DAY 46–55 · Part 5 · Looking Inside

Part 5 Review — Looking Inside

From one dot to the phone in your pocket · with the real terms · DAY 46-55

No test — a review to read through

One dot can only multiply and add

1. A neural network is dots passing numbers around, stacked in layers. A dot is called a node, one tier a layer.

2. Each dot multiplies every incoming number by a **weight** and adds them all up. That total is the weighted sum.

3. Then it pushes the total through a function that acts as a threshold and hands it to the next layer.

What the threshold does – the activation function

1. That threshold function is called an activation function. It decides whether a signal passes straight through, bends, or stops.

2. Without it, stacking any number of layers ends up the same as **one multiply-and-add**.

3. That's because stacking only straight, proportional math collapses into a single straight calculation.

Where weights come from – backpropagation

1. The first weights are arbitrary. The network answers, then walks backward through its own error to fix them.

2. Sending the error from back to front and nudging each weight is called **backpropagation**, made widely known in 1986.

3. Repeat that an uncountable number of times and the numbers left behind are what it learned.

Attention is a share you split up

1. Attention builds a query, a key and a value for every word, then matches queries against keys to get scores.

2. The step that turns those scores into **shares adding up to 1** is called softmax.

3. Because they are shares, the total is fixed - stare hard at one word and the rest must weaken.

Why this settles the meaning

1. When "it" has a query that matches some earlier noun's key, the staring piles onto that spot.

2. The values mix in by those shares, producing **a new number bundle with the context blended in**.

3. That's why the same word comes out as a different value in a different sentence.

From reading in a line to looking all at once

1. A 2014 translation study used it first, as a device against losing distant words in long sentences.

2. The 2017 paper dropped the read-in-order part entirely and **kept only attention** - the Transformer.

3. Looking all at once made the math easy to split across machines, which is how models got big.

Meaning gets turned into a place

1. A word turned into a bundle of numbers is called an embedding. One bundle holds hundreds to thousands of numbers.

2. The learning rule is simple - **words with similar company get similar numbers**.

3. Nobody writes the meanings in. It works them out while reading an enormous amount of text.

Once it's a place, it's math

1. Once meaning is a place, the distance between two words can be calculated.

2. A 2013 study showed that **even the direction between places carries meaning** - take man away from king, add woman, and you land near queen.

3. That directions hold relationships is why the method stayed in use so long.

From whole words to pieces

1. Models today don't use whole words; they cut text into pieces called tokens.

2. Chopping unfamiliar words into pieces made **the unknown-word problem** much smaller.

3. That's why length gets counted in tokens rather than letters.

It measures meaning, not letters

1. Sentences can become embeddings too. Measure the angle between two bundles and you see how much the meanings overlap.

2. The common yardstick is **cosine similarity** - it drops the size of the bundle and compares direction only.

3. Same direction lands near 1; unrelated falls toward 0.

Counting letters and measuring meaning are different

1. "That was fun" and "that was no fun" share almost every letter, but their directions are far apart.

2. The other way around, with no letters in common at all, **same meaning still comes out close**.

3. Those two cases are exactly where the old letter-overlap methods kept going wrong.

How to build good sentence bundles

1. A 2019 study put out a way to train two sentences onto the same yardstick.

2. That let people turn millions of text pieces into numbers in advance and, when a question arrives, **pull out only the close ones**.

3. That pulling-out is the first half of answering with the book open, the next day's lesson.

Look it up and read it, instead of reciting

1. Laying found text open beside the question is called retrieval-augmented generation, RAG for short.

2. The question becomes an embedding, **the close text pieces** get pulled out, and those pieces go into the model with the question.

3. The model reads what's in front of it instead of what it memorized.

Why inventing drops off

1. Inventing happens when it tries to fill a gap it doesn't know with memory.

2. With the source text in front of it there's less gap to fill, and it can show **where the words came from**.

3. With a source attached, the reader can check it themselves.

The weak spot is the finding

1. Not having to retrain the model is the big win - **swap out only the text you lay open** and the answers follow.

2. But if the piece it found is off-target or wrong, the answer goes wrong with it.

3. That's why the real effort goes into the finding rather than the answering.

Nudge a grown model instead of raising a new one

1. A model that did the big round of learning is called a base model, and teaching it one more job on top is fine-tuning.

2. Rather than changing every number inside, the common approach changes **only a tiny slice**.

3. In cost it is a completely different thing from training from scratch.

LoRA – cutting down the numbers you touch

1. LoRA, from 2021, freezes the original numbers and bolts two small plates on beside them, training only those.

2. That cut the numbers it touches down to **one ten-thousandth**.

3. Swap the bolted-on plates and one base model can do several different jobs.

Which to use – tuning or laying it open

1. Fine-tuning is strong on **tone, format and the way a job gets done**.

2. Putting newly arrived facts in is the job of answering with the book open.

3. They aren't rivals but different roles, so in practice they often run together.

Not the right answer, just which one is better

1. You have the model produce several answers to one question and let a person pick the better one. That record is called preference data.

2. The picks are used to build a **reward model** first - a small model that guesses the score a person would give.

3. Tuning the main model to push that score up is the reinforcement learning step.

Why not use an answer key

1. A good answer can't be written down as one thing. "Be kind" or "keep it short" is hard to turn into a rubric.

2. So instead of writing a score, you only ask **which of the two is better**. That's a far easier question for a person.

3. That's what it means to say you collect preference rather than right-and-wrong.

The effect, and what's left

1. In a 2022 study, people preferred this tuned small model's answers over those of a model **more than 130 times its size**.

2. It's a rare case where tuning made a bigger difference than scaling up.

3. But whose preference you collected stays baked in. That's why 2022 also brought a method that writes the rules down and has AI do the scoring.

The shape of the math was the same all along

1. A graphics card (GPU) is a part built to do easy math a huge number of times at once. That's called parallel computing.

2. Painting millions of dots on a screen at once needed **thousands of simple workers rather than a few clever ones**.

3. Inside AI training is matrix multiplication - trillions of easy multiplications and additions, all at once.

One card by the numbers

1. One card of the kind used for training holds about **80 billion** tiny switches called transistors.

2. That single card does roughly 1,000 trillion calculations a second.

3. Today's cards also take the route of handling numbers at a coarser scale to get more math done.

Because one card isn't enough

1. Llama 3 was built by tying **16,000 cards** into one block and running it for 54 days.

2. A block like that is called a cluster, and the speed of the links between cards can decide performance too.

3. In 2024 one place said it had tied together 100,000 cards.

One is tiny, the pile is not

1. In a measurement released in 2025, the median electricity for one text question was **0.24 watt-hours**.

2. That's about a TV left on for 9 seconds, so on its own it is a small amount.

3. That said, it is a company-reported figure on the company's own terms, and the value shifts with what you count.

Looking at the whole building

1. Data centers worldwide used **415 terawatt-hours** of electricity in 2024 - about 1.5% of all electricity.

2. The International Energy Agency expects 945 terawatt-hours by 2030, more than double.

3. 1.5% looks small, but things that double in six years are not common.

So where should you look

1. The per-question figure is **dropping fast**, because the field keeps moving toward getting the same answer with less math.

2. But if the number of questions grows faster, the total keeps rising anyway.

3. When you read these numbers, start by not mixing up one question with the total.

It splits the work into pieces

1. Today's phones carry a small model of about **3 billion numbers**. Running inside the phone is called on-device.

2. Even with no internet it can summarize, polish text and tidy up notifications.

3. It differs in size from the model on the server by tens to hundreds of times.

How it fits on a phone

1. It uses **quantization**, storing the big model's numbers at a coarser scale. Fewer digits also means more speed.

2. When memory runs short, loading only the part needed at that moment is used alongside it.

3. Still, shrinking costs it some jobs, so not everything goes on the phone.

How the pieces get split

1. The default is that the phone tries first and sends out only the pieces too heavy for it.

2. Finishing inside means **faster, less electricity, and your words never leaving**.

3. Jobs that need that building from before are still there. Mixing the two is today's answer.

The inside is math, not magic.
Aiming it and running it is all of it.



Part 6 looks at what comes next.