models sit here and people deploy them anyway; that is an infringement, not a grey area.
- **Gated but permissive.** Some Apache-2.0 models still require you to accept terms on Hugging Face and be approved before download. The gate is a distribution control, not a licence term.

The part almost nobody checks

The base model's licence follows its descendants. A fine-tune of a Llama model is still under the Llama licence no matter what the uploader wrote in their own README. When you download `SomeLab/CoolModel-8B`, the questions are: what was it fine-tuned from, and what did *that* licence say. The model card usually names the base model in the first paragraph; the config file names the architecture, which is a strong hint.

The same applies to synthetic training data. If a dataset was generated by a commercial API, that API's terms may forbid using it to train a competing model. Several popular fine-tunes carry that history and do not mention it.

Where the law is genuinely unsettled

Two questions have no settled answer, and the answer differs by country:

1. **Was training on copyrighted text lawful?** The United States is arguing this through fair use, the EU through the text-and-data-mining exception and its opt-out, Japan has a broad exception, the United Kingdom has consulted repeatedly without concluding. Cases are live in several jurisdictions and have gone both ways on different facts.
2. **Who owns the output?** Several offices, including the US Copyright Office, have said purely machine-generated output is not protectable, while human-authored contributions around it may be. What counts as enough human contribution is not defined.

This course does not resolve either, because they are not resolved. What it can tell you is the practical shape: running a model locally does not change your copyright position at all, and a licence from a model provider is a licence to use the weights, never a warranty that the weights were lawfully made. If your use is commercially serious, that is a question for a lawyer in your jurisdiction, not for a forum.

The practical routine

Before downloading anything you intend to build on:

1. Open the model card and find the licence field and the base model.
2. Read the acceptable use policy if there is one. It is short and it binds you.
3. Save a copy of the licence text alongside the weights. Licences have been changed after release; the copy you agreed to is the one you can point at.
4. If you will redistribute anything — a fine-tune, a quant, a container image — check the attribution and naming clauses, because those are the ones that catch people.

That takes ten minutes and removes an entire class of problem that only ever surfaces once something you built has become important.

BEFORE YOU MOVE ON

You fine-tune a Llama-family model on your own data and want to publish the result on Hugging Face under Apache-2.0, since your training code and your dataset are both yours. What is wrong with that plan?

A. Nothing, provided you credit Meta in the README — attribution satisfies the community licence for derivative works.

B. Fine-tuned weights cannot be published at all, because the resulting parameters are a derivative of copyrighted training data.

C. The derivative weights remain under the original community licence, which you cannot relicense, and it carries naming and use-policy conditions that pass to your users.

D. Apache-2.0 is the wrong choice only because it lacks a patent grant appropriate for model weights.

The answer and why: p. 444

Lesson 3 · 9 min

Reading a model card without being sold to

THE ONE THING TO KEEP

The Files tab of a model card is verifiable and the prose is a claim — config.json, the shard sizes and the chat template tell you more in three minutes than any benchmark table on the page.

The page is marketing and specification at once

A Hugging Face model card is written by the people who want you to use the model. Some of it is verifiable fact and some is a claim. Learn which fields are which, and you can assess a model in three minutes.

CASE STUDY · MODULE 1

Thirty discharge summaries and a privacy problem

An illustrative case, built from documented patterns.

A district hospital outside Nashik, where a registrar wanted discharge summaries drafted from ward notes, and the consultant had already said the notes do not leave the building.

The registrar had read that an 8B model at 4-bit runs on a card costing about half a month's salary, and that it summarises well. Both claims are true. He wrote a note asking for the money.

The hospital's medical superintendent asked one question before signing: how do we know it is good enough at *this*. Nobody could answer, because nobody had tried it. The obvious way to try it is the audition — take thirty real items, run them through a hosted copy of the exact open weights, and grade the outputs. It costs under a dollar and it answers the model question without answering the hardware question.

Except that the reason for the project was that ward notes cannot leave the building. Running thirty real discharge notes through a hosted endpoint, even for an hour, even for a test, is exactly the thing the whole deployment exists to prevent. The registrar had proposed to establish privacy by breaching it.

The two ways out, and what each costs

Write synthetic notes. Invent thirty plausible ward notes and audition on those. Free, immediate, and nobody's data moves. The problem is the one the course states plainly: invented examples are always easier than real ones in a specific way. They are shorter, cleaner, better punctuated, and free of the internal shorthand that actually breaks models. Ward notes at this hospital contain abbreviations used only in that ward, drug names written three ways, Marathi words in Devanagari inside English sentences, and handwriting

transcribed by whoever was on duty. A synthetic set would have contained none of that, and the audition would have returned a clear pass on a task the model had not been asked to do.

De-identify thirty real notes by hand. Two people, a morning each, removing names, registration numbers, addresses and dates, and checking each other's work. Real structure, real abbreviations, real mess, and nothing identifiable leaves. The cost is a day of two clinicians' time before a single line of the project exists, spent on something that produces no deliverable — which is precisely the kind of work that gets cut when a project is trying to prove itself quickly.

There was a third option nobody liked: buy the card first and audition locally. It removes the privacy question entirely. It also spends the money before learning whether the weights can do the job, which is the sequence this course exists to prevent.

What the set was for

The registrar argued for synthetic notes on the grounds that the model would be corrected by a doctor anyway. The superintendent's objection was not about the risk of a bad summary. It was that an audition on easy items cannot distinguish between two models, which means it cannot tell you which weights to shop for, which means the purchase is still a guess with a document attached.

They also had to decide what the thirty items were for afterwards. The course is insistent that the set is the most valuable artefact the project owns — it is what tells you later whether a quantised copy still works, whether a new release is better, and whether a prompt change helped. A de-identified set can be kept, versioned and re-run for years. A set of real notes cannot be kept at all.

That argument turned out to matter more than the audition itself.

YOUR ANSWERS

1. The registrar's plan would have sent real ward notes to a hosted provider for one hour. Why is that not a small, temporary exception?

2. Synthetic notes were free and immediate. What specifically would the audition have failed to measure?

3. The audition came back at nineteen usable, eight near-usable and three wrong. What should happen next, and what should not?

STOP HERE

Write your answers before you turn the page. How it played out starts on p. 50.

CASE STUDY · MODULE 1

How it played out

They de-identified thirty notes by hand over one morning, with a second clinician checking. Two candidate models were run through a hosted endpoint at temperature 0, the outputs were shuffled and graded blind on a three-point scale, and the whole exercise cost about seventy rupees of tokens. Nineteen of thirty were usable as drafts; eight needed a small edit; three were wrong in ways that mattered, all of them on notes containing ward-specific abbreviations. That is the awkward middle, and it moved to a clear pass after they added a two-line glossary of those abbreviations to the prompt — a change that cost twenty minutes and would never have been discovered on synthetic notes, because synthetic notes do not contain abbreviations nobody outside the ward uses. The card was bought six weeks later, for a model they had already tested. The de-identified set is now run against every engine upgrade, and it caught a regression the following March.

The questions, discussed

1. The registrar's plan would have sent real ward notes to a hosted provider for one hour. Why is that not a small, temporary exception?

Because the exception is the whole project. The deployment exists because the notes may not leave the building, and a test that breaks that rule has not tested the system, it has performed the thing the system was built to avoid. There is also no such thing as a temporary transfer: once the notes have been sent, they have been processed by a third party, and the hospital cannot demonstrate otherwise. The de-identification cost a morning and left the hospital able to say truthfully that no identifiable note has ever left.

2. Synthetic notes were free and immediate. What specifically would the audition have failed to measure?

The failures. Invented examples are shorter, cleaner and better punctuated than real ones, and they omit exactly the internal shorthand that breaks models. The three items that failed all contained ward-specific abbreviations, which is the class of problem a synthetic set cannot contain, because the

person writing it does not think to include what they already understand. A set with no failures in it cannot distinguish between two candidate models, which is what the audition is for.

3. The audition came back at nineteen usable, eight near-usable and three wrong. What should happen next, and what should not?

That is the awkward middle, and the course gives an order: fix the prompt, then try the next size up in the same family, then a task-specialised model, then a newer model in the same class. Each takes about twenty minutes. What should not happen is a decision about hardware. A bigger card runs the same weights faster and does not make them better, so buying one at this point spends money against a result that has not settled. Here the prompt fix alone moved it to a clear pass.

MODULE 1 TEST

Check what stuck

6 questions, one right answer each. This one is for you, not for a record. The answers, and the reason behind each, are in the key on p. 442. If two go wrong, re-read the module before the exam.

1 An audition of thirty real items against a hosted copy of the exact open weights returns eleven items unusable. What does buying a faster graphics card change about that result?

A. Nothing at all
B. It lifts quality, because more memory means a longer context window
C. It lifts quality, because a faster card can run a higher bit rate
D. It lifts quality, because generation speed and answer quality track together

2 You download SomeLab/CoolModel-8B. The uploader′s README says MIT. The model card′s first line says it was fine-tuned from a model released under Meta′s community licence. Which terms govern what you may ship?

A. MIT, because the uploader owns the fine-tuned weights
B. Meta′s terms, because a base licence follows its descendants
C. Whichever is more permissive, since both were granted
D. Neither, because weights are automated output and carry no licence at all

3 You are on a slow connection and want to know in three minutes whether a model will fit and behave. Which two files tell you most?

A. The README′s benchmark table and the downloads count
B. generation_config.json and the licence field
C. The model card prose and the list of supported languages
D. config.json and tokenizer_config.json

4 A machine has 64 GB of system RAM at about 30 GB/s and no useful GPU. Why does a 30B-A3B model generate several times faster on it than a dense 30B at the same quantisation?

A. Because the router keeps the popular experts in cache and streams the rest
B. Because only the active parameters are read per token
C. Because mixture-of-experts files are smaller on disk at the same bit rate
D. Because expert layers are computed in parallel across the processor's cores

5 You ask a freshly downloaded model for the capital of France and it replies with three more exam questions. What is the most likely explanation?

A. The chat template is missing a system role
B. The sampler's temperature is set too high
C. You have the base variant, not the instruct one
D. The quantisation has damaged the instruction-following weights

6 A team runs weights locally and wants evidence that nothing leaves the machine. Which check is the strongest?

A. Reading the application's privacy policy and telemetry settings
B. Confirming the model file came from a reputable host
C. Blocking it at the firewall and confirming it still answers
D. Checking that the server is bound to 127.0.0.1 rather than 0.0.0.0

Lesson 10 · 9 min

Does it fit: the arithmetic

THE ONE THING TO KEEP

Weights are fixed; the KV cache grows with context and layers, and that is what usually breaks the budget.

Stop looking up tables of "models that fit in 8GB". Learn the two sums and you can answer for any model, any quant, any context length.

Total memory = weights + KV cache + overhead.

Weights

```
bytes = parameters x bytes_per_weight
```

Bytes per weight comes from the format. Useful figures, in bits per weight (divide by 8):

format	bits/weight
fp16 / bf16	16
Q8_0	8.5
Q6_K	6.6
Q5_K_M	5.7
Q4_K_M	4.8
Q3_K_M	3.9
IQ2_XS	2.4

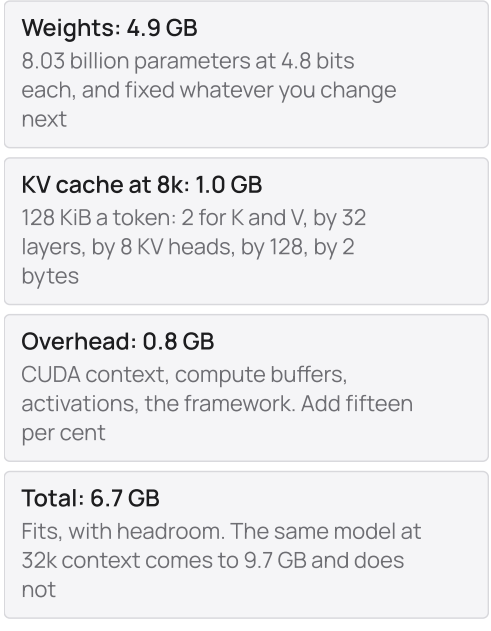
The 4-bit formats are above 4 bits because each block of weights carries a scale. Lesson 3 explains why.

So Llama 3.1 8B at Q4_K_M: 8.03e9 x 4.8 / 8 = 4.8 GB. The published file is 4.9 GB. Close enough to plan with.

A 70B at Q4_K_M: 70e9 x 0.6 = 42 GB. Two 24 GB cards, or a 64 GB Mac, or it does not happen.

Figure 2.1

An 8 GB card, an 8B model, and where the gigabytes go



Weights are the number people quote and the cache is the term that breaks the budget. Raise the window to 32k and the cache quadruples to 4.0 GB while the weights do not move at all, which is why the first remedy is almost never a smaller quant.

KV cache

Every token you have already processed leaves behind a key and a value vector in every layer. This is what "context costs memory" means.

EXAMINATION

Running Models Yourself — final examination

This paper covers the whole course:

- deciding whether a task belongs on your own hardware
- the memory and bandwidth arithmetic that says whether a model fits and how fast it will run
- quantisation formats and how to measure what they cost
- getting a model to a working endpoint with the right build and the right template
- serving engines and the tuning that actually moves the number you care about
- diagnosing a disappointing output to its real cause
- exposing an endpoint to other people safely
- the honest cost comparison against a hosted API
- adapters and training

63 questions, the whole pool. The site draws 25 for a sitting, pass mark 70%. Work any 25 under your own conditions. Answers from page 497.

1 Module 1 · Deciding to run it yourself

A model's weights are downloadable and its licence forbids use above a monthly-active-user threshold. Which of the three senses of open does it satisfy?

A. Open weights and open data, since the corpus can be inferred
B. Open weights only
C. Open weights and open source
D. Open source only, since the weights are published under it

2 Module 1 · Deciding to run it yourself

A reasoning variant answers the same question as an instruct model but emits 1,800 tokens of deliberation first. At 20 tokens a second locally, what follows?

A. The answer costs about ninety seconds before it begins
B. Nothing, because the thinking block is computed in parallel
C. The answer becomes more accurate on summarising and extraction
D. The deliberation must be fed back so the next turn stays coherent

3 Module 1 · Deciding to run it yourself

A repository is called BigModel-R1-Distill-Qwen-7B. What have you downloaded?

A. A 7B adapter that must be applied to the large model
B. A pruned copy of the large model with layers removed
C. The large model compressed to seven billion parameters
D. A Qwen 7B fine-tuned on the large model's outputs

4 Module 1 · Deciding to run it yourself

A task needs six dependent steps and the model is right about 90 per cent of the time at each. What is the chance of a correct final answer, and what follows?

A. About 75 per cent, because later steps correct earlier mistakes
B. About 90 per cent, because the steps are independent of each other
C. About 53 per cent, so shorten the chain and verify in code
D. About 53 per cent, which a reasoning variant removes entirely

5 Module 1 · Deciding to run it yourself

You paste a client's data into a locally hosted model to summarise it. What has changed about your data protection position?

A. The processing no longer counts, because no third party was involved
B. You have removed a processor and kept every duty as controller
C. The obligations transfer to whoever published the model weights
D. Consent is no longer required, because the data did not leave the country

6 Module 1 · Deciding to run it yourself

You grade thirty audition outputs knowing which model produced each one. What does that cost you?

A. Scores move towards whichever model you already favoured
B. The comparison becomes invalid unless you re-run at temperature 0
C. Only the borderline items are affected, which is a small minority
D. Nothing, provided the scale has only three points

7 Module 1 · Deciding to run it yourself

Which property do classification, field extraction and answering from retrieved passages share, that makes them the right shape for a small local model?

A. They tolerate a wrong answer, because a person checks the result
B. They can be run at temperature 0, which removes sampling error
C. They are short prompts, so prefill is never the bottleneck
D. The facts are in the prompt and there is one step

Module 1 · Deciding to run it yourself

The module starts on p. 11.

MODULE 1 TEST

Module 1 test, Q1, p. 52
An audition of thirty real items against a hosted copy of the exact open weights returns eleven items unusable. What does buying a faster graphics…
Answer A: Nothing at all
Why: The failures are a property of the weights. A faster machine runs the same weights more quickly and does not make them better, so eleven unusable items out of thirty is unchanged by any hardware. The useful next moves are the prompt, a newer or task-specialised model in the same size class, decomposing the task into steps you verify separately, or retrieval so the answer comes from a document rather than the model's memory.

Module 1 test, Q2, p. 52
You download SomeLab/CoolModel-8B. The uploader's README says MIT. The model card's first line says it was fine-tuned from a model released under Meta's community licence…
Answer B: Meta's terms, because a base licence follows its descendants
Why: A base model's licence follows everything made from it, whatever the uploader wrote in their own README. Meta's terms are not open source: there is a monthly-active-user threshold above which a separate licence must be requested, an acceptable use policy attached, and a naming requirement for derivative models. The two questions to ask of any download are what it was fine-tuned from and what that licence said.

Module 1 test, Q3, p. 52
You are on a slow connection and want to know in three minutes whether a model will fit and behave. Which two files tell you…
Answer D: config.json and tokenizer_config.json
Why: config.json carries the layer count, the attention head count and the KV head count, which is every number the memory arithmetic needs, plus the architecture name that says whether your engine supports the model at all. tokenizer_config.json carries the chat template, which is the most common cause of poor local output. Both are a couple of kilobytes, so they cost nothing to fetch before committing to gigabytes.

Module 1 test, Q4, p. 53
A machine has 64 GB of system RAM at about 30 GB/s and no useful GPU. Why does a 30B-A3B model generate several times faster…
Answer B: Because only the active parameters are read per token
Why: Memory holds the total parameter count, because the router may choose any expert at any token and all of them must be resident. Decode speed is bandwidth divided by the bytes read per token, and only the active experts plus the shared attention weights are read. About 2.5 GB instead of 18 turns a fraction of a token per second into ten or more, on the same machine, reading the same file.

Module 1 test, Q5, p. 53

You ask a freshly downloaded model for the capital of France and it replies with three more exam questions. What is the most likely explanation?

Answer C: You have the base variant, not the instruct one

Why: A base model is trained only to predict the next token over a corpus. It has no notion of a conversation, no instinct to answer a question and no instinct to stop, so the most plausible continuation of an exam question is more exam questions. It is not broken and it was never taught the format. Check tokenizer_config.json: a base model has no chat_template field, or a trivial one.

Module 1 test, Q6, p. 53

A team runs weights locally and wants evidence that nothing leaves the machine. Which check is the strongest?

Answer C: Blocking it at the firewall and confirming it still answers

Why: The weights running in your own process do not phone anyone; the application around them, its plugins, its hybrid features and its analytics might. Blocking the process entirely and finding that answers still arrive proves that nothing in the chain needed the network, and anything that breaks was reaching out for something. The bind address governs who may connect inbound, which is a different question with a different answer.

THE LESSON CHECKS

Lesson 1, p. 14

Someone runs Llama 3.1 8B locally and finds it noticeably worse than the same model served by a hosted provider. They conclude they need a...

Answer A: Same weights have the same ceiling, so check quantisation level, chat template, context truncation and sampler settings before buying anything.

Why: The weights set the quality ceiling; hardware only decides whether you can reach it and how fast.

B A larger GPU gives the model...: Treats VRAM as if it were model capability, when it only determines what you can load and how fast.

C Providers serve a larger private model...: Reaches for an unfalsifiable explanation instead of the four testable causes sitting in front of you.

D Quantisation always costs a large share...: Assumes 4-bit is a big loss, when a well-made 4-bit quant is usually a small and measurable one.

Lesson 2, p. 18

You fine-tune a Llama-family model on your own data and want to publish the result on Hugging Face under Apache-2.0, since your training code and…

Answer C: The derivative weights remain under the original community licence, which you cannot relicense, and it carries naming and use-policy conditions that pass to your users.

Why: "Open" splits into open weights, open source and open data — most famous models are only the first, and a base model's licence and use policy follow every fine-tune and quant made from it.

A Nothing, provided you credit Meta in…: Reduces a licence to a courtesy; the naming and downstream-terms clauses are obligations, not credits, and cannot be swapped for a mention.

B Fine-tuned weights cannot be published at…: Imports an unsettled copyright question as if it were settled law, and would prohibit the entire open fine-tune ecosystem.

D Apache-2.0 is the wrong choice only…: Invents a technical objection; Apache-2.0 does contain a patent grant, and the actual problem is that you never had the right to choose the licence.

Lesson 3, p. 22

Two 8B models advertise the same 128k context. Reading config.json, model A has 32 layers and 8 key-value heads; model B has 32 layers and…

Answer B: Model B's KV cache is four times larger per token, so at long context it will need far more memory than model A despite the same parameter count.

Why: The Files tab of a model card is verifiable and the prose is a claim — config.json, the shard sizes and the chat template tell you more in three minutes than any benchmark table on the page.

A Model B will be more accurate…: Treats a memory-layout choice as a quality guarantee; grouped-query attention was adopted because it costs little accuracy, and accuracy at length is not readable from head counts.

C They will behave identically, since KV…: Drops the key-value head count from the cache formula, which is the term that differs sixfold between model families.

D Model A will generate tokens roughly…: Confuses cache size with the dominant cost of decoding, which is reading the weights; the cache affects capacity and long-context speed, not the base rate.

GO UNDERNEATH

Running Models Yourself

Local and self-hosted LLMs, with the arithmetic instead of the marketing.

WHAT IT TEACHES	A practical course on running large language models on your own hardware. It teaches the arithmetic that decides whether a model fits, what quantisation actually costs you, which serving engine wins under which conditions, and when local is genuinely cheaper than an API — with real commands, real numbers, and no engine advertising.
WHAT IS INSIDE	Nine modules and 82 lessons, 28 figures drawn from data, nine case studies, nine module tests, a 63-question examination, and every answer with the reason behind it.
LICENCE	CC BY-NC-SA 4.0. Copy, print, share and adapt it for any non-commercial purpose, with credit, under the same licence. There is no paid edition.
THE COURSE	Free to read, with the lessons, the films and the examination, at **addaly.com/learn/running-models-yourself**

Scan to open the course

Addaly

First edition, September 2026 · build 62a26075