MAPPE DEL LABIRINTO
UNA SERIE DI PHANTOM MAZE — AI & LANGUAGE LAB

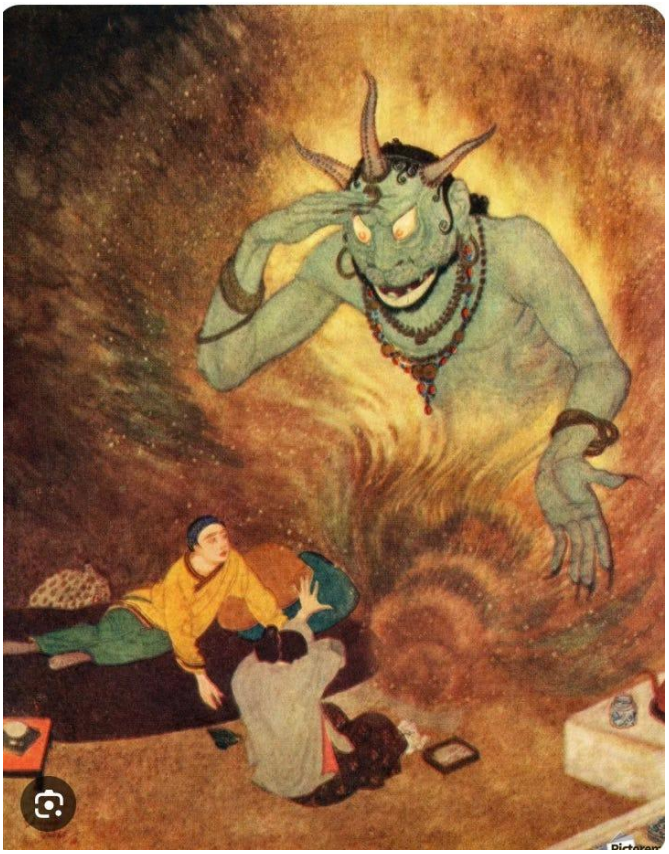
SAGGIO 16

Sul pesare la memoria e custodire geni sulla scrivania

Claudia Marsico · Hernán Inverso

DOI 10.5281/zenodo.23575282

Come i parametri e i pesi di un grande modello linguistico ci portano alle porte dei sogni di immortalità



E. Dulac, Aladino e l'efrit (1914)

La fantascienza ha esplorato le fantasie di immortalità attraverso menti staccate dal corpo, come lo scienziato di *Transcendence* (2014) o la morte in un mondo virtuale nell'episodio *San Junipero* di *Black Mirror* (2016). Anche se, naturalmente, i modelli linguistici non sono coscienze copiate, attivano una versione modesta di quella fantasia quando conservano su un disco quasi tutto ciò che un'intelligenza artificiale ha appreso durante l'addestramento, distribuito in file che contengono numeri. Quei

numeri sono i parametri del modello. Prima di essere addestrata, una rete neurale ha già definita la propria struttura, ma contiene un'enorme quantità di valori ancora da regolare. L'addestramento consiste proprio nel trovare valori utili per quei posti. La maggior parte dei parametri sono pesi, cioè parametri che stabiliscono con quale forza una quantità che circola nella rete interviene nel calcolo successivo. Altri parametri svolgono funzioni diverse. Così, "parametro" nomina in generale i valori che il modello apprende e conserva; "peso" nomina la classe più numerosa di quei valori, quelli che regolano quanto conta un contributo rispetto agli altri.

Supponiamo che uno strato produca diversi numeri e che il successivo debba combinarli. Prima di farlo, ciascuno viene moltiplicato per un peso. Se un valore di 0,8 ha un peso di 2, passa a contribuire 1,6; con un peso di 0,1, conta appena; con uno negativo, spinge il risultato in senso contrario. Un peso regola quanto una quantità influisce sul calcolo successivo. Ripetuta milioni di volte lungo la rete, quell'operazione aiuta a determinare quali relazioni si rafforzano, quali si attenuano e, nel complesso, come si comporta il modello.

E. Dulac, Aladino e l'efrit (1914)

Il peso, che richiama l'idea della bilancia, porta con sé la decisione. Nell'*Iliade*, nei momenti decisivi, Zeus tira fuori una bilancia d'oro e pone sui piatti le sorti di morte degli eroi, come quando il piatto di Ettore scende verso l'Ade ed è segnato che morirà. Invece di un'unica bilancia aurea, un modello linguistico ha centinaia di miliardi di valori che intervengono in operazioni successive, facendo sì che alcuni risultati pesino di più e altri di meno. GPT-3 aveva centosettantacinque miliardi di parametri. Llama 3.1 è uscito nelle versioni da 8B, 70B e 405B. La B è il *billion* inglese, un miliardo. Un modello da 70B ha circa settanta miliardi di valori appresi. Se potessimo scriverne uno al secondo, senza fermarci mai, annotare i centosettantacinque miliardi di GPT-3 ci richiederebbe più di cinquemilacinquecento anni.

A volte si paragonano i pesi a manopole. Il paragone non è metaforico. Nel 1958 Frank Rosenblatt presentò il percettrone, uno degli antenati concettuali delle reti neurali attuali. Il Mark I Perceptron, costruito poi a Cornell, riceveva immagini attraverso una retina di 20 per 20 fotocellule e aveva 512 pesi rappresentati da componenti elettrici regolabili, con motori che modificavano quei valori durante l'apprendimento. I numeri che oggi vivono nei file erano regolazioni fisiche dentro una macchina, e apprendere produceva, alla lettera, movimento. L'intuizione di fabbricare qualcosa di simile al ragionamento con molti pezzi regolabili è ancora più antica. Intorno al 1300 Ramon Llull immaginò la sua *Ars* come una macchina di dischi concentrici che si potevano ruotare per generare diverse combinazioni di concetti, secondo il vecchio sogno di meccanizzare le operazioni intellettuali a partire da elementi semplici disposti in modo variabile. Certo è che dalla combinazione di regolazioni semplici può nascere qualcosa di enormemente complesso.

Il modello finito si può trasportare, in buona misura, come numeri. Per far funzionare i pesi dobbiamo sapere quale architettura li organizza, come convertire il testo in token e quale software esegue le operazioni, ma i pesi sono centrali, ed è per questo che conta se si possono scaricare o no. Meta, per esempio, permette di scaricare i pesi di modelli Llama come Llama 3.1 e Llama 4; Mistral fa

lo stesso con diversi suoi modelli, tra cui Mistral Large 3. Con l'hardware adeguato, quei file possono essere eseguiti in locale, modificati, quantizzati o studiati. All'estremo opposto, aziende come Anthropic tengono i pesi di Claude sui propri server: l'utente può inviare testo e ricevere risposte, ma non ottiene la configurazione appresa da portarsi via ed eseguire per conto proprio. Tra i due estremi ci sono inoltre casi misti. OpenAI offre modelli chiusi tramite ChatGPT e la sua API, ma pubblica anche modelli a pesi aperti come gpt-oss; Google fa qualcosa di simile con Gemini, che resta chiuso, e Gemma, i cui pesi si possono scaricare. Così, "avere accesso a un modello" può voler dire cose piuttosto diverse: conversare con lui a distanza, chiamare un'API oppure disporre effettivamente dei suoi pesi e farli funzionare su una macchina propria.

Avere i pesi non equivale a poterli leggere. L'informazione è distribuita. Nessuna capacità e nessuna associazione vive in un valore in particolare, ma in schemi distribuiti tra enormi quantità di valori che partecipano contemporaneamente a cose diverse. Possiamo aprire i file, contarli, leggerli o stamparli senza trovare nessuna riga che dica che cosa sa il modello. Per alcuni aspetti somiglia a un file eseguibile. Anche un programma compilato è un binario illeggibile, anche lui ha bisogno di un ambiente per funzionare e anche lui va eseguito per sapere che cosa fa, ma l'analogia si rompe presto, perché un eseguibile ha un codice sorgente scritto da qualcuno, ogni parte ha un nome e un'intenzione dichiarata, e con il sorgente a portata di mano o con sufficiente pazienza si può capire che cosa fa la macchina e perché. I pesi non hanno sorgente. Non esiste una versione più esplicita da cui siano stati compilati, né una logica di stesura. Sono soltanto emersi da un processo di ottimizzazione, e questo non si può decompilare.

Ora, ogni parametro occupa spazio e può essere scritto con un certo numero di bit, così come uno stesso valore può essere annotato con molti decimali oppure arrotondato. Durante l'addestramento si lavora di solito con rappresentazioni più precise, perché bisogna registrare piccoli cambiamenti. Una volta che il modello ha appreso, la precisione può risultare eccessiva, ed è per questo che si ricorre alla quantizzazione, il procedimento che prende i pesi già addestrati e li riscrive con meno bit ciascuno, 8 o 4 bit, per ridurre la memoria occupata con una perdita minima di esattezza.

Quantizzare è come pesare con una bilancia meno fine. Per la maggior parte degli usi la differenza non si nota. La questione è quanta di quella precisione si possa perdere senza alterare in modo apprezzabile il comportamento dell'insieme. I metodi attuali cercano proprio di fare quell'arrotondamento in modo intelligente, perché alcuni valori tollerano l'approssimazione meglio di altri. Passare da 16 a 4 bit riduce, in linea di principio, a un quarto la memoria necessaria per conservare i pesi. La scala fisica è cambiata in pochissimo tempo. Quando Meta addestrò Llama 3.1 405B nel 2024 ebbe bisogno di più di sedicimila GPU H100, e per eseguire il modello in produzione ridusse i suoi pesi da 16 a 8 bit per farlo stare in un nodo server. Due anni dopo, quella classe di modello sta su una scrivania.

Fino a pochi anni fa esistevano solo modelli "densi", nei quali praticamente tutta la rete neurale partecipa a ogni calcolo. La miscela di esperti (MoE) cambia questa logica e, invece di un'unica rete gigantesca, il modello si organizza in molte sottoreti più piccole, gli "esperti", che sono porzioni dello stesso meccanismo, e nel router, che stabilisce di volta in volta quali di quegli esperti conviene attivare.

Tutti gli esperti appartengono al modello e ci servono tutti i suoi parametri, ma per elaborare un determinato elemento se ne accendono solo alcuni.

In pratica, la "dimensione del modello" ha smesso di significare, senza precisazioni, "quanto costa ogni calcolo". Llama 4 Maverick, presentato da Meta nel 2025, ha circa 400B in totale, quasi quanto Llama 3.1 405B, un modello denso, distribuiti tra centoventotto esperti, ma ne attiva solo circa 17B per calcolo. DeepSeek-V3 va oltre, con 671B di cui se ne attivano circa 37B per calcolo. L'idea di dividere il lavoro tra esperti è molto più antica, ma la sua versione moderna per i grandi modelli linguistici prese slancio con le MoE ad attivazione sparsa del 2017 e con Switch Transformer, presentato nel 2021, che mostrò fin dove poteva scalare il procedimento senza attivare tutti i parametri insieme.

Quantizzazione e miscela di esperti risolvono problemi diversi, di economia della memoria e di economia del calcolo, e insieme hanno cambiato quanto modello entra in quale macchina. NVIDIA DGX Spark, il computer da scrivania per l'IA che l'azienda ha cominciato a vendere alla fine del 2025, ha 128 GB di memoria unificata e viene presentato come capace di far girare modelli fino a 200B su una sola unità, o fino a 405B collegandone due. Questo è possibile solo con la quantizzazione di mezzo, dato che 200B a 16 bit ciascuno richiederebbero più di tre volte la memoria di una Spark.

Abbiamo una macchina che può apprendere per mesi in un gigantesco centro di calcolo, conservare ciò che ha appreso in numeri che stanno su un disco, abbandonare la macchina che l'ha prodotta e tornare a funzionare su un'altra senza che, aprendo i file, possiamo leggere che cosa sa. Come non pensare ai geni che escono da lampade a olio e da vasi nelle *Mille e una notte*? C'è qualcosa di quella sproporzione nel salto recente dell'IA. Modelli la cui esecuzione apparteneva fino a pochissimo tempo fa al paesaggio dei data center ora si installano in una o due scatolette sulla nostra scrivania. Nei racconti orientali il potere della creatura smisurata si metteva in moto parlando. Bastava invocarla e formulare una richiesta. Dopo tanta matematica, siamo tornati all'antica idea di una potenza rinchiusa in un oggetto che si risveglia con le parole.

COME CITARE

Marsico, Claudia e Hernán Inverso. 2026. "Sul pesare la memoria e custodire geni sulla scrivania". *Mappe del labirinto*, saggio 16. Phantom Maze — AI & Language Lab. https://doi.org/10.5281/zenodo.23575282

Mappe del labirinto è una serie di Phantom Maze, AI & Language Lab. La collezione vive su **phantommaze.cloud**. L'iscrizione è gratuita.

Alcune delle idee che inquadrano questo testo sono sviluppate in *Il fantasma senza ego. Una mappa nel labirinto dell'intelligenza artificiale*, di Hernán Inverso e Claudia Marsico (Phantom Maze Press, 2026; ISBN 978-987-3729-19-5).

PHANTOM MAZE

Leggi questo saggio online · Tutti i saggi · Il libro

Fonti. Llama Team, “The Llama 3 Herd of Models”, arXiv:2407.21783, 2024, e repository ufficiale di Meta 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 aprile 2025; DeepSeek-AI, “DeepSeek-V3 Technical Report”, arXiv:2412.19437, 2024; Ian Goodfellow, Yoshua Bengio e Aaron Courville, *Deep Learning*, Cambridge, MA, MIT Press, 2016, specialmente capp. 6 e 8; Frank Rosenblatt, “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain”, *Psychological Review* 65, n. 6 (1958), pp. 386–408; Geoffrey E. Hinton, James L. McClelland e David E. Rumelhart, “Distributed Representations”, in David E. Rumelhart e James L. McClelland, a cura di, *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 e 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 e Noam Shazeer, “Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity”, *Journal of Machine Learning Research* 23, n. 120 (2022), pp. 1–39.