Zero to One: Understanding AI

Zero to One: Understanding AI

You've probably heard the phrase "neural network" a hundred times without anyone actually telling you what it is. You've also probably heard that AI needs chips, that Nvidia is suddenly one of the most valuable companies on earth, and that tech companies are building data centers the size of small towns. What you probably haven't heard is why. Why does software need a specific kind of chip? Why does a chip shortage become a national conversation? Why does training a model require something closer to a power plant than a computer?

None of that makes sense until you understand what's actually happening inside the machine. So start there.

What a neural network actually is

Forget the word "intelligence" for a second. A neural network, at the level of what's physically happening, is a big pile of numbers. Picture something like a dial on a stereo, except there are billions of them, and each one is set to some value. Those numbers are called parameters. When you type a sentence into an AI chatbot, your text is first broken into chunks called tokens — roughly word-sized pieces — and converted into numbers. Those numbers run through the whole pile of dials, and out the other end comes something like a ranked list of guesses for what token should come next, each with a probability attached. The model usually picks one of the likely candidates, feeds it back in, and repeats — token by token — until you have a full response.

The question, obviously, is how those dials get set to values that produce something coherent instead of gibberish. Nobody sits down and decides them by hand; there are far too many. Instead, throughout training, the model is shown text with a piece hidden and asked to predict it — not just the last word of a sentence, but the next token at essentially every position in enormous amounts of text at once. Early on, before any training has happened, the dials are random, so the guesses are garbage. But you can measure exactly how wrong a guess was, and there's a well-understood method (it's called backpropagation, if you want the term) for nudging every single dial a tiny amount in whatever direction would have made that particular guess slightly less wrong. Do that once and nothing changes. Do it a few trillion times, across a large chunk of everything humans have ever written down, and the dials settle into a configuration that is startlingly good at continuing text in ways that look like reasoning, coding, argument, and explanation.

Researchers designed the architecture — the layout of the dials and how signals flow between them — and they designed the training process that adjusts those dials. But nobody designed the specific capabilities that emerged from running that process at scale. Those were discovered by trial and error, at a scale no human could do by hand, by the training process itself. Which is genuinely strange if you sit with it: something that can walk you through a legal contract or debug your code runs on rules that no person wrote line by line. The people built the machine that finds the rules; the machine found the rules.

That word, training, is the key to everything that follows. It's also where all the chips and buildings and electricity come in.

Why this keeps getting more capable, on schedule

Here's the fact that actually explains why the whole industry looks the way it does right now. Take that same training process and make it bigger in three ways at once: more dials, more training data, more raw computation. Researchers have found that the model doesn't just get a little better — it gets measurably better along a curve that can be plotted and, within a range, predicted in advance. This pattern, known as a scaling law, was documented in detail starting around 2020 and has held up remarkably well since. It isn't a law in the sense of physics — it's an empirical regularity, and one that a few research teams have shown can bend depending on how compute is split between model size and training data. But it has been robust enough, for long enough, that companies now plan multibillion-dollar bets around it.

You can see the underlying trend by lining up recent generations of models. Public benchmarks and released research show real, measurable jumps in ability from one generation to the next — from struggling to hold a paragraph together, to writing fluent prose, to passing professional exams and sustaining complex, multi-step reasoning. Scale — more parameters, more data, more compute — has been the dominant driver of that improvement. But it wasn't the only one: architectural changes (like the transformer itself), better training data curation, and post-training techniques such as instruction tuning and reinforcement learning from human feedback all meaningfully shaped how capable and how usable these models became. Scale set the ceiling; the other techniques determined how much of that ceiling was actually reached.

This is a big part of why companies are racing to build bigger models rather than waiting around for a single clever new idea. Scale reliably produces capability, even if it isn't the whole story. So the question every AI company is actually asking is: how do we get more scale, faster, than everyone else? And that question turns out to be an industrial question, not just a software one.

Why this matters: the near-straight line on a logarithmic axis is the visual signature of the scaling law itself. Training compute for notable models has grown by roughly an order of magnitude every one to two years for over a decade — the underlying force pulling everything else in this article (chips, clusters, power, data centers) along with it.

Why "more scale" means more chips

Training a model means running that dial-adjustment step trillions of times, and each step involves an enormous volume of a specific kind of arithmetic: matrix multiplication, applied over and over to large blocks of numbers at once. A normal computer chip, the kind in your laptop, is built to do one complicated task quickly, in sequence. That's a poor fit here. What you want instead is a chip built to do millions of simple, identical calculations simultaneously — the essence of parallel computing.

Such a chip already existed, for a completely different reason: the graphics card, originally built to render video game pixels. Rendering a scene means calculating the color and lighting of millions of pixels at once, which relies on the same underlying operation — massively parallel linear algebra — as training a neural network. That coincidence, gaming hardware turning out to be nearly ideal for AI, is the accident sitting underneath the entire modern AI industry. Nobody designed graphics cards with AI in mind. They just happened to be the right shape for the job when the job showed up.

Nvidia makes the leading version of that chip, and that part is common knowledge by now. What's less appreciated is that a large part of Nvidia's advantage isn't the chip at all — it's CUDA, the software platform and ecosystem of tools, libraries, and code built on top of the hardware over nearly two decades. Most

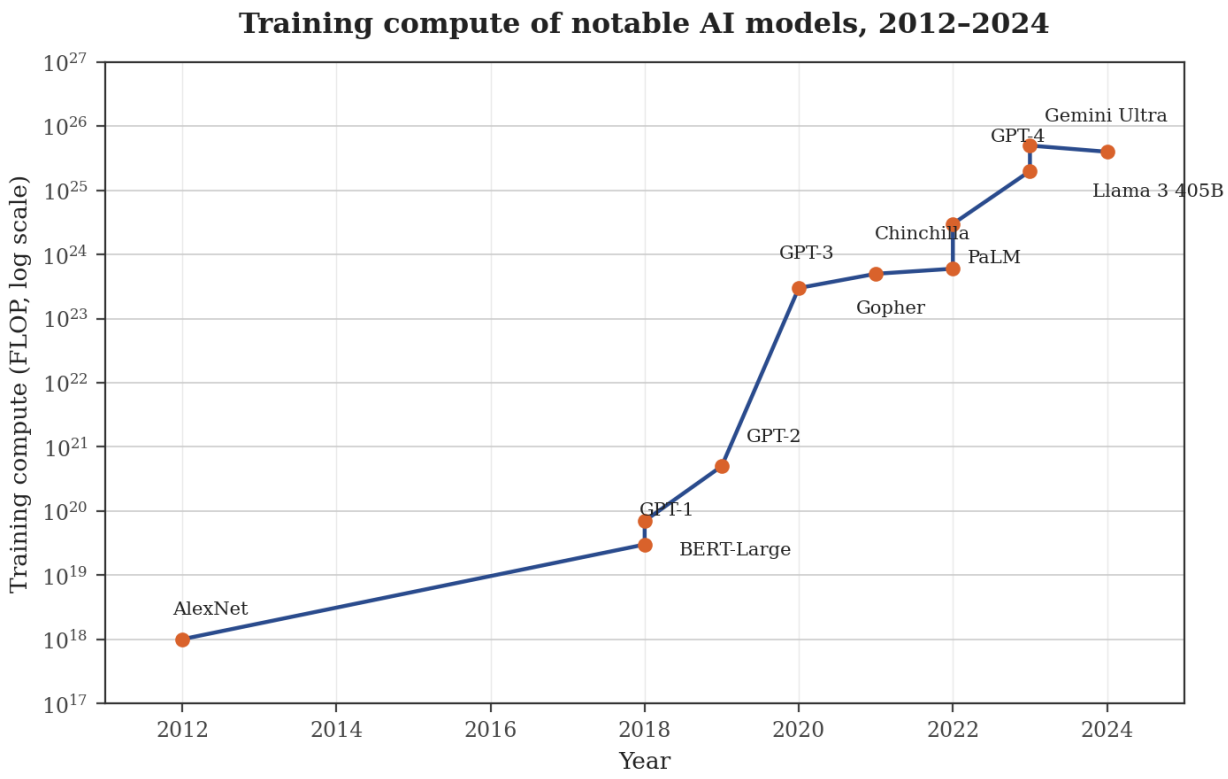

Figure 1: Training compute of notable AI models, 2012–2024

major AI frameworks and research code are deeply optimized for it. Moving away from Nvidia hardware doesn't just mean buying a different part; it means rebuilding software that was written and tuned against that ecosystem, in some cases across an entire industry's worth of tooling. And a single chip, however good, does almost nothing on its own. Training a frontier model means tens of thousands of these chips wired together closely enough to behave like one giant computer — its own extremely hard networking and engineering problem, and one Nvidia also sells tools to solve. That's why the chip shortage isn't really about a factory failing to make enough chips fast enough. It's about a company that controls both the hardware and much of the surrounding software ecosystem the industry has built itself around.

Why "more chips" means a data center

Here's where it stops being about any one company's product and starts being about physical infrastructure. Tens of thousands of chips, all running at full power, all day, generate an enormous amount of heat and draw an enormous amount of electricity. Large training clusters now draw power on the scale of hundreds of megawatts to over a gigawatt — comparable to a mid-sized city — continuously, for months at a time. Not a large office building's worth of servers. Something closer to a power plant with computers attached to it.

That single fact explains most of what you're currently seeing in the news. It explains why AI companies are signing deals for nuclear, gas, and other dedicated power sources and building their own electrical substations. It explains why cooling has become its own engineering discipline, with companies increasingly turning to liquid cooling because moving air alone struggles to keep up with the heat that tens of thousands of densely packed chips produce. It explains why a data center built for AI isn't just a warehouse full of computers — it's a purpose-built facility with its own power supply, its own cooling system, and networking fast enough that all those chips can exchange data constantly without slowing each other down. And it explains why, when a chip or a network link fails partway through a months-long training run — which happens routinely at this scale — the whole system has to be engineered with checkpoints and redundancy, so the run can pick back up instead of restarting from scratch.

A note on this chart: only the August 2024 data point (Colossus 1) is a measured anchor. Everything else is an extrapolation of Epoch AI's tracked ~7-month doubling rate for the record-holding AI data center, shown here as a dashed projection rather than a settled multi-year trend — it illustrates the *shape* of the growth curve, not a guaranteed forecast for any specific facility.

Putting the whole picture together

Start from the bottom and the whole conversation snaps into focus. A neural network gets more capable through a training process that's really just repeated, small corrections applied billions of times. That correction process happens to be a task well suited to a specific kind of chip — one invented for video games and repurposed by accident. That kind of chip is made best, and is hardest to switch away from, by one company in particular, largely because of the software built up around it over two decades. And using enough of those chips to train something genuinely capable requires a purpose-built facility with dedicated power, because ordinary buildings and ordinary electrical grids were never designed to run tens of thousands of chips at once.

None of it is arbitrary, and none of it is hype dressed up as infrastructure. It's the physical cost of the bet this entire industry has made: that a bigger version of the same training process — reinforced by real but secondary improvements in architecture and training technique — keeps producing a more capable machine. Whether that bet keeps paying off at the same rate is a genuinely open question among researchers; scaling has slowed and shifted before, and it may again. But now you know what the bet actually is, and why, for now, it costs as much electricity as a small country.

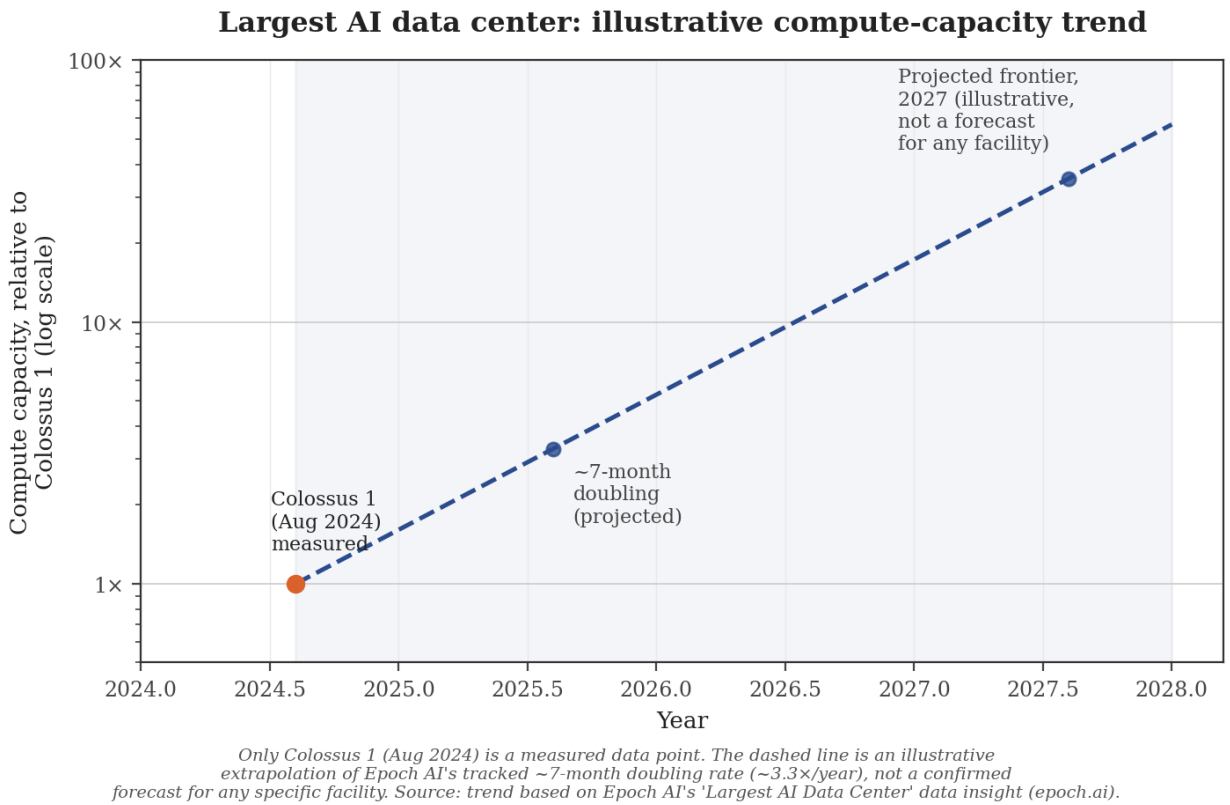

Figure 2: Largest AI data center: illustrative compute-capacity trend