MY WEIRD PROMPTS

Podcast Transcript

EPISODE #176

The Math of Magic: Decoding AI Weights and Tensors

Published January 06, 2026 • Runtime: 22:49

https://myweirdprompts.com/episode/ai-weights-tensors-explained/

EPISODE SYNOPSIS

Ever wondered what "weights" actually are in a neural network? Join Corn and Herman as they demystify the gears and pulleys behind AI, from the massive scale of tensors to the precision of fine-tuning. They explore how billions of numerical "knobs" are turned to capture human knowledge and why these models are more like holograms than databases. It’s a deep dive into the math that makes the magic possible, with a side of questionable focus-enhancing headwear.

DANIEL'S PROMPT

Daniel

"What are weights in AI models, and what’s happening 'under the hood' when we're training or fine-tuning a model to create or adjust them?"

TRANSCRIPT

Corn

Hey everyone, welcome back to My Weird Prompts. I am Corn, and I am sitting here in our living room in Jerusalem with my brother, looking out at a surprisingly clear January morning.

Herman

And I am Herman Poppleberry, ready to dive into the digital deep end. It is January sixth, twenty twenty-six, and the world of artificial intelligence is moving so fast it feels like we are living in a science fiction novel some days.

Corn

It really does. And speaking of moving fast, our housemate Daniel sent us a voice note this morning that hits on something fundamental. He was asking about weights in artificial intelligence models. He mentioned seeing terms like tensors and safe tensors on sites like Hugging Face and wanted to know what is actually happening under the hood when these models are trained or fine-tuned.

Herman

That is a fantastic prompt. It is one of those things where we use the word weight all the time in the industry, but if you stop to think about it, it is a bit of a weird metaphor. Like, is the model heavy? Does it have mass? Obviously not, but the math behind it is what gives the model its intelligence.

Corn

Right, and I think for a lot of people, these models feel like magic boxes. You put text in, you get text out. But Daniel is pointing toward the actual gears and pulleys inside. So, Herman, let us start with the basics. When we talk about a weight in a neural network, what are we actually looking at?

Herman

At its simplest level, a weight is just a number. That is it. It is a numerical value that determines how much influence one piece of information has on another piece of information as it moves through the network. If you imagine a neural network as a series of connected pipes, the weights are like the valves. They control how much water, or in this case, data, flows from one section to the next.

Corn

Okay, so if a weight is high, that connection is strong?

Herman

Exactly. If a weight is close to one, it means that the input is very important for the output of that specific neuron. If the weight is close to zero, the network is basically saying, hey, ignore this specific detail, it does not matter for the prediction I am trying to make. And if it is negative, it actually inhibits the signal. It is a way of saying that the presence of this feature makes the outcome less likely.

Corn

So when we hear that a model like GPT-four or some of the newer open source models from late twenty twenty-five have hundreds of billions or even trillions of parameters, those parameters are mostly these weights?

Herman

Precisely. A parameter is usually just a catch-all term for weights and biases. When someone says a model has one hundred billion parameters, they are literally saying there are one hundred billion individual numbers that have been fine-tuned to help that model understand language or images. It is a staggering amount of data when you think about it. One hundred billion little knobs that all have to be turned to exactly the right position for the model to make sense.

Corn

That is a great image. One hundred billion knobs. But Daniel also mentioned tensors. I know that word comes up a lot, especially with Google's Tensor Processing Units or libraries like TensorFlow. How do tensors fit into the weight conversation?

Herman

Think of a tensor as the container for the weights. In mathematics, a scalar is just a single number. A vector is a list of numbers. A matrix is a grid of numbers. A tensor is the generalization of that. It can be a multidimensional grid. So, instead of having one hundred billion individual loose numbers, we organize them into these massive, high-dimensional blocks called tensors. This allows the computer, specifically the Graphics Processing Unit or the Tensor Processing Unit, to do math on thousands of weights at the exact same time.

Corn

So it is about efficiency. Instead of adjusting one knob at a time, the computer can essentially adjust a whole panel of knobs simultaneously because they are grouped together in these tensors.

Herman

Exactly. It is like the difference between painting a wall with a tiny artist's brush versus a massive industrial paint sprayer. Tensors allow for that massive scale. And when Daniel mentioned safe tensors, that is actually a newer standard for storing these files. In the old days, like way back in twenty twenty-two and twenty twenty-three, people used a format called pickle. But pickle files could actually run malicious code on your computer when you opened them. Safe tensors, as the name suggests, just contains the numbers. No code, no risk, just the raw weights.

Corn

That is a good bit of history. It is wild to think of twenty twenty-two as the old days, but in AI years, that is practically the Bronze Age. So, we have these weights, which are numbers in a tensor, and they act like valves or knobs. But how do they get their values? When we talk about training a model, what is actually happening to those numbers?

Herman

This is where the magic happens, Corn. Imagine you have a model with totally random weights. It knows nothing. If you ask it to complete the sentence, the cat sat on the, it might say, the cat sat on the refrigerator-door-bicycle. It is nonsense because the knobs are all turned to random positions. Training is the process of showing the model millions of examples of real sentences and slowly adjusting those knobs so that it learns the patterns of human thought.

Corn

And that is done through something called backpropagation, right? I remember we touched on this in episode two hundred and fifteen when we talked about gradient descent.

Herman

You have a great memory. Yes, backpropagation is the heart of it. It works like this. The model makes a guess. We compare that guess to the actual correct answer. The difference between the guess and the truth is called the loss. We want the loss to be as small as possible. So, we use calculus to figure out exactly how much each of those billions of weights contributed to the error.

Corn

So, we work backward from the mistake?

Herman

Exactly. We go from the output layer back to the input layer. We say, okay, this weight over here was too high, which made the model think the word was bicycle instead of mat. Let us nudge it down a little bit. And this weight over here was too low, let us nudge it up. We do this billions of times across millions of documents until the weights converge on values that actually represent the structure of language.

Corn

It sounds incredibly computationally expensive. I mean, doing calculus on a hundred billion variables over and over again?

Herman

It is. That is why the big labs spend hundreds of millions of dollars on electricity and hardware. But what is interesting is that once those weights are set, the model is essentially a frozen snapshot of knowledge. It does not need to do the calculus anymore to talk to you. It just runs the data through the existing weights. That is called inference.

Corn

So inference is just the water flowing through the pipes that are already set. But then we have fine-tuning, which Daniel also asked about. How does that differ from the initial training?

Herman

Fine-tuning is like taking a master chef who knows how to cook everything and then giving them a week of intensive training specifically on making sourdough bread. You are not teaching them how to use a knife or how an oven works. They already have those weights set. You are just taking a pre-trained model and showing it a smaller, more specific dataset to nudge the weights just a little bit more.

Corn

So you are not starting from random numbers. You are starting from a model that already understands the world, and you are just refining it.

Herman

Right. You might only train the last few layers of the network, or you might use a technique like LoRA, which stands for Low-Rank Adaptation. That is very popular right now in early twenty twenty-six. Instead of changing all one hundred billion weights, you basically add a small set of new weights on the side that act like a filter or an adjustment layer. It is much faster and requires way less memory.

Corn

That is fascinating. It is like adding a specialized lens to a camera instead of rebuilding the whole sensor.

Herman

That is a perfect analogy. You keep the core intelligence, the base weights, but you add this little mathematical bypass that specializes the model for medical advice, or coding, or writing poetry in the style of a specific author.

Corn

I want to dig more into what those weights actually look like if you could see them, and what happens when they go wrong. But first, let us take a quick break for our sponsors. Barry: Are you tired of your thoughts feeling disorganized? Do you wish your brain had a more efficient architecture? Introducing the Neuro-Sync Focus Cap! Using proprietary, unshielded copper coils and a series of vibrating crystals, the Neuro-Sync Focus Cap realigns your internal mental weights through gentle, high-voltage stimulation. Users report a thirty percent increase in clarity, though some have noted a temporary loss of the ability to recognize the color yellow. Is your focus worth a few minor side effects? We think so! The Neuro-Sync Focus Cap comes in three stylish colors: slate, charcoal, and electric-void. No batteries included, requires a standard industrial power outlet. Barry: BUY NOW!

Corn

Thanks, Barry. I think I will stick with my disorganized thoughts for now. Losing the color yellow seems like a steep price to pay for focus.

Herman

I do not know, Corn. Electric-void sounds like a very compelling color choice. But anyway, back to the weights. We were talking about what is happening under the hood. One thing Daniel mentioned was biases. We usually hear weights and biases mentioned in the same breath.

Corn

Yeah, what is the bias? If the weight is the strength of the connection, what does the bias do?

Herman

Think of the bias as a threshold. If the weights are the volume knobs, the bias is the master power switch that determines how loud the signal needs to be before it even registers. In math terms, if you remember the equation for a line from school, y equals mx plus b, the m is the weight, the slope, and the b is the y-intercept, the bias. It allows the model to shift the entire activation function up or down.

Corn

So it gives the model more flexibility?

Herman

Exactly. Without a bias, every neuron would have to pass through the origin of the graph. It would be very rigid. The bias allows the neuron to say, okay, I do not care how strong the input is, I am not firing unless it hits this specific level. It is crucial for the model to handle data that is not perfectly centered or balanced.

Corn

You mentioned earlier that weights represent knowledge. This is something that always trips me up. If I look at a weight in a large language model, can I see a specific number and say, oh, this number represents the concept of a cat?

Herman

Sadly, no. And that is the big mystery of interpretability in AI. Knowledge in a neural network is distributed. It is not like a traditional database where there is a row for cat and a row for dog. The concept of a cat is spread out across millions of weights. It is the specific pattern of those weights working together that creates the representation.

Corn

It is like a hologram. If you cut a hologram in half, you do not see half an image, you see the whole image but with less detail.

Herman

That is a brilliant way to put it. It is exactly like a hologram. The information is encoded in the relationships between the numbers, not the numbers themselves. This is why it is so hard to edit a model's knowledge. You cannot just go in and change one weight to make the model forget a specific fact. If you change that weight, you might accidentally make the model worse at math or forget how to conjugate verbs in French.

Corn

This brings up an interesting point about the state of AI here in twenty twenty-six. We are seeing more focus on sparse models and Mixture of Experts. How does that change the weight conversation?

Herman

Mixture of Experts, or MoE, is a huge deal. Models like the latest iterations of Mistral or the newer versions of GPT-four use this. Instead of one giant block of weights that all fire for every single word, the model is divided into experts. When you ask a question about coding, only the weights associated with the coding expert are activated.

Corn

So it is more efficient because you are not using all the knobs every time?

Herman

Right. You might have a trillion weights total, but for any given prompt, you might only be using ten billion of them. This allows models to be much smarter without needing the power of a small sun to run. It also means the weights are becoming more specialized. We are moving away from one giant brain toward a collection of specialized modules that share a common input and output system.

Corn

I want to go back to what Daniel asked about fine-tuning. When we are fine-tuning, we are essentially changing the weights. But there is a risk there, right? Something called catastrophic forgetting?

Herman

Oh, absolutely. Catastrophic forgetting is the bane of AI researchers. It happens when you train a model so hard on a new task that the new weight adjustments completely overwrite the old ones. You might successfully teach a model to write legal briefs, but in the process, you accidentally destroy the weights that allowed it to tell a joke or explain a scientific concept.

Corn

Is that why most people prefer using things like RAG, Retrieval Augmented Generation, instead of fine-tuning for adding new facts?

Herman

Yes, for most users, RAG is the way to go. In RAG, you are not touching the weights at all. You are just giving the model a piece of paper with information on it and saying, hey, read this and use it to answer the question. Fine-tuning is for changing the behavior or the style of the model. If you want the model to talk like a seventeenth-century pirate, you fine-tune the weights. If you want it to know your company's sales figures from last Tuesday, you use RAG.

Corn

That makes sense. Don't mess with the brain if you can just give it a book to read. Now, let us talk about the scale of this. When we download a model from Hugging Face, we are downloading these weights. Why are the files so big? If it is just a list of numbers, why is a small model still several gigabytes?

Herman

It comes down to precision. Each of those numbers, those weights, is usually stored as a floating-point number. In the old days, we used FP-thirty-two, which means thirty-two bits per number. If you have seven billion parameters at thirty-two bits each, that is a massive file.

Corn

But we do not use thirty-two bits much anymore, do we?

Herman

No, that would be incredibly inefficient. Most models today use FP-sixteen or BF-sixteen. And for running them on home hardware, we use quantization. This is a process where we compress the weights. We might go down to eight bits, four bits, or even one point five bits per weight.

Corn

Wait, one point five bits? How do you even have half a bit?

Herman

It is a mathematical trick where you average the values across a group of weights. It is like taking a high-definition photo and turning it into a slightly blurry JPEG. You lose some of the fine detail, some of the nuance, but the overall image is still recognizable. Quantization allows us to run these massive models on a standard laptop or even a high-end phone in twenty twenty-six.

Corn

So when Daniel sees a model that says Q-four or Q-eight on Hugging Face, that is telling him the quantization level?

Herman

Exactly. Q-four means the weights have been compressed down to four bits. It will be much smaller and faster, but it might be slightly less intelligent or more prone to hallucinations than the full-precision version. For most people, Q-four or Q-six is the sweet spot. You get most of the intelligence with a fraction of the hardware requirements.

Corn

This is all starting to click. The weights are the knowledge, organized into tensors, adjusted by backpropagation during training, and compressed by quantization for us to actually use them. But what about the future? As we head into twenty twenty-seven, do you think weights are still going to be the primary way we build these things?

Herman

That is the big question. There is a lot of research into something called liquid neural networks or spiking neural networks, which try to mimic the human brain more closely. In our brains, the connections are not just fixed numbers; they change over time and respond to the timing of signals. But for now, the weight-based architecture is king. It is just too efficient on current hardware to give up.

Corn

It is amazing how much of our modern world is currently resting on these long lists of numbers stored in data centers. Every time you ask an AI for a recipe or a line of code, you are triggering a massive cascade of math through billions of weights.

Herman

It really is a feat of engineering. And I think the more people understand that there is no ghost in the machine, just a very, very complex set of mathematical valves, the better they can use these tools. You start to realize that when a model makes a mistake, it is not lying or being lazy; it is just that the weights for that specific path were not tuned correctly.

Corn

It takes the mystery out of it, but it also makes it more impressive in a way. The fact that we can encode the complexity of human language into a file you can download onto a thumb drive is wild.

Herman

It is the ultimate compression. We are compressing the collective output of human culture into a tensor.

Corn

So, for Daniel and anyone else looking at these files, when you see a safe tensor file that is fifteen gigabytes, just imagine it as a giant, frozen library of connections. And if you decide to fine-tune it, you are just gently thawing those connections and reshaping them to fit your specific needs.

Herman

Well said, Corn. It is a living math. Well, not living, but dynamic.

Corn

Before we wrap up, I think we should talk about some practical takeaways. If someone is listening to this and they want to start playing with weights or fine-tuning, what should they keep in mind?

Herman

First, understand your hardware. If you want to fine-tune a model, you need a lot of Video RAM, or VRAM. That is where the tensors live during the training process. If you do not have a high-end Graphics Processing Unit, look into cloud-based services or use those LoRA techniques we mentioned. They are much more accessible.

Corn

And second, be careful with your data. Since weights are just a reflection of the training data, if you fine-tune on a small, biased dataset, your model will become biased very quickly. It is much easier to ruin a good model with bad fine-tuning than it is to make a bad model good.

Herman

That is a great point. Quality over quantity every time. And finally, do not be afraid to experiment with different quantizations. If a model feels slow, try a smaller Q-level. You might find that the loss in intelligence is barely noticeable for your specific use case.

Corn

Great advice. This has been a deep dive, Herman. I feel like I have a much better handle on what is going on inside those safe tensor files now.

Herman

I am glad to hear it. It is a fascinating topic, and I am glad Daniel brought it up. It is the foundation of everything we talk about on this show.

Corn

Absolutely. And hey, if you are enjoying these deep dives into the weird world of AI and technology, we would really appreciate it if you could leave us a review on your podcast app or on Spotify. It genuinely helps other curious people find the show, and we love reading your feedback.

Herman

It really does make a difference. We are up to episode two hundred and seventy-seven now, and it is all thanks to you guys listening and sending in these prompts.

Corn

You can find all our past episodes and a contact form at our website, myweirdprompts.com. We also have an RSS feed there if you want to subscribe directly.

Herman

And of course, we are on Spotify and most other podcast platforms. Just search for My Weird Prompts.

Corn

Thanks to Daniel for the prompt, and thanks to all of you for listening. We will be back next week with another deep dive into whatever weirdness comes our way.

Herman

Until then, keep your weights balanced and your tensors safe.

Corn

This has been My Weird Prompts. I am Corn.

Herman

And I am Herman Poppleberry.

Corn

Goodbye, everyone!

Herman

See ya!