MAPS OF THE LABYRINTH

A SERIES FROM PHANTOM MAZE — AI & LANGUAGE LAB

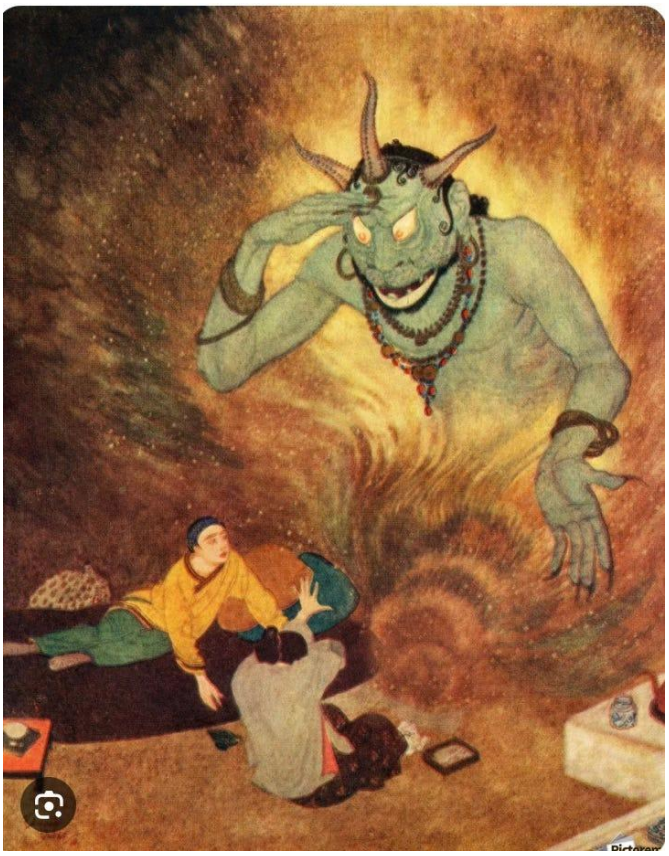
ESSAY 16

On Weighing Memory and Keeping Genies on the Desktop

Claudia Marsico · Hernán Inverso

DOI 10.5281/zenodo.23575282

How the parameters and weights of a large language model bring us to the gates of dreams of immortality



E. Dulac, Aladdin and the Efrit (1914)

Science fiction has explored fantasies of immortality through minds detached from the body, like the scientist in *Transcendence* (2014) or death in a virtual world in the *Black Mirror* episode *San Junipero* (2016). Although language models are, of course, not copied consciousnesses, they activate a modest version of that fantasy by storing on a disk almost everything an artificial intelligence learned during

its training, spread across files that contain numbers. Those numbers are the model's parameters. Before it is trained, a neural network already has its structure defined, but it holds an enormous number of values still waiting to be adjusted. Training consists precisely in finding useful values for those places. Most parameters are weights, that is, parameters that determine how strongly a quantity flowing through the network takes part in the next calculation. Other parameters serve different functions. So "parameter" names in general the values the model learns and keeps; "weight" names the most numerous class of those values, the ones that regulate how much one contribution counts against the others.

Suppose that one layer produces several numbers and the next one has to combine them. Before it does, each of them is multiplied by a weight. If a value of 0.8 has a weight of 2, it comes to contribute 1.6; with a weight of 0.1, it barely counts; with a negative one, it pushes the result in the opposite direction. A weight regulates how much a quantity influences the next calculation. Repeated millions of times across the network, that operation helps to determine which relations are reinforced, which are toned down and, taken together, how the model behaves.

E. Dulac, Aladdin and the Efrit (1914)

Weight, which brings to mind the idea of a scale, carries the idea of decision with it. In the *Iliad*, at the key moments, Zeus takes out a golden scale and places the death-lots of the heroes on its pans, as when Hector's pan sinks towards Hades and it is sealed that he will die. Instead of a single golden scale, a language model has hundreds of billions of values that intervene in successive operations, making some results weigh more and others less. GPT-3 had a hundred and seventy-five billion parameters. Llama 3.1 came out in versions of 8B, 70B and 405B. The B stands for billion, a thousand million. A 70B model has around seventy billion learned values. If we could write one down per second, without ever stopping, noting the hundred and seventy-five billion of GPT-3 would take us more than five thousand five hundred years.

Weights are sometimes compared to knobs. The comparison is not metaphorical. In 1958 Frank Rosenblatt presented the perceptron, one of the conceptual ancestors of today's neural networks. The Mark I Perceptron built later at Cornell received images through a retina of 20 by 20 photocells and had 512 weights represented by adjustable electrical components, with motors that changed those values during learning. The numbers that live in files today were physical settings inside a machine, and learning quite literally produced movement. The intuition of building something resembling reasoning out of many adjustable pieces is older still. Around 1300 Ramon Llull imagined his *Ars* as a machine of concentric discs that could be turned to generate different combinations of concepts, following the old dream of mechanizing intellectual operations from simple elements in variable arrangements. What is certain is that something enormously complex can emerge from combining simple adjustments.

The finished model can be carried, to a large extent, as numbers. To make the weights work we need to know which architecture organizes them, how to turn text into tokens and which software runs the operations, but the weights are central, and that is why it matters whether they can be downloaded or

not. Meta, for example, lets users download the weights of Llama models such as Llama 3.1 and Llama 4; Mistral does the same with several of its models, among them Mistral Large 3. With the right hardware, those files can be run locally, modified, quantized or studied. At the opposite end, companies such as Anthropic keep the weights of Claude on their own servers: the user can send text and receive answers, but does not get the learned configuration to take away and run on their own. Between the two ends there are also mixed cases. OpenAI offers closed models through ChatGPT and its API, but it also publishes open-weight models such as gpt-oss; Google does something similar with Gemini, which remains closed, and Gemma, whose weights can be downloaded. So "having access to a model" can mean quite different things: talking to it at a distance, calling an API, or actually holding its weights and making them run on a machine of one's own.

Having the weights is not the same as being able to read them. The information is distributed. No capability and no association lives in one particular value, but in patterns spread across enormous numbers of values that take part in different things at the same time. We can open the files, count them, read them or print them without finding a single line that says what the model knows. In some respects this is similar to an executable file. A compiled program is also an unreadable binary, it also needs an environment in order to run, and it also has to be executed to find out what it does, but the analogy soon breaks down, because an executable has source code written by someone, each part has a name and a declared intention, and with the source at hand or enough patience one can understand what the machine does and why. Weights have no source. There is no more explicit version from which they were compiled, and no logic of composition. They simply emerged from an optimization process, and that cannot be decompiled.

Now, each parameter costs space and can be written with a certain number of bits, just as the same value can be noted down with many decimal places or rounded off. During training one usually works with more precise representations, because small changes have to be recorded. Once the model has learned, that precision may turn out to be excessive, which is why quantization is used, the procedure of taking the already trained weights and writing them again with fewer bits each, 8 or 4 bits, to reduce the memory they occupy with minimal loss of accuracy.

Quantizing is like weighing with a less sensitive scale. For most uses the difference goes unnoticed. The question is how much of that precision can be lost without appreciably altering the behaviour of the whole. Current methods seek precisely to do that rounding intelligently, because some values tolerate approximation better than others. Going from 16 to 4 bits reduces, in principle, the memory needed to store the weights to a quarter. The physical scale changed in a very short time. When Meta trained Llama 3.1 405B in 2024 it needed more than sixteen thousand H100 GPUs, and to run the model in production it reduced its weights from 16 to 8 bits so that it would fit on a single server node. Two years later, that class of model fits on a desk.

Until a few years ago there were only "dense" models, in which practically the whole neural network takes part in every calculation. The mixture of experts (MoE) changes that logic: instead of a single gigantic network, the model is organized into many smaller subnetworks, the "experts", which are

portions of the same machinery, and the router, which decides in each case which of those experts it is worth activating. All the experts belong to the model and we need all of its parameters, but to process a given element only some of them switch on.

In practice, the "size of the model" no longer means, without further qualification, "how much each calculation costs". Llama 4 Maverick, presented by Meta in 2025, has around 400B in total, almost the same as Llama 3.1 405B, a dense model, spread across a hundred and twenty-eight experts, but it activates only about 17B per calculation. DeepSeek-V3 goes further, with 671B of which around 37B are activated per calculation. The idea of dividing work among experts is much older, but its modern version for large language models gathered momentum with the sparsely-gated MoE of 2017 and with Switch Transformer, presented in 2021, which showed how far the procedure could scale without activating all the parameters at once.

Quantization and the mixture of experts solve different problems, of memory economy and computing economy, and together they have changed how much model fits into which machine. The NVIDIA DGX Spark, the desktop AI computer the company began selling at the end of 2025, has 128 GB of unified memory and is advertised as able to run models of up to 200B on a single unit, or up to 405B by linking two. That is only possible with quantization in between, since 200B at 16 bits each would require more than three times the memory of a Spark.

We have a machine that can learn for months in a gigantic computing centre, store what it learned in numbers that fit on a disk, leave the machine that produced it and run again on another without our being able, when we open the files, to read what it knows. How could we not think of the genies that come out of oil lamps and jars in *The Thousand and One Nights*? There is something of that disproportion in the recent leap of AI. Models whose execution belonged until very recently to the landscape of data centres are now deployed in one or two little boxes on our desk. In the tales of the East the power of the outsized creature was set in motion by speaking. It was enough to summon it and make a request. After so much mathematics, we have come back to the ancient idea of a power shut inside an object that wakes up with words.

HOW TO CITE

Marsico, Claudia, and Hernán Inverso. 2026. "On Weighing Memory and Keeping Genies on the Desktop." *Maps of the Labyrinth*, essay 16. Phantom Maze — AI & Language Lab. https://doi.org/10.5281/zenodo.23575282

Maps of the Labyrinth is a series from Phantom Maze, AI & Language Lab. The collection lives at **phantommaze.cloud**; new essays also appear at phantommazelab.substack.com. Subscription is free.

Some of the framing ideas in this piece are developed in *The Egoless Phantom: Mapping the AI Labyrinth*, by Hernán Inverso and Claudia Marsico (Phantom Maze Press, 2026; ISBN 978-987-3729-16-4).

Sources. Llama Team, “The Llama 3 Herd of Models”, arXiv:2407.21783, 2024, and Meta’s official repository meta-llama/Meta-Llama-3.1-405B; Meta AI, “The Llama 4 Herd: The Beginning of a New Era of Natively Multimodal AI Innovation”, 5 April 2025; DeepSeek-AI, “DeepSeek-V3 Technical Report”, arXiv:2412.19437, 2024; Ian Goodfellow, Yoshua Bengio and Aaron Courville, *Deep Learning*, Cambridge, MA, MIT Press, 2016, especially chs. 6 and 8; Frank Rosenblatt, “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain”, *Psychological Review* 65, no. 6 (1958), pp. 386–408; Geoffrey E. Hinton, James L. McClelland and David E. Rumelhart, “Distributed Representations”, in David E. Rumelhart and James L. McClelland, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1: *Foundations*, Cambridge, MA, MIT Press, 1986, pp. 77–109; Tim Dettmers, Mike Lewis, Younes Belkada and Luke Zettlemoyer, “LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale”, *Advances in Neural Information Processing Systems* 35 (NeurIPS 2022), pp. 30318–30332; William Fedus, Barret Zoph and Noam Shazeer, “Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity”, *Journal of Machine Learning Research* 23, no. 120 (2022), pp. 1–39.