DAY 11–20 · Part 2 · How We Got Here

Part 2 Review — How We Got Here

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

No test — a review to read through

A neuron machine that learned on its own

1. In 1958, Frank Rosenblatt built the perceptron.

2. It's one artificial neuron: multiply each signal by a weight, add them up, output 1 if the sum clears a threshold, else 0.

3. The real breakthrough: when wrong, it fixed its own weights — the first machine to learn from examples, not written rules.

Why is this the starting point?

1. Take today's giant AI apart and you'll find huge numbers of these neurons, stacked and wired together.

2. 'Multiply signals by weights and add them up' and 'fix the weights when wrong' — those two ideas haven't changed.

3. McCulloch and Pitts wrote up neurons as logic in 1943, 15 years earlier, but the perceptron was the first real machine to learn on its own.

Now in the real terms

1. In the real terms, the perceptron multiplies each input by its weight and adds them up (the weighted sum), adds a bias, and a step function outputs 1 if that's above 0.

2. The learning rule is simple — when it gets an example wrong, nudge the weights a little toward the right answer.

3. A one-layer model like this only learns properly if one straight line can split the data — a condition called linearly separable, and exactly where the next day's story starts.

When hopes outrun skill, winter comes

1. In 1969, Minsky and Papert proved mathematically that one layer of neurons can't solve some problems. Money and people left — the first winter.

2. In the 1980s, expert systems with hand-written rules boomed but failed just off-script; in 1987 their special-computer market crashed — the second winter.

3. Both times, it wasn't that AI was bad — the promises ran ahead of its skill.

The paper that melted the winter

1. The first winter came from the limits of a single layer.
2. Stacking layers would work, but there was no widely used way to tell which weights to fix, and by how much, across layers.
3. A 1986 backpropagation paper in Nature spread the word that stacked layers' weights can be fixed, and the winter began to thaw.

XOR and backpropagation

1. The classic problem one layer can't solve is XOR — output 1 when exactly one of two inputs is 1; no straight line splits its four points, so it isn't linearly separable.

2. Adding just one more layer solves it, but then you have to work out how much each middle-layer weight is to blame for the error.

3. Backpropagation uses the chain rule from calculus to hand the output's error backward, starting from the last layer, and nearly all neural network training today stands on it.

The day three things came together

1. In 2012, a University of Toronto team's neural network won the ImageNet photo-guessing contest with a 15.3% error rate.

2. Second place had 26.2%, so the gap was more than 10 percentage points.

3. Hardly anything was new: millions of photos, gaming graphics cards and deep neural networks came together for the first time that year.

Why 2012, of all years?

1. ImageNet is a database of millions of labeled photos, released in 2009.

2. Gaming graphics cards do lots of the same math at once, a perfect fit for training neural networks; the winners trained on two for 5–6 days.

3. With every ingredient in place, an old idea finally delivered, and from then on, image research went all-in on neural networks.

What was deep about it?

1. The winner was a convolutional neural network (CNN), strong on grid-shaped input: it slides small filters across the whole photo to find features like edges and patterns.

2. The deeper the stack, the more the features build up: early layers spot lines and edges, later ones spot bigger chunks like eyes or wheels.

3. 'Top-5 error' is how often the right answer isn't among the model's five guesses, so 15.3% means the answer was in its top five for about 85 of every 100 photos.

Words started looking at each other

1. In 2017, a Google team's paper, 'Attention Is All You Need', set out the Transformer — the backbone of today's AI.

2. Before, sentences were read front to back, one word at a time; the Transformer let every word look at all the others at once.

3. Looking all at once means computing all at once, so it was a perfect match for graphics cards.

What does 'looking' mean?

1. In 'The animal didn't cross the street because it was too tired', 'it' is the animal; change it to 'too wide' and 'it' is the street.

2. The Transformer scores how much each word should look at every other word, so it finds links like that in one go.

3. Reading in order, far-apart words got blurry; once words look at each other directly, distance stops mattering.

Inside self-attention

1. This setup is called self-attention.

2. Each word makes three vectors (lists of numbers): a query ('what I want'), a key ('what I have') and a value ('what I hand over'); the better a query fits a key, the more of that word's value it takes.

3. The recurrent neural networks (RNNs) before it had to finish one word before the next, so they were hard to parallelize; self-attention computes every pair at once with matrix multiplication — what graphics cards do best.

Scaled up, it copied from examples alone

1. Same Transformer backbone, just made bigger: in 2020 that gave us GPT-3, with 175 billion numbers inside.

2. The surprise: no retraining needed — put two or three examples in the question and it followed along right away.

3. From then on, 'what you tell it to do' — the prompt — started to really matter.

How much bigger did it get?

1. GPT-1 had about 117 million, GPT-2 about 1.5 billion, and GPT-3 175 billion.

2. That's more than 1,000 times bigger between 2018 and 2020.

3. Earlier research had predicted how much performance would rise as the numbers grew; GPT-3 actually pushed that math all the way.

In-context learning and scaling

1. GPT-3's trick is called in-context learning — without changing a single weight, it picks up the rule on the spot just from the examples in its input.

2. Depending on how many examples you give, it's called zero-shot (0), one-shot (1) or few-shot (a few) — that's the 'Few-Shot' in the paper's title.

3. The finding that performance improves along a predictable curve as model size, data and compute grow is called the scaling laws — it became the case for investing in 'make it bigger'.

A chat window is what changed the world

1. ChatGPT, opened to everyone on November 30, 2022, was a GPT-family model put in a back-and-forth chat window.

2. A big part was polishing it with people's picks of the answers they liked better, so it actually got what you asked.

3. So what changed that day was less the tech than who got to use it.

The test where a small model beat a big one

1. In the 2022 paper on that polishing method, people preferred the answers of a model with 1.3 billion numbers, tuned with human feedback, to those of GPT-3 with 175 billion.

2. It was a sign that size isn't the only answer.

3. Add a window where you just talk — no need to know about adding examples — and only then did everyone get their hands on AI.

The polishing called RLHF

1. Polishing with human feedback is called RLHF (reinforcement learning from human feedback).

2. First it's tuned on model answers people wrote; then people rank several answers to one question, that data builds a reward model scoring 'which answer is better', and the language model is retrained to push that score up.

3. Getting a model to act the way people intend like this is called alignment, and research into making answers useful and harmless really got going here.

Work it out before you answer

1. AI got much better at hard problems not by getting bigger, but by working through steps before answering.

2. At first, people had to say 'think step by step'; later came reasoning models trained to do it on their own.

3. The catch: longer thinking means slower answers and more text, so it's not worth it for easy questions.

factnotes

By the numbers

1. A 2022 paper showed that just adding the line 'Let's think step by step' to the end of a question makes math scores jump.

2. In 2024 came a model trained to write out its working unprompted: on the same US high school math contest, an instant-answer model got 12% and this one got 74%.

3. That came from changing the order of answering, not the backbone.

Chain-of-thought and the cost of thinking

1. Having it write out the working first is called chain-of-thought.

2. A language model writes one token at a time, so writing down in-between results lets it look back at them when picking the next token — scratch paper instead of mental math.

3. Reasoning models tend to solve more hard problems the more they think before answering; this compute spent at answer time is called test-time compute, and it became a second dial besides making the model bigger.

Grabbing a calculator when you don't know

1. AI's main job is adding the next word, so it's weak at exact things like big multiplications or today's weather.

2. So it got tools — a calculator, search, running code — and from 2023, models chose by themselves when to grab one.

3. A good AI isn't one that knows everything — it's one that notices it doesn't know and grabs a tool.

Four steps to grabbing a tool

1. It gets the question, decides if it needs a tool, calls the tool in a set format, and finishes the answer with what comes back.

2. Step two is the key — it used to take a person saying 'search for it', but now the model decides by itself.

3. In 2023, the Toolformer paper had a model teach itself when to call a calculator, search or a calendar; that year, models officially got a set format for naming a tool and its values.

Function calling and agents

1. This format for a model to call a tool is called function calling.

2. The model just writes 'which function, with which values' as text in a preset structure; an outside program runs it and feeds the result back — the model isn't pressing calculator buttons itself.

3. A system that chains these steps many times — planning on its own, calling tools and checking the results — is called an agent, and most of today's 'hand it a task' AI is built this way.

Chop anything into pieces, line them up

1. AI didn't learn to see photos and hear sounds by getting eyes and ears.

2. Photos are cut into square grid pieces and sound into tiny time slices, then lined up in one row with the text pieces.

3. Then no new backbone is needed — just feed them into the Transformer, where pieces look at each other.

Why learn them together?

1. At first, a photo was written out as text separately, then handed to a language model.

2. Models that learned text, images and sound together in one backbone from the start skip that middle step: faster and more accurate.

3. Photo reading opened up widely in 2023; in 2024 a model trained on all three together cut spoken replies to 0.32 seconds on average.

Patches and multimodal

1. The square pieces you get by cutting a photo into a grid are called patches.

2. Turn each piece into a list of numbers and tag it with where it sat in the photo, and to the Transformer it's no different from a word token.

3. Handling different kinds of input together like this — text, images, sound — is called multimodal; speed and accuracy depend on whether a model learned them together from the start or had separate parts joined up.

From using one to picking one

1. In late 2022 you could count good chat AIs on one hand; now dozens of groups in the US, Europe, China and Korea make their own.

2. The biggest change: open models, whose whole bundle of numbers is released for anyone to download and run on their own gear.

3. So now you don't just use one AI — you pick, weighing price, speed and what each one is good at.

How the closed door opened

1. Models anyone could download grew from 65 billion in February 2023 to 70 billion that July, then 405 billion in July 2024.

2. That's six times bigger in a year and a half — anyone could now get a model bigger than GPT-3's 175 billion.

3. In 2025, a Chinese research team openly released a reasoning model too — so the open-model race isn't just one country's thing.

What open weights open — and what they don't

1. The 'bundle of numbers' here is the full set of weights fixed by training, and a model that releases them is called an open-weight model.

2. With open weights, anyone can do extra training on their own data (fine-tuning), or shrink the model down to run on a laptop.

3. But opening the weights doesn't mean opening the training data or code too, so to see how far 'open' goes, you have to read each model's license.

Ten days, one road.
From one neuron to everyone's models.



In Part 3: what you can do with all this — starting with summarizing and translation, hands-on, one at a time.