DAY 01–10 · Part 1 · What AI Is

Part 1 Review — What AI Is

The real terms, defined and then used · down to the mechanics and backstory

No test — a review to read through

Pick a piece, add it, pick again

1. AI doesn't make up a whole sentence at once. It picks just the next piece, adds it, then picks again.

2. Repeat that pick hundreds of times, and you get the long answer we read.

3. So AI isn't fetching an answer stored somewhere — it builds a new one every time.

Why one piece at a time?

1. Because that's exactly the job it got in training — cover part of a sentence and guess the next piece.

2. A machine that learned to guess well works the same way when it answers.

3. That's also why answers seem to trickle out bit by bit. That really is how they're being made.

This method's name, and its weak spot

1. This is called autoregressive generation — each piece it just made goes back into the input to pick the next.

2. The weak spot: it never goes back. Pick one odd piece early on, and it keeps building on top of it.

3. Most language models today stand on this setup. What's changed is how they pick — not the fact that they pick.

That piece has a name: token

1. The piece AI picks has a name. It's called a token.

2. A token isn't a letter or a word — common chunks stay whole, and unfamiliar ones get chopped up small.

3. To you, 'What's for lunch today?' is four words. To an AI, it's about six tokens.

Token count is time and money

1. The cutting rules are called a tokenizer. Each model has its own, so token counts for one sentence vary.

2. More tokens means more math to do — so it gets slower and pricier.

3. Languages like Korean often need more tokens than English to say the same thing.

How are the cutting rules made?

1. BPE (Byte Pair Encoding) is a common method: it builds a vocabulary by merging frequent pairs, over and over.

2. So even a never-seen word can be built from small pieces — words not in the vocabulary don't stump it.

3. The 'context length' coming up later is counted in tokens, not letters — tokens are the unit of all the math.

The probability table and the temperature knob

1. To pick the next piece, AI scores every candidate, then holds a raffle where higher scores get more tickets.

2. Temperature is the knob that sets how adventurous that raffle gets.

3. Turn it down and #1 wins every time; turn it up and lower candidates pop up more often.

When to turn it down, when to turn it up

1. Same question, different answer each time? That's not a mood — it's the raffle.

2. For jobs with one right answer (math, summaries, code), turn it down; when you want lots of ideas, turn it up.

3. Turning it up doesn't make it smarter — just more varied, and more likely to say something off the wall.

Where temperature fits into the math

1. Softmax turns scores into probabilities. Temperature divides scores first; higher means smaller gaps, flatter table.

2. Set temperature close to 0, and it basically just picks #1 every time.

3. Other knobs come along: top-k keeps just the top few; top-p keeps candidates until their odds hit a set total.

Training is tiny fixes, repeated

1. AI training isn't a sudden 'aha' — it's 'tiny fixes,' repeated.

2. Hide part of a sentence, have it guess the next piece, then nudge the numbers inside by how much it missed.

3. Repeat that an enormous number of times, and grammar and common sense soak into those numbers.

A lot like grading homework

1. It's close to giving every miss an 'off by this much' score, then shifting a little in that direction.

2. No big fixes all at once. Small ones — but a huge number of times.

3. That's why training takes huge amounts of time and electricity — and once it's done, the numbers are locked.

Loss, backpropagation, and locked numbers

1. Loss measures how wrong it was. Backpropagation works backward to tweak numbers and cut it — set out in 1986.

2. Once training is over, chatting with us doesn't teach the model anything. Those numbers stay put.

3. So 'Remember what I told you yesterday?' doesn't work. What looks like memory is another mechanism — DAY 07.

Parameters: how many numbers get tuned

1. How many numbers training tunes — that's the parameter count.

2. The more there are, the bigger the container for patterns.

3. Today's big models run from tens of billions to around 1 trillion — not a number you could ever count.

Just how big is this number?

1. Counting one per second, 100 billion would take you more than 3,000 years.

2. The human brain has about 100 trillion synapses, so models still fall short of that.

3. But bigger doesn't always mean smarter — what it trained on, and how much, has to come with it.

From a size race to a balance question

1. In 2020, the GPT-3 paper made 175 billion the yardstick for this race.
2. In 2022, the Chinchilla study showed balancing size with the amount of training data beats just scaling up.
3. So now people talk more about 'how was it trained?' than 'how many?'

Confidently making things up — hallucination

1. When AI calmly makes up facts that don't exist, it's called hallucination.

2. It isn't trying to lie. It comes naturally with a setup that picks a plausible next piece.

3. AI has no separate built-in way to flag 'I don't know this.'

Where it slips up most

1. It slips most on things rare in its training data, very recent events, and specifics: numbers, dates, names.

2. It sounds the same whether it knows or not. Looking confident doesn't make it true.

3. The habit of asking for sources and opening them yourself is the cheapest, surest defense.

Why we can only shrink it, not wipe it out

1. A 2023 survey sorts hallucinations in two: clashing with the input vs. clashing with real-world facts.

2. No way to fully remove it yet — people shrink it by adding search, requiring sources, or scoring confidence.

3. So asking in a way you can check — that's the skill on the user's side.

Not remembering — rereading

1. AI doesn't save your chat. Every time, it rereads the whole conversation so far, then answers.

2. There's a limit to how much it can read at once. That limit is called the context length.

3. As the chat gets longer, the early part gets pushed out of the window.

Too long is a problem — so is the middle

1. Sometimes 'I already told you!' doesn't work. What you said may already be out the window.

2. Even with room to spare, there's a catch — studies found the middle tends to count less than the start or end.

3. For really important info, the surest move is to point it out again later in the chat.

The window is measured in tokens

1. This window is measured in tokens (from DAY 02), not letters.
2. The 2023 'Lost in the Middle' study showed that info in the middle of a long input gets used less.
3. However long the window gets, reading isn't free — the longer it is, the slower and pricier.

New question, new probability table

1. A prompt is the whole input you hand to AI.
2. Change the words, and the probability table for the next piece changes too — so the answer changes a lot.
3. You're not changing the AI — you're changing the question.

Three things that work

1. Give it a role, spell out the conditions, name the format you want — these three work best.

2. 'A 100-word notice for middle schoolers, with one example' gets way better answers than 'Write something good.'

3. It's not a trick. You're just saying what you want more clearly.

What prompts can and can't do

1. A 2021 survey described prompts as an interface for getting the behavior you want without retraining.

2. Sometimes just changing how you ask — like having it solve step by step — changes how accurate it is.

3. But a prompt can't erase the hallucination from DAY 06 — it can only shrink it.

Size means parameter count

1. A 'big' model just means it has lots of parameters — the ones from DAY 05.

2. Big models know a lot and can handle hard problems, but they're slow, pricey and only run on servers.

3. Small models fit on your own device, so they're fast and work even without internet.

Which job goes to which size?

1. Quick, frequent, light jobs go small; jobs that need long thinking and lots of reading go big.

2. You can also teach a small model, using a big model's answers as its textbook.

3. Then the small model punches above its weight — but it can't fully replace the big one.

Distillation — passing the learning down

1. This is called distillation, and Geoffrey Hinton's team formalized it in 2015.

2. A lot of the AI running on phones today came this way.

3. It's not better vs. worse, it's placement — where it'll be used decides the right size.

Where the core lives

1. A closed model's core stays on the company's servers. We can only talk to it through the front desk.

2. With an open model, you can download that core — the weights — in full and run it on your own computer.

3. You ask questions the same way — the difference is where the model lives.

Splitting 'open' into four pieces

1. 'Open' isn't one thing. Split it into four: weights, code to run it, training data, the right to change it.

2. Most open up the weights and code, but not the training data.

3. So don't stop at the word 'open' — ask which pieces are actually open.

Where the license draws the line

1. Llama 2 (2023) even allowed commercial use, but its custom license sparked an 'is this open source?' debate.

2. The same year, Mistral 7B went with the widely used Apache 2.0 license as is, dropping nearly all limits.

3. 'Open' doesn't mean free — the hardware to run it, and the updates, are on whoever downloads it.

Ten days, one story.
A machine that picks the next piece.



In Part 2: where this thing came from — all the way back to the neuron that learned on its own, in 1958.