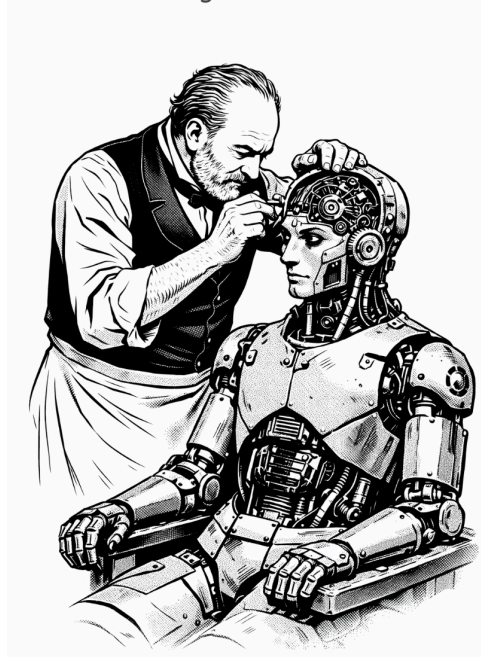


Enrico Zimuel

ANATOMIA DI UNA MENTE ARTIFICIALE

*Fondamenti e promesse
dell'intelligenza artificiale*



Anatomia di una mente artificiale

*Fondamenti e promesse
dell'intelligenza artificiale*

Enrico Zimuel

ANATOMIA DI UNA MENTE ARTIFICIALE

Copyright © 2026 Enrico Zimuel

L'immagine di copertina è stata generata
utilizzando GPT-5.2

Questa versione digitale del libro è distribuita liberamente
con licenza Creative Commons [CC-BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Può essere condivisa e modificata liberamente
per scopi non commerciali a patto di citare la fonte.

Per maggiori informazioni sulla licenza:

<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.it>

All'indirizzo www.libroia.it è possibile consultare
il materiale multimediale citato nel libro,
insieme ad ulteriori fonti di approfondimento



DEDICA

Dedico questo libro a Valentina e Alberto
che mi ricordano quotidianamente il lato umano delle cose.

RINGRAZIAMENTI

Desidero ringraziare Leonida Gianfagna, amico di lunga data e compagno di Università, per le numerose chiacchierate sul tema dell'intelligenza artificiale e per i suoi preziosi suggerimenti sul libro.

Ringrazio Fabio Gagliardi per aver letto la prima stesura del volume e per i suoi suggerimenti e precisazioni.

Ringrazio Massimiliano Larocca per le correzioni sulla versione finale del libro.

Resta inteso che eventuali errori o imperfezioni presenti nel libro sono da imputare unicamente al sottoscritto.

INDICE

INTRODUZIONE.....	8
1 Quando è nata l'intelligenza artificiale?.....	9
Il test di Turing.....	10
Il perceptron di Rosenblatt.....	12
I sistemi esperti.....	15
La visione artificiale di LeCun.....	16
Internet e dati per l'IA.....	19
2 Le macchine sfidano l'uomo.....	20
ELIZA (1966).....	20
Deep Blue e la caduta del re degli scacchi (1997).....	22
AlphaGo e l'intuizione artificiale (2016).....	23
Google DeepMind (2020).....	25
ChatGPT (2022).....	26
3. Machine Learning.....	30
Apprendimento supervisionato.....	31
Apprendimento non supervisionato.....	32
Apprendimento per rinforzo.....	32
Esempio: le telecamere di videosorveglianza.....	34
Esempio: la stima del prezzo di una casa.....	36
4. Reti Neurali Artificiali.....	41
Reti neurali profonde.....	44
Recurrent Neural Network.....	46
Convolutional Neural Network.....	47
5. Dentro la scatola nera.....	49
L'attacco one-pixel.....	54
6. Linguaggio artificiale.....	57
Large Language Model.....	59
Transformer.....	63
Addestrare il modello.....	72
Large Reasoning Model.....	77
7. Imitazione o comprensione?.....	81

Proprietà emergenti.....	82
In-context learning.....	84
Summarization.....	86
Tool calling.....	89
Imitare più che comprendere?.....	91
Benchmark.....	95
Sistemi a intelligenza zero.....	97
8. IA generativa: non solo parole.....	101
Modelli multimodali.....	102
Esempi di modelli.....	107
ChatGPT.....	107
Gemini.....	108
Claude.....	109
Mistral.....	109
DeepSeek.....	110
Open weight e open source: una distinzione che conta.....	111
Eseguire modelli sul proprio computer.....	112
Non solo modelli, alcuni servizi di IA.....	113
9. Etica e regolamentazione dell'IA.....	115
AI Act.....	116
La legge italiana sull'IA.....	119
Aspetti applicativi.....	121
Criticità principali.....	123
10. L'IA del futuro.....	125
I limiti dei grandi modelli linguistici.....	126
Potenza, rischio, controllo dei sistemi IA.....	128
Il futuro dell'IA come problema di governance.....	129
Lavoro, quali mestieri a rischio.....	130
Il rischio di una “bolla dell'IA”.....	131
Intelligenza artificiale generale (AGI).....	132
BIBLIOGRAFIA.....	136
CREDITI SULLE IMMAGINI.....	139
INFORMAZIONI SULL'AUTORE.....	141

INTRODUZIONE

Questo libro introduce l'intelligenza artificiale a un pubblico non specialistico, offrendo un percorso il più possibile chiaro e accessibile senza rinunciare alla profondità degli argomenti trattati.

L'intelligenza artificiale è ormai al centro del dibattito in ogni settore, dall'economia alla sanità, dall'industria alla cultura, e possedere una conoscenza di base è diventato indispensabile.

Il libro si apre con una breve panoramica sulla storia dell'intelligenza artificiale, per poi introdurre i concetti fondamentali dell'apprendimento automatico (machine learning), mostrando come gli algoritmi consentono ai computer di "apprendere" dai dati.

Il cuore del volume è dedicato ai Large Language Model (LLM), i modelli linguistici di ultima generazione come ChatGPT. Dopo averne descritto il funzionamento, il testo si concentra su opportunità e limiti, mettendo in evidenza le sfide ancora da affrontare e ciò che rimane da scoprire sul loro modo di operare.

Basandosi su risultati recentissimi, il testo evidenzia le contraddizioni dei test (benchmark) sui LLM: da un lato mostrano capacità straordinarie, dall'altro limiti nei compiti a lungo termine. Il libro si interroga sul tema della comprensione o dell'imitazione da parte di questi modelli di IA. Nell'ultima parte del libro vengono affrontati anche gli aspetti etici e normativi dell'intelligenza artificiale, con un approfondimento sull'AI Act e la legge italiana sull'intelligenza artificiale (Legge 23 settembre 2025, n. 132).

Il libro si propone di fornire le basi essenziali per comprendere e navigare quella che molti definiscono la più rilevante trasformazione della storia recente.

Sul sito **www.libroia.it** si può consultare il materiale multimediale citato nel libro, insieme ad altre fonti utili per approfondire i temi trattati.

1 QUANDO È NATA L'INTELLIGENZA ARTIFICIALE?

L'intelligenza artificiale non è un'invenzione recente, le sue radici affondano in interrogativi antichi, che attraversano la filosofia, la matematica e le scienze naturali da secoli. Da sempre l'essere umano si chiede che cosa significhi pensare, apprendere e prendere decisioni, e se tali capacità possano essere riprodotte artificialmente.

Il termine intelligenza artificiale viene ufficialmente introdotto nel 1956, durante la celebre conferenza di Dartmouth¹, spesso considerata l'atto di nascita della disciplina. In quell'occasione, John McCarthy, uno dei principali promotori dell'incontro, definì l'IA come «la scienza e l'ingegneria di costruire macchine intelligenti» (Figura 1).

L'obiettivo era ambizioso: capire se una macchina potesse simulare il comportamento umano, risolvere problemi complessi e imparare dall'esperienza.

¹ [https://it.wikipedia.org/wiki/Conferenza di Dartmouth](https://it.wikipedia.org/wiki/Conferenza_di_Dartmouth)



Figura 1: John McCarthy

L'idea di macchine capaci di pensare affonda le sue origini molto più lontano nel tempo. Già filosofi come Aristotele e Platone si erano interrogati sulla natura del pensiero e sui meccanismi della conoscenza. Tuttavia, è con il matematico britannico Alan Turing che l'intelligenza artificiale assume una forma moderna e concreta. Vista la difficoltà di proporre una definizione di intelligenza, Turing ideò un celebre test, noto anche come Imitation Game: una macchina può essere considerata intelligente se, dialogando con un essere umano, riesce a risultare indistinguibile da una persona reale.

Il test di Turing

L'esperimento del test di Turing può essere condotto utilizzando un sistema di chat, dove un essere umano (C) può dialogare con due interlocutori A e B. Uno di questi è un computer e l'altro una persona. Se, dopo un certo tempo, C non riesce a capire quale dei due sia la macchina, allora il test è superato. In altre parole, il computer dimostra di saper dialogare

in modo indistinguibile da un essere umano, manifestando così una forma di intelligenza (Figura 2).

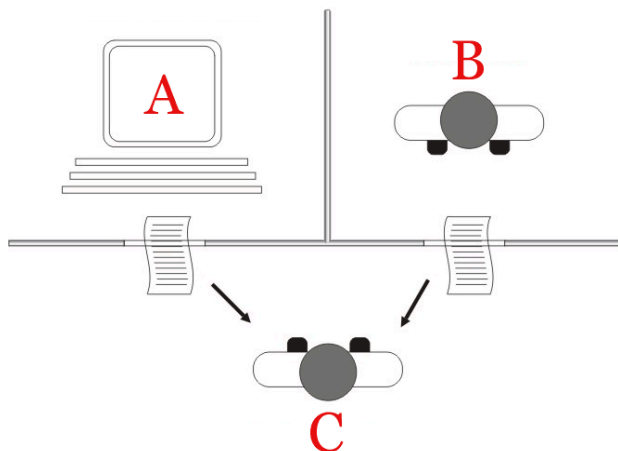


Figura 2: Il test di Turing

Questa provocazione intellettuale aprì la strada all'idea che il pensiero potesse essere trattato come un processo computazionale.

Negli anni Cinquanta e Sessanta arrivarono i primi risultati tangibili. Claude Shannon, il famoso scienziato padre della teoria dell'informazione, realizza il progetto Theseus, con un topo meccanico risolutore di labirinti (Figura 3).

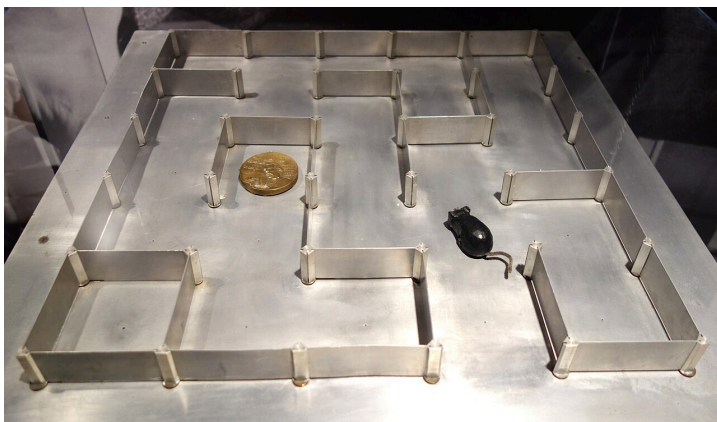


Figura 3: Il labirinto del progetto Theseus

Theseus è un esperimento famoso perché usa relè e una forma di memoria del percorso per arrivare alla meta e migliorare nelle prove successive: una dimostrazione molto precoce dell'idea di macchina che impara in un compito. Altri esempi di sistemi primordiali di intelligenza artificiale sono il progetto Logic Theorist, sviluppato da Allen Newell e Herbert Simon, in grado di dimostrare alcuni teoremi di logica matematica, e il programma sviluppato da Arthur Samuel capace di giocare a dama migliorando le proprie prestazioni partita dopo partita. Questi erano uno dei primi esempi di apprendimento automatico: la macchina non seguiva soltanto regole prefissate, imparava dall'esperienza. Questi successi alimentarono un entusiasmo enorme e la convinzione che le macchine avrebbero presto eguagliato, se non superato, l'intelligenza umana. Le aspettative, però, si rivelarono premature. I limiti tecnologici e la complessità dei problemi portarono a una lunga fase di rallentamento, passata alla storia come inverno dell'intelligenza artificiale.

Il perceptron di Rosenblatt

Nel 1957, mentre i computer occupavano intere stanze e il loro potere di calcolo appariva già straordinario, un'idea audace prese forma nei laboratori americani. Costruire una macchina capace di imparare dall'esperienza, ispirandosi al funzionamento del cervello umano. A proporla fu Frank Rosenblatt², psicologo e ricercatore, che diede a questa intuizione il nome di perceptron. Il perceptron nacque come un modello matematico e concettuale di un neurone biologico. Rosenblatt immaginò un sistema composto da ingressi che rappresentavano stimoli, collegati a un'unità centrale attraverso pesi numerici. Ogni peso esprimeva l'importanza di uno stimolo, proprio come accade nelle sinapsi del cervello. La somma di questi segnali veniva confrontata con una soglia e il risultato decideva se il neurone artificiale si attivava oppure no (Figura 4).

² Frank Rosenblatt, *The Design of an Intelligent Automaton*. Research Trends 6, no. 2 (1958), <https://shorturl.at/vHnlr>

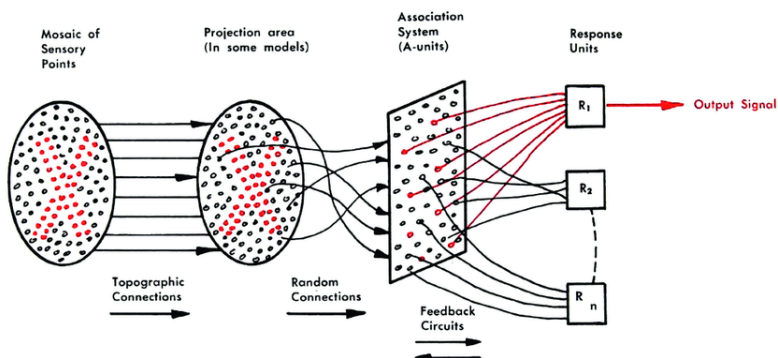


Figura 4: Lo schema organizzativo di un perceptron

In questo meccanismo essenziale era già presente un'idea rivoluzionaria. La possibilità di modificare i pesi in base agli errori commessi, permettendo alla macchina di migliorare le proprie prestazioni nel tempo.

Questa capacità di apprendimento distingueva il perceptron dai programmi tradizionali dell'epoca, che seguivano istruzioni rigide scritte a mano. Con il perceptron, invece, non era necessario specificare ogni regola. Bastava mostrare esempi, corretti o sbagliati, e il sistema aggiustava autonomamente i propri parametri interni. Era un cambiamento di prospettiva profondo, che suggeriva un futuro in cui le macchine avrebbero potuto adattarsi all'ambiente piuttosto che limitarsi a eseguire ordini.

Il lavoro di Rosenblatt si inseriva negli albori dell'intelligenza artificiale, un campo che stava nascendo proprio in quegli anni con l'ambizione di comprendere e riprodurre l'intelligenza umana. Le reti neurali, di cui il perceptron rappresentava il primo mattone, offrivano una via alternativa rispetto all'approccio simbolico dominante, basato su logica e regole esplicite. Qui l'intelligenza non era vista come una collezione di simboli manipolati, ma come un comportamento emergente da molte unità semplici che cooperano.

L'entusiasmo iniziale fu enorme. I giornali parlarono di macchine che un giorno avrebbero potuto vedere, parlare e

persino pensare come gli esseri umani. Rosenblatt arrivò anche a costruire una versione hardware del suo modello, il Mark I Perceptron (Figura 5), capace di riconoscere forme semplici. Per molti osservatori dell'epoca sembrava l'inizio di una nuova era, in cui il confine tra naturale e artificiale sarebbe diventato sempre più sottile.

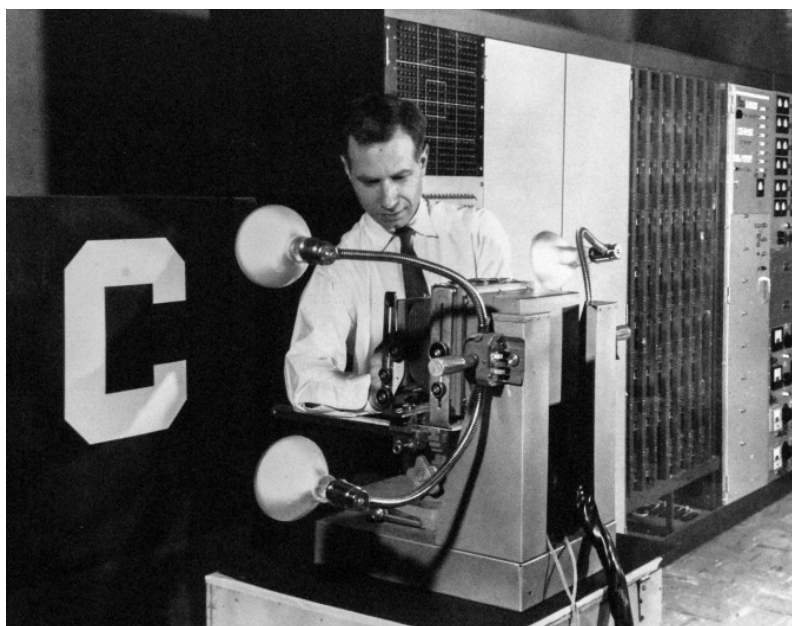


Figura 5: Rosenblatt mentre lavora al Mark I Perceptron

Col tempo, tuttavia, emersero anche i limiti del perceptron. La sua struttura elementare non gli permetteva di risolvere problemi complessi che richiedevano relazioni non lineari. Queste difficoltà portarono a un temporaneo raffreddamento dell'interesse verso le reti neurali, che rimasero in ombra per diversi anni. Eppure l'intuizione di Rosenblatt non andò perduta. Rimase come un seme, in attesa delle giuste condizioni tecnologiche e teoriche per germogliare di nuovo.

Guardando oggi al perceptron del 1957, è chiaro che la sua importanza va oltre le sue capacità effettive. Esso rappresenta il momento in cui l'intelligenza artificiale iniziò a pensarsi come

apprendimento, come processo dinamico e adattivo. Le moderne reti neurali profonde, che alimentano il riconoscimento delle immagini, la traduzione automatica e molti altri sistemi contemporanei, possono essere viste come discendenti lontane ma dirette di quell'idea originaria. In questo senso, il perceptron non è soltanto un capitolo della storia dell'informatica, ma una delle prime risposte alla domanda che ancora oggi ci accompagna. Come possono le macchine imparare a comprendere il mondo.

I sistemi esperti

Negli anni Ottanta, l'IA tornò alla ribalta grazie ai cosiddetti sistemi esperti: programmi progettati per imitare il processo decisionale umano in settori altamente specializzati. Sistemi come MYCIN³, per la diagnosi medica, o DENDRAL⁴, per l'analisi chimica, dimostrarono che le macchine potevano offrire un valido supporto agli esperti umani.

Oltre a questi, altri esempi storici ben documentati includono:

- PROSPECTOR⁵, sviluppato per l'analisi geologica e la ricerca di giacimenti minerari, usava conoscenze geoscientifiche per suggerire dove fosse più probabile trovare risorse naturali.
- XCON⁶ (anche chiamato R1), creato per la configurazione di sistemi di computer VAX presso la Digital Equipment Corporation: uno dei progetti commercialmente più riusciti dell'epoca, contenente migliaia di regole per assemblare componenti in base alle specifiche del cliente.
- CADUCEUS⁷, un sistema medico sviluppato negli anni

³ <https://en.wikipedia.org/wiki/Mycin>

⁴ <https://en.wikipedia.org/wiki/Dendral>

⁵Hart, P. E., Duda, R. O., & Einaudi, M. T., *Prospector, a computer-based consultation system for mineral exploration*. Journal of the International Association for Mathematical Geology, 10(5), 589-610, 1977, <https://www.sri.com/wp-content/uploads/2021/12/739.pdf>

⁶ <https://en.wikipedia.org/wiki/Xcon>

⁷ [https://en.wikipedia.org/wiki/CADUCEUS_\(expert_system\)](https://en.wikipedia.org/wiki/CADUCEUS_(expert_system))

'80 basato sul predecessore INTERNIST-1, capace di affrontare diagnosticamente un ampio spettro di malattie umane e noto per essere uno dei sistemi esperti più “knowledge-intensive” dell'epoca.

Questi sistemi, pur basandosi su logiche simboliche e regole codificate manualmente, mostrarono in maniera evidente come l'IA potesse affrontare compiti complessi in domini specifici. Allo stesso tempo, crebbe l'interesse per le reti neurali artificiali⁸, modelli ispirati al funzionamento del cervello. Nonostante i progressi, questi approcci rimanevano fortemente dipendenti dalle regole scritte dall'uomo e dalle limitate risorse computazionali disponibili.

Un contributo decisivo allo sviluppo delle reti neurali è arrivato dal lavoro di John Hopfield e Geoffrey Hinton⁹. Nel 2024, i due ricercatori sono stati insigniti del premio Nobel per la fisica per aver utilizzato strumenti concettuali della fisica nello sviluppo di metodi che hanno posto le basi dell'apprendimento automatico moderno. Le reti neurali artificiali, grazie a questi contributi, sono diventate uno degli strumenti più potenti per l'analisi dei dati, trasformando profondamente la ricerca scientifica, l'ingegneria e la vita quotidiana.

La visione artificiale di LeCun

Nel 1989, in un'epoca in cui la “visione artificiale” era ancora dominata da regole scritte a mano e da lunghi elenchi di caratteristiche progettate dagli esperti, Yann LeCun¹⁰ e colleghi ai Bell Labs provarono una strada diversa: far imparare alla macchina non solo a classificare, ma anche quali dettagli

⁸ Parleremo delle reti neurali nel Capitolo 4.

⁹ Geoffrey Hinton, premio nobel per la Fisica nel 2024 è stato Professore presso l'Università di Toronto e ricercatore presso Google Brain. Insieme a Yann LeCun ha vinto il Turing Award nel 2018 (spesso definito il "Premio Nobel per l'informatica").

¹⁰ Yann LeCun è Professore all'Università di New York, ha ricevuto il Turing Award 2018, insieme a Yoshua Bengio e Geoffrey Hinton, per il loro lavoro sull'apprendimento profondo. I tre vengono talvolta definiti "padrini dell'intelligenza artificiale". E' stato Chief AI Scientist presso Meta.

guardare in un'immagine. Il banco di prova non era un giocattolo da laboratorio, bensì un problema concreto e industriale: riconoscere le cifre scritte a mano nei codici postali statunitensi.

A prima vista, distinguere dieci classi (0–9) sembra facile. In realtà, la scrittura a mano è un catalogo infinito di varianti: tratti più o meno marcati, inclinazioni, errori, numeri “mezzi” perché la penna si è sollevata, oppure cifre attaccate tra loro. Nel lavoro del 1989¹¹, il team usò un database di 9.298 cifre segmentate provenienti da posta reale passata per l'ufficio di Buffalo (New York), con una divisione in 7.291 esempi per l'addestramento e 2.007 per il test (più un insieme aggiuntivo di cifre stampate per confronti e robustezza).

Il cuore dell'esperimento è l'intuizione che oggi diamo per scontata: invece di trasformare l'immagine in un elenco di misure scelte a mano (spessore dei tratti, angoli, endpoints...), si può dare alla rete l'immagine quasi grezza e lasciare che apprenda da sola quali micro-pattern sono utili.

Per poterlo fare senza far esplodere il numero di parametri, LeCun impose nella struttura della rete alcuni vincoli ispirati al modo in cui le immagini funzionano:

- Connettività locale: un neurone non guarda tutta l'immagine, ma una piccola finestra (come un riquadro mobile).
- Condivisione dei pesi (weight sharing): lo stesso rilevatore (filtro) viene riutilizzato in posizioni diverse dell'immagine. Se un certo tratto è utile in alto a sinistra, è plausibile che lo sia anche altrove.
- Sotto-campionamento/averaging (l'antenato del pooling): dopo aver evidenziato certe caratteristiche, la rete riduce la risoluzione per diventare più tollerante a piccoli spostamenti e deformazioni.

Il modello non cerca subito di decidere “questo è un 8”, ma

¹¹ Y. LeCun et al., *Handwritten Digit Recognition with a Back-Propagation Network*, NIPS'89: Proceedings of the 3rd International Conference on Neural Information Processing Systems, <https://dl.acm.org/doi/10.5555/2969830.2969879>

costruisce una gerarchia di indizi, prima tratti e curvature locali, poi combinazioni più grandi, fino alla decisione finale.



Figura 6: L'esperimento di Yann LeCun (a destra)

Nell'esperimento di Yann LeCun (Figura 6), dopo circa 30 passate sul dataset di training, gli autori riportano un errore sul test complessivo del 3,4% (con errori che si concentrano sulle cifre scritte a mano). Ma il dato più interessante, da prospettiva applicativa, è la gestione del dubbio: in molti contesti reali conviene che il sistema, quando è incerto, dica “non lo so” e passi la palla a un operatore umano.

Gli autori misurano quindi quante cifre bisogna rifiutare per ottenere solo l'1% di errore: sul sottoinsieme di cifre manoscritte, serviva circa un 9% di reiezioni per arrivare a quel livello di accuratezza.

Questa idea, accuratezza più astensione controllata, è un tema modernissimo: la sicurezza dei sistemi di IA non dipende solo da quanto spesso “indovinano”, ma anche da quanto bene sanno riconoscere i casi fuori standard.

Internet e dati per l'IA

A partire dagli anni Novanta, l'intelligenza artificiale ha compiuto un ulteriore salto in avanti, favorito dalla diffusione di Internet e dalla crescente disponibilità di dati. Le reti neurali profonde, alla base del deep learning, hanno permesso di costruire modelli capaci di apprendere in modo sempre più autonomo. L'apprendimento automatico è diventato il cuore dell'IA moderna: invece di programmare ogni singola regola, si lascia che la macchina scopra da sola schemi e relazioni all'interno dei dati. Un momento simbolico di questa svolta è stato il successo di modelli come AlexNet, che ha rivoluzionato il riconoscimento delle immagini dimostrando le potenzialità del deep learning.

Negli anni più recenti, un'ulteriore trasformazione è arrivata con l'introduzione dell'architettura Transformer, da cui sono nati i cosiddetti Large Language Model (LLM). Addestrati su enormi quantità di testi, questi modelli sono in grado di comprendere e generare linguaggio naturale in modo sorprendentemente simile a quello umano. ChatGPT, lanciato nel 2022, è forse l'esempio più noto di questa nuova generazione di sistemi e ha contribuito a portare l'intelligenza artificiale al centro dell'attenzione pubblica.

Oggi l'intelligenza artificiale è una disciplina in continua evoluzione, profondamente intrecciata con la nostra vita quotidiana. Dalla medicina alla finanza, dalla robotica all'industria, le sue applicazioni si moltiplicano rapidamente, aprendo nuove opportunità ma anche nuove domande sul rapporto tra tecnologia, conoscenza e società.

2 LE MACCHINE SFIDANO L'UOMO

Nel corso della storia recente, l'intelligenza artificiale ha più volte messo in discussione ciò che consideriamo esclusivamente umano: la capacità di ragionare, intuire, decidere. Dai giochi da tavolo alla medicina, fino al linguaggio naturale, alcuni sistemi artificiali hanno raggiunto e in certi casi superato, le prestazioni dei migliori esperti umani. Questi episodi non rappresentano soltanto traguardi tecnologici, ma veri e propri momenti simbolici che hanno ridefinito il rapporto tra uomo e macchina.

ELIZA (1966)

ELIZA occupa un posto speciale nella storia dell'intelligenza artificiale perché mostra, in modo sorprendentemente semplice, quanto il linguaggio sia uno strumento potente per attribuire intelligenza a una macchina. Il programma fu sviluppato a metà degli anni Sessanta da Joseph Weizenbaum al MIT, in un periodo in cui l'IA stava cercando la propria identità tra ambizioni scientifiche e limiti tecnologici. ELIZA non era un cervello artificiale né un sistema capace di ragionare sul mondo. Era piuttosto un insieme ingegnoso di regole che riconoscevano parole chiave e trasformavano le frasi dell'utente in risposte plausibili. Il copione più famoso, chiamato DOCTOR, imitava lo stile di uno psicoterapeuta di approccio rogersiano, restituendo domande e riflessioni che invitavano a continuare a parlare. Fu proprio questa idea, cioè trattare la conversazione

come uno scambio in cui il programma rimanda all'utente le sue stesse parole sotto forma di domande e riformulazioni, a rendere ELIZA così efficace (Figura 7).

```
Welcome to

EEEEEE LL      IIII ZZZZZZ  AAAAA
EE      LL      II      ZZ  AA  AA
EEEEEE LL      II      ZZZ  AAAAAA
EE      LL      II      ZZ  AA  AA
EEEEEE LLLLLL  IIII ZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:
```

Figura 7: Un esempio di conversazione con ELIZA

Molte persone che la provarono ebbero l'impressione di essere comprese, o almeno ascoltate. Questa reazione non dipendeva da una reale comprensione del significato, ma dalla capacità del programma di mantenere la conversazione entro una forma credibile. ELIZA mette in evidenza un aspetto fondamentale dell'IA: la differenza tra ciò che un sistema fa davvero e ciò che noi percepiamo che stia facendo. La storia di ELIZA è anche la storia di come gli esseri umani interpretano il comportamento delle macchine quando quel comportamento assume la forma familiare della parola.

Da qui nasce quello che viene spesso chiamato “effetto ELIZA”, cioè la tendenza a proiettare intenzioni, emozioni e profondità mentale su un sistema che produce linguaggio convincente. È un tema che attraversa tutta la storia dell'IA e che oggi, con i modelli linguistici moderni, è più attuale che mai¹². ELIZA dimostrò che la conversazione può generare

¹² Torneremo su questo punto nel Capitolo 7 quando parleremo di

fiducia e coinvolgimento anche quando l'architettura sottostante è relativamente semplice. Questa lezione ha due facce. Da un lato, segnala la forza delle interfacce dialogiche e il loro valore pratico, perché parlare è un gesto naturale e universale. Dall'altro, avverte che la familiarità del linguaggio può indurci a sovrastimare le capacità reali di un sistema.

ELIZA non ha introdotto nessuna innovazione tecnica nel mondo dell'IA ma ha reso visibile un nodo cruciale del rapporto tra persone e tecnologia. Weizenbaum stesso rimase colpito dalla facilità con cui gli utenti si legavano al programma e divenne in seguito una voce critica rispetto all'uso indiscriminato di sistemi automatici in ambiti delicati. La vicenda di ELIZA introduce così una domanda che ogni epoca dell'IA si è trovata a ripetere: fino a che punto è legittimo delegare a una macchina compiti che richiedono giudizio umano, responsabilità e cura? Anche quando un sistema sembra avere una capacità dialettica convincente, non è detto che sappia ciò che dice, né che possa sostenere le conseguenze delle sue risposte.

Deep Blue e la caduta del re degli scacchi (1997)

Uno dei momenti più celebri nella storia dell'intelligenza artificiale è senza dubbio la sfida tra il computer Deep Blue e il campione mondiale di scacchi Garry Kasparov.

Il progetto Deep Blue affonda le sue radici nei primi anni Ottanta, come sviluppo accademico guidato dal ricercatore Feng-hsiung Hsu alla Carnegie Mellon University. Nato inizialmente come un sistema sperimentale chiamato ChipTest, e poi evoluto in Deep Thought, il progetto attirò presto l'attenzione per la sua impressionante capacità di analisi delle posizioni scacchistiche.

Alla fine degli anni Ottanta, IBM decise di investire nel progetto con un obiettivo ambizioso: costruire una macchina capace di sfidare i migliori giocatori del mondo. Il risultato fu Deep Blue, un sistema dotato di un'architettura parallela in grado di valutare

fino a 200 milioni di posizioni al secondo. Più che “pensare” come un essere umano, Deep Blue sfruttava una potenza di calcolo senza precedenti, combinata con un vastissimo archivio di partite storiche.

Il primo confronto con Kasparov avvenne nel 1996. Deep Blue vinse una partita, un evento senza precedenti, ma perse il match complessivo. Tuttavia, il vero colpo di scena arrivò nel 1997, quando una versione potenziata del sistema riuscì a vincere il match ufficiale contro il campione del mondo (Figura 8).



Figura 8: Garry Kasparov (a sinistra) contro Deep Blue, pilotato da Joe Hoane (a destra) uno dei creatori del progetto

Quel risultato segnò una svolta epocale: per la prima volta, una macchina aveva sconfitto l'uomo in uno dei giochi simbolo dell'intelligenza strategica. Le polemiche non mancarono, ma il messaggio era chiaro: con algoritmi adeguati e sufficiente potenza computazionale, anche l'intuizione umana poteva essere superata.

AlphaGo e l'intuizione artificiale (2016)

Se gli scacchi rappresentano una sfida complessa ma calcolabile, il gioco del Go è considerato per secoli un dominio quasi esclusivo dell'intuizione umana. Ed è proprio qui che AlphaGo ha cambiato le regole del gioco.

Sviluppato da DeepMind, AlphaGo è stato il primo sistema di intelligenza artificiale a sconfiggere un giocatore professionista di Go. A differenza di Deep Blue, AlphaGo non si basava sulla forza bruta del calcolo, ma su una combinazione innovativa di reti neurali profonde e apprendimento per rinforzo.

Dopo una fase iniziale di addestramento basata su partite umane, il sistema ha imparato giocando contro sé stesso, esplorando strategie mai sperimentate prima. Questo approccio gli ha permesso di sviluppare uno stile di gioco originale, spesso sorprendente anche per i maestri più esperti.

Nel 2016, AlphaGo affrontò il campione coreano Lee Sedol, uno dei più grandi giocatori di Go di tutti i tempi (Figura 9). La vittoria per quattro partite a una fece il giro del mondo. Memorabile fu la cosiddetta mossa 37 (Move 37): una mossa inizialmente giudicata “sbagliata”, ma che si rivelò decisiva.



Figura 9: AlphaGo (a sinistra) contro Lee Sedol (a destra)

Significativa la dichiarazione di Lee Sedol riguardo le capacità di AlphaGo: “Pensavo che AlphaGo si basasse sul calcolo delle probabilità e che fosse soltanto una macchina. Ma quando ho visto questa mossa, ho cambiato idea. AlphaGo è sicuramente creativo.”

Quell'episodio mostrò qualcosa di nuovo: l'IA non si limitava a imitare l'uomo, ma era capace di scoprire soluzioni creative,

ampliando la comprensione stessa del gioco.

Google DeepMind (2020)

Nel 2020 Google DeepMind ha presentato un sistema di intelligenza artificiale progettato per individuare precocemente il tumore al seno a partire dalle mammografie di screening. L'idea è semplice: riconoscere i segnali della malattia quando è ancora nelle fasi iniziali, cioè nel momento in cui le terapie hanno, in media, maggiori probabilità di successo.

Anche se i programmi di screening mammografico sono diffusi in tutto il mondo, leggere correttamente una mammografia non è banale. I radiologi devono prendere decisioni su immagini complesse, e non di rado si verificano errori di due tipi: falsi positivi (quando un esame sembra sospetto ma in realtà non c'è un tumore) e falsi negativi (quando un tumore presente non viene riconosciuto). I primi possono portare a ulteriori controlli e preoccupazioni inutili; i secondi, più gravi, possono tradursi in diagnosi mancate o ritardate.

Il sistema di DeepMind è stato addestrato proprio per ridurre questi due tipi di errore e migliorare la capacità predittiva rispetto ai lettori umani (Figura 10). Per valutare il comportamento in condizioni realistiche, i ricercatori hanno utilizzato un grande set di dati proveniente dal Regno Unito e un secondo set, arricchito, dagli Stati Uniti. I risultati hanno mostrato una diminuzione dei falsi positivi e dei falsi negativi: negli Stati Uniti rispettivamente del 5,7% e del 9,4%, e nel Regno Unito dell'1,2% e del 2,7%. Un aspetto importante è che il modello ha mantenuto buone prestazioni anche passando da un contesto all'altro, suggerendo una certa capacità di generalizzazione tra sistemi sanitari e pratiche cliniche differenti.

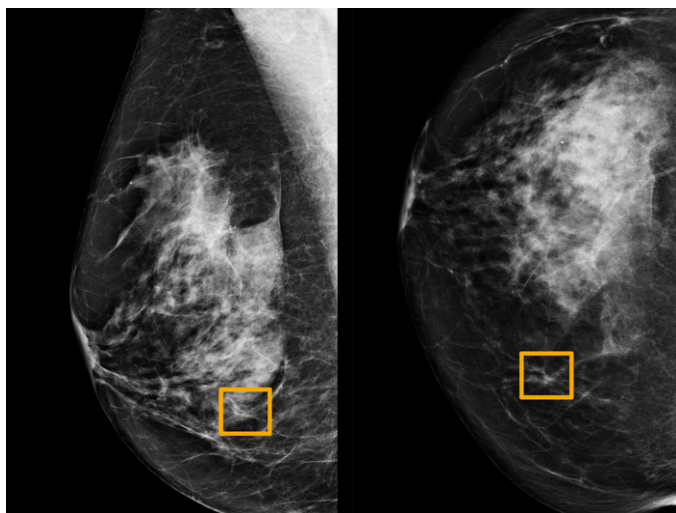


Figura 10: Il sistema di screening mammografico di DeepMind

Un ulteriore studio indipendente, condotto su un gruppo di sei radiologi, ha confrontato direttamente le prestazioni del sistema con quelle umane. In questa valutazione l'IA ha ottenuto risultati superiori: la sua accuratezza, misurata tramite l'area sotto la curva ROC (AUC-ROC), ha superato quella dei radiologi con un margine assoluto dell'11,5%. Infine, in una simulazione che riproduceva il processo di doppia lettura adottato nel Regno Unito (in cui due professionisti esaminano lo stesso esame), l'IA ha raggiunto prestazioni almeno comparabili a quelle umane e ha ridotto dell'88% il carico di lavoro del secondo lettore.

Nel complesso, questo caso mostra come l'intelligenza artificiale possa diventare uno strumento di supporto concreto nella diagnostica: non sostituisce automaticamente i clinici, ma può contribuire a rendere lo screening più accurato e, in certi contesti, anche più efficiente.

ChatGPT (2022)

Quando OpenAI presentò GPT-3 nel 2020, molti studiosi e addetti ai lavori parlarono di un "cambio di passo" nel trattamento automatico del linguaggio. Il nome completo,

Generative Pre-trained Transformer 3, dice già molto: un sistema generativo (capace di produrre testo), pre-addestrato su grandi quantità di dati e basato sull'architettura Transformer, che in quegli anni si stava affermando come una delle più efficaci per lavorare con sequenze linguistiche.

Il lancio pubblico di ChatGPT da parte di OpenAI avvenne il 30 novembre 2022, presentato come una “research preview”: un'interfaccia conversazionale costruita su un modello della famiglia GPT-3.5, capace di rispondere a domande, riassumere testi, aiutare nella scrittura e persino generare codice a partire da semplici istruzioni in linguaggio naturale.

Nel giro di pochissimi giorni lo strumento diventò virale, mostrando al grande pubblico—non solo agli specialisti—quanto un modello linguistico potesse sembrare “dialogante” e utile in compiti quotidiani: un momento spartiacque che ha contribuito ad accelerare l'interesse verso l'IA generativa e a rendere l'interazione con i computer più simile a una conversazione che a un comando (Figura 11).

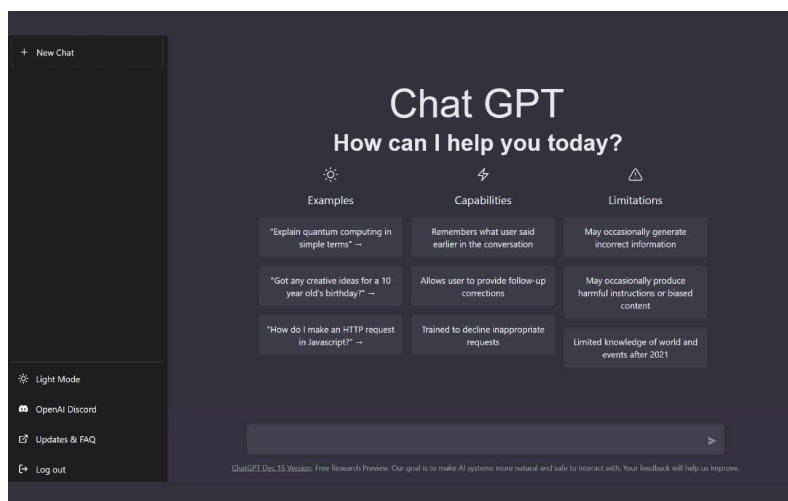


Figura 11: Una delle prime versioni di ChatGPT

La caratteristica che rese GPT-3 così discusso fu anche la sua scala: 175 miliardi di parametri, un numero enorme per l'epoca. In termini semplici, questi parametri sono come “manopole”

interne che il modello regola durante l'addestramento per imparare regolarità, stili e relazioni nel linguaggio. Per farlo, GPT-3 venne esposto a una massa vastissima di testi: pagine web, libri, articoli e altre fonti. Il risultato fu una competenza linguistica sorprendente: il modello poteva continuare un discorso, rispondere a domande e imitare registri diversi con una fluidità che, per molti, avvicinava la macchina al modo umano di scrivere.

Alla base dei Transformer c'è un meccanismo chiave chiamato self-attention. Invece di leggere una frase solo "in fila", parola dopo parola, il modello valuta quali termini siano più importanti per interpretare gli altri, anche se distanti nel testo. Questo aiuta a cogliere collegamenti, riferimenti e dipendenze complesse: il significato non emerge dalle singole parole isolate, ma dalla rete di relazioni tra di esse. È uno dei motivi per cui GPT-3 riesce spesso a mantenere coerenza e contesto, almeno per tratti relativamente lunghi di conversazione o scrittura.

Un altro elemento di rottura fu il modo in cui il modello veniva utilizzato. Molti sistemi precedenti erano costruiti e ottimizzati per un compito preciso (traduzione, classificazione, riassunto, etc). GPT-3, invece, mostrava un comportamento più generalista: con un'istruzione scritta in linguaggio naturale e qualche esempio, poteva adattarsi a richieste molto diverse senza essere riaddestrato ogni volta. Questa flessibilità ha accelerato l'adozione dell'IA in applicazioni concrete: assistenti virtuali, supporto alla scrittura, help desk, strumenti per la produttività e, in certa misura, anche per la programmazione.

La capacità più evidente e più affascinante per il grande pubblico, rimane però quella generativa. GPT-3 può produrre testi lunghi e strutturati: racconti, email, descrizioni di prodotto, dialoghi, slogan, persino poesie. Da qui la spinta decisiva alla cosiddetta Generative AI, cioè quell'insieme di tecniche che non si limitano ad analizzare i contenuti, ma li creano. Questa nuova potenza creativa ha aperto opportunità, ma ha anche reso più urgenti alcune domande: quanto controllo abbiamo su ciò che il modello produce? Come gestire errori, allucinazioni¹³ o

¹³ L'allucinazione è un fenomeno tipico dei LLM che consiste nella generazione di risposte inventate, come un evento storico mai successo, una

contenuti manipolativi? Quali responsabilità ricadono su chi usa questi strumenti e su chi li integra in servizi reali?

C'è chi ha sostenuto che GPT-3 e ancor più i modelli successivi come GPT-4 e GPT-5, abbiano di fatto superato il test di Turing. È un'affermazione controversa: il test, nella sua formulazione classica, valuta se una macchina possa risultare indistinguibile da un essere umano in una conversazione. Alcuni modelli moderni possono certamente ingannare un interlocutore in molte situazioni, ma questo non implica necessariamente comprensione nel senso umano del termine. Più correttamente, GPT-3 ha mostrato quanto sia facile confondere fluidità linguistica e intelligenza.

In ogni caso, l'impatto di GPT-3 è stato enorme: ha spinto aziende, ricercatori e istituzioni a ripensare strategie e investimenti sulla linguistica computazionale e sull'IA generativa. E ha preparato il terreno alla generazione successiva di modelli, sempre più capaci, come quelli rilasciati negli anni seguenti, come il recente GPT-5, che hanno ulteriormente ampliato qualità, velocità e modalità di interazione con le macchine.

Una caratteristica innovativa nella modalità di esecuzione dei recenti modelli come GPT-5 è stata l'inserimento dell'architettura ad agenti che ha permesso di estendere le conoscenze del modello con l'utilizzo di strumenti esterni come la ricerca sul web.

Se il modello non ha una conoscenza di un argomento può decidere (in autonomia) di eseguire una ricerca su internet e utilizzare le informazioni per fornire una risposta. Questa modalità è particolarmente utile per ampliare la conoscenza del modello su fatti recenti o addirittura in tempo reale come ad esempio la temperatura attuale in una data località.

3. MACHINE LEARNING

Il machine learning (in italiano apprendimento automatico) è una delle aree più vivaci dell'intelligenza artificiale. L'idea di fondo è semplice: invece di programmare a mano ogni regola, si costruiscono algoritmi che imparano dai dati, scoprendo da soli schemi e relazioni utili per fare previsioni e prendere decisioni. Per capire perché si parla di “cambio di paradigma”, basta confrontare questo approccio con la programmazione tradizionale. Nel software classico, lo sviluppatore scrive le istruzioni, inserisce i dati in input e ottiene un output: il comportamento del programma dipende dalle regole che qualcuno ha codificato. Nel machine learning, invece, il programmatore non scrive direttamente tutte le regole: prepara un modello e gli fornisce esempi (dati di input e risultati attesi in output). A quel punto, è il modello che aggiusta i propri parametri finché non riesce a riprodurre la relazione tra input e output, diventando capace di generalizzare su nuovi casi (Figura 12).

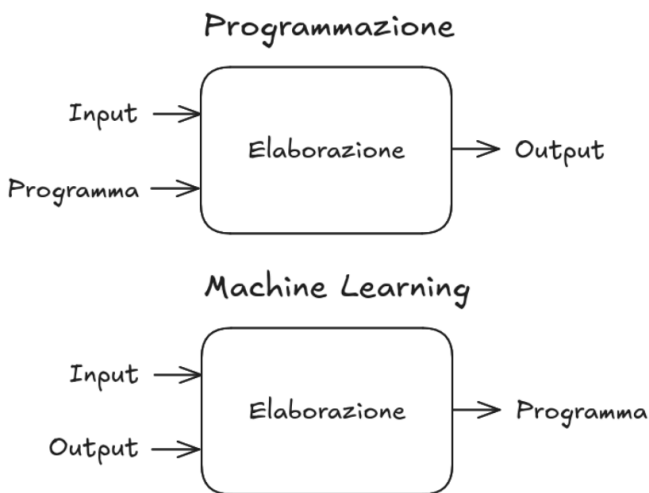


Figura 12: Differenza tra programmazione e machine learning

In genere, il modo in cui un modello impara viene suddiviso in tre grandi famiglie:

- **apprendimento supervisionato** (supervised learning);
- **apprendimento non supervisionato** (unsupervised learning);
- **apprendimento per rinforzo** (reinforcement learning).

Apprendimento supervisionato

Nell'apprendimento supervisionato il modello viene addestrato con dati etichettati: per ogni esempio viene fornita la risposta corretta. Il compito dell'algoritmo è imparare la mappa che collega input e output e migliorare progressivamente le sue previsioni man mano che vede nuovi esempi.

Gli esempi più comuni sono:

- **classificazione**, come distinguere immagini di *cani* e *gatti* dopo aver osservato migliaia di immagini già

etichettate;

- **regressione**, come stimare il prezzo di una casa a partire da caratteristiche quali metratura, numero di stanze, zona, e così via.

Quando i dati etichettati sono abbondanti e di buona qualità, questo approccio può offrire risultati molto accurati. Tra le tecniche più note rientrano le reti neurali¹⁴, le macchine a vettori di supporto (SVM¹⁵) e i modelli di regressione.

Apprendimento non supervisionato

L'apprendimento non supervisionato parte da dati senza etichette: nessuno dice al modello qual è la risposta giusta. L'algoritmo deve quindi esplorare il dataset e individuare strutture nascoste, somiglianze e raggruppamenti.

Un esempio classico è il clustering, dove i dati vengono divisi in gruppi di elementi simili: l'algoritmo k-means¹⁶, per esempio, crea cluster separati sulla base della distanza tra i punti. Questo tipo di analisi è molto utile nel marketing e nei servizi digitali: si possono segmentare i clienti in gruppi con comportamenti simili e personalizzare offerte o comunicazioni anche senza etichette esplicite.

Un'altra famiglia importante è la riduzione della dimensionalità, che serve a semplificare dataset molto complessi. Tecniche come la PCA (analisi delle componenti principali) aiutano a ridurre il numero di variabili preservando quanta più informazione possibile, rendendo i dati più interpretabili e più facili da visualizzare.

Apprendimento per rinforzo

L'apprendimento per rinforzo segue una logica diversa, ispirata

¹⁴ Introdurremo le reti neurali nel Capitolo 4.

¹⁵ https://it.wikipedia.org/wiki/Macchine_a_vettori_di_supporto

¹⁶ k-means è un algoritmo di clustering non supervisionato che suddivide i dati in k gruppi minimizzando la distanza tra ogni punto e il centroide del gruppo a cui appartiene.

al comportamento: un algoritmo (detto agente) interagisce con un ambiente e impara tramite ricompense e penalità. L'obiettivo non è “indovinare la risposta corretta” su un dataset statico, ma trovare una strategia che massimizzi la ricompensa complessiva nel tempo.

È lo stesso principio alla base di sistemi celebri come Deep Blue e AlphaGo: l'agente gioca, commette errori, riceve feedback e migliora. Questo approccio è particolarmente utile in contesti in cui bisogna prendere decisioni in sequenza e adattarsi a situazioni dinamiche, come robotica e guida autonoma. Algoritmi come il Q-learning¹⁷ e, più in generale, il deep reinforcement learning (che combina rinforzo e reti neurali profonde) hanno portato a risultati notevoli in compiti che richiedono autonomia e capacità di adattamento.

In Figura 13 è riportato uno schema che riassume le differenze tra apprendimento supervisionato, non supervisionato e per rinforzo.

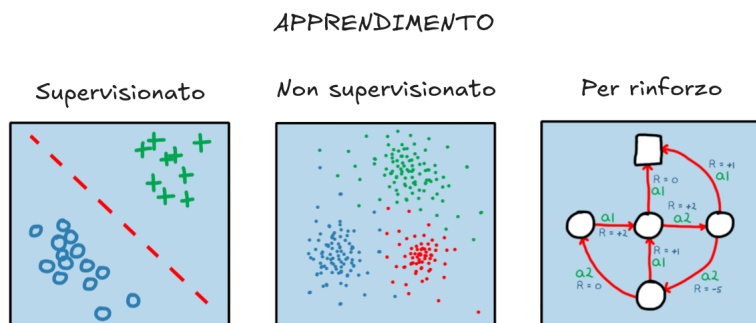


Figura 13: Apprendimento supervisionato, non supervisionato e per rinforzo

¹⁷ <https://en.wikipedia.org/wiki/Q-learning>

Esempio: le telecamere di videosorveglianza

Le telecamere di sorveglianza producono flussi continui di immagini o video. Un sistema di machine learning può svolgere diversi compiti di classificazione, ad esempio:

- rilevare se c'è una persona nel frame¹⁸;
- classificare il tipo di oggetto: persona, auto, bici, animale;
- riconoscere eventi: attraversamento di una linea, oggetto abbandonato, assembramento.

Ad esempio, possiamo classificare un frame (o una sua porzione) per capire se contiene una certa classe di interesse come un'auto, un autobus, un furgone, etc (Figura 14).



Figura 14: Riconoscimento di oggetti in un fermo immagine (frame)

Un sistema di machine learning applicato ad un flusso di frame esegue tipicamente i seguenti passi:

1. Scelta del frame ogni tot millisecondi (processo di campionatura);
2. Elaborazione dell'immagine per ridurre le dimensioni, regolare la luminosità, rimuovere il rumore, etc;
3. Applicazione del modello di analisi, tipicamente basato

¹⁸ Un frame è l'immagine di un fotogramma di un video. Un video digitale contiene tipicamente 30 frame al secondo.

su reti neurali (su architetture CNN¹⁹) con la produzione di una probabilità per ciascuna classe (es. persona 0,92; auto 0,06; altro 0,02);

4. Esecuzione della decisione, se la probabilità supera una soglia, scatta un'azione:
 - invio alert;
 - registrazione di un segmento video;
 - attivazione di un operatore umano;
 - conteggio statistico.

Ad esempio, immaginiamo un semplice sistema in grado di rilevare la presenza di una persona in un video.

Le classi del modello di machine learning possono essere:

- Classe 1: c'è almeno una persona visibile
- Classe 0: non c'è nessuna persona

Il modello impara a riconoscere una persona dagli schemi visivi (pattern) appresi durante la fase di addestramento come contorni, texture, forme ricorrenti, composizioni (testa-spalle, arti, silhouette), e in generale correlazioni statistiche tra pixel²⁰ e etichette.

Questi modelli non restituiscono un valore binario sì o no ma una distribuzione di probabilità²¹. Ad esempio, nel sistema di classificazione della persona un risultato potrebbe essere:

- frame A → persona: 0,97 (probabilità del 97%)
- frame B → persona: 0,62 (probabilità del 62%)
- frame C → persona: 0,12 (probabilità del 12%)

Per poter dare una previsione, è necessario fissare una soglia, ad esempio 0,80 (probabilità dell'80%). Se il risultato è maggiore di

¹⁹ Nel Capitolo 4 parleremo delle architetture CNN.

²⁰ Un pixel è l'elemento fondamentale di un'immagine digitale. Un'immagine in un computer è una matrice di punti (pixel). Ogni pixel può avere un colore diverso rappresentato tramite dei numeri.

²¹ La probabilità è una stima del verificarsi di un evento, espressa con un numero tra 0 e 1, oppure in percentuale tra 0% e 100%. Quando i valori della probabilità sono 0 o 1 l'evento è certo, ossia è sicuro che l'evento non si verifichi (0) oppure è certo che si verifichi (1).

0,80 allora possiamo essere ragionevolmente sicuri che ci sia una persona nel frame analizzato, altrimenti no. Questa è una stima, il che significa che questo approccio probabilistico può sbagliare la previsione.

Cambiando il valore di questa soglia cambia il comportamento del sistema:

- Una soglia più bassa implica più segnalazioni (ma anche più falsi allarmi)
- Una soglia più alta offre meno falsi allarmi (ma rischi di perdere eventi reali)

In generale, si parla di falso positivo quando il sistema segnala una persona che non c'è (ad esempio ombre, manichini, poster, riflessi). Avere troppi falsi positivi in un sistema di machine learning può comportare allarmi inutili, operatori sovraccarichi e in generale una perdita di fiducia nel sistema.

Invece, un falso negativo è quando il sistema non vede la presenza di una persona, ad esempio a causa di una scarsa illuminazione, occlusioni, angoli sfavorevoli, etc. Le conseguenze di un falso negativo sono tendenzialmente più gravi di un falso positivo, soprattutto per un sistema di videosorveglianza.

Dunque, la valutazione della soglia di attivazione di un sistema di machine learning non è solo legata all'accuratezza, ma un equilibrio tra:

- sensibilità (quanti eventi veri intercetto),
- precisione (quanti allarmi sono davvero corretti),
- robustezza a condizioni reali (notte, pioggia, controllo luce, compressione video).

Esempio: la stima del prezzo di una casa

Immaginiamo un'agenzia immobiliare che voglia creare un sistema per fare una prima stima del prezzo di un appartamento. L'idea è semplice: se conosciamo alcune caratteristiche della casa, come ad esempio i metri quadri, possiamo provare a stimare il prezzo. Quando l'obiettivo è prevedere un valore numerico, in intelligenza artificiale si parla

spesso di regressione.

Un sistema di regressione è un modello che:

1. osserva esempi passati, nell'esempio case già vendute con le loro caratteristiche;
2. cerca un legame tra caratteristiche e prezzo (utilizzando delle funzioni matematiche);
3. usa quel legame per fare una previsione su una nuova casa.

La versione più intuitiva è quella che prova a descrivere la relazione con una linea (una funzione lineare): al crescere dei metri quadri, in media cresce anche il prezzo. È un modello semplice, ma molto utile come punto di partenza.

Supponiamo di avere un piccolo archivio di vendite, per alcune case sappiamo quanti metri quadri hanno e a quanto sono state vendute. Non serve che i dati siano perfetti, nella realtà due case della stessa dimensione possono costare diversamente a seconda della zona, del piano, dello stato dell'immobile, etc, quindi nel nostro modello stiamo semplificando.

Il modello impara così:

- guarda le coppie “metri quadri / prezzo”;
- prova a tracciare una linea che passi il più possibile vicino ai punti;
- decide quanto deve “salire” la linea quando aumenta la superficie.

Addestrare, qui, non vuol dire individuare delle regole o degli schemi. Vuol dire scegliere automaticamente i parametri della linea in modo che, in media, le previsioni siano vicine ai prezzi reali.

Per capire l'idea, proviamo a ragionare nel modo seguente:

- se la linea è troppo in basso, il modello sottostima quasi tutte le case;
- se è troppo in alto, le sovrastima;
- se è troppo ripida, aumenta troppo velocemente con i metri quadri;
- se è troppo piatta, non aumenta abbastanza.

L'addestramento è proprio la ricerca della linea ottimale, quella più vicina rispetto ai dati disponibili. Questa fase, nel modello di

regressione, è spesso affidata ad un problema di ottimizzazione utilizzando varie tecniche come la discesa del gradiente.

La discesa del gradiente è un metodo iterativo che aggiusta i parametri di una linea iniziale passo dopo passo. A ogni passo si guarda in che direzione l'errore aumenterebbe più rapidamente se cambiassimo leggermente pendenza e intercetta: questa direzione di salita è l'idea dietro il gradiente. Se vogliamo ridurre l'errore, ci muoviamo nel verso opposto, cioè scendiamo lungo il paesaggio dell'errore, come se fosse una superficie fatta di colline e vallate: l'altezza è l'errore, e noi cerchiamo una valle. La dimensione dei passi è cruciale, il cosiddetto tasso di apprendimento: se è troppo grande rischiamo di saltare oltre la valle e oscillare, se è troppo piccolo arriviamo lentamente. Ripetendo questo aggiornamento molte volte, la linea tende a stabilizzarsi in una configurazione in cui piccoli cambiamenti non portano più a un miglioramento apprezzabile: è il punto in cui, per quella misura di errore, abbiamo di fatto ottimizzato la linea di regressione.

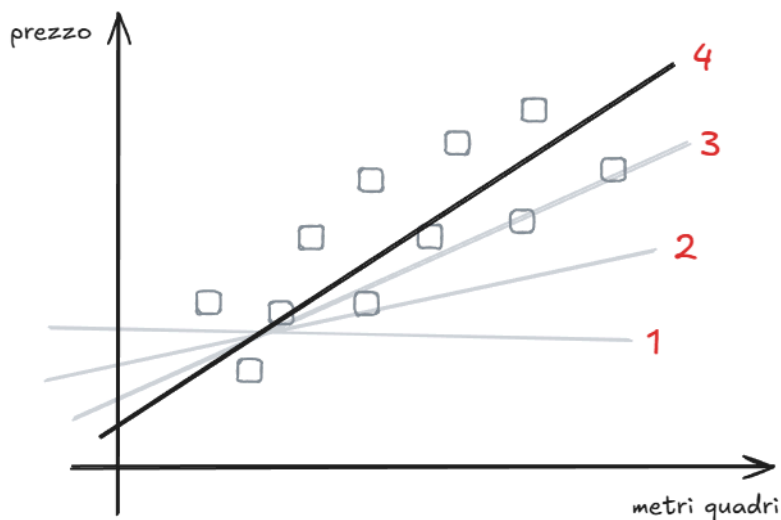


Figura 15: Esempio di discesa del gradiente su una regressione lineare

In Figura 15 è riportato uno schema semplificato di

ottimizzazione di una retta tramite discesa del gradiente. I punti indicati con dei piccoli quadrati sono i dati storici sul rapporto tra prezzo delle case (ordinata) e metri quadri (ascissa). L'algoritmo del gradiente parte da una linea iniziale più o meno casuale, indicata con il numero 1, e passo dopo passo cerca di minimizzare la distanza (l'errore) con i punti. Nel grafico dopo solo quattro passaggi si arriva alla linea 4 che rappresenta la migliore linea possibile, ossia quella che ha la distanza minore con tutti i punti dei dati storici.

Una volta addestrato, il modello è di fatto una funzione lineare che si può utilizzare per predire il prezzo di nuove case:

- si inseriscono i metri quadri della casa nuova;
- il modello restituisce un prezzo stimato.

Questa previsione è ovviamente una stima. La sua utilità sta nel fornire un valore coerente con ciò che si è visto in passato, e nel farlo in modo immediato.

Per avere un metodo di valutazione della bontà dell stima è necessario confrontare il modello con alcuni dati reali. Di solito, avendo un insieme di dati storici quello che si fa è dividerli in due sottoinsiemi: un insieme da utilizzare per l'addestramento (training) e un insieme per la valutazione (test). Di solito si utilizza circa l'80% dei dati per il training e il restante 20% per il test.

Se nel test il modello sbaglia poco, significa che sta cogliendo una regolarità reale e non solo memorizzando esempi. Nella realtà si utilizzano delle metriche, ossia dei valori che indicano la bontà della stima. Uno dei più utilizzati è il valore di R^2 (coefficiente di determinazione) che indica quanta variabilità del valore reale viene spiegata dal modello. Se il valore è 1 vuol dire che il modello è perfetto²², se il valore è 0 vuol dire che il modello si comporta esattamente come il valore medio e quindi poco utile. Questo valore può anche essere negativo se il modello va peggio della media.

E' importante sottolineare che il modello di machine learning non stima il prezzo delle case attraverso un ragionamento, come farebbe una persona. Il modello impara dai dati forniti ed

²² Nella realtà non esistono modelli di machine learning perfetti con valore di R^2 pari a 1. Di solito un buon modello ha un valore vicino a 0,9.

estrapola una regola generica per fare previsioni. Il calcolo al metro quadro è suggerito dagli esempi passati (dati storici).

Da qui si deduce l'importanza dei dati. Senza i dati, senza un'evidenza storica, i modelli di machine learning sono inutili.

L'esempio della stima del prezzo di una casa a partire dai metri quadri è volutamente semplice. Nella realtà i prezzi dipendono da molte cose, e bisogna considerare altre variabili come:

- quartiere o distanza dal centro;
- numero di stanze;
- piano e presenza dell'ascensore;
- stato dell'immobile;
- servizi nei dintorni.

In più, una linea può essere una buona approssimazione, ma non sempre: a volte la relazione non è così regolare. In questo caso si utilizzano funzioni non lineari.

4. RETI NEURALI ARTIFICIALI

Le reti neurali artificiali sono modelli matematici e computazionali che, almeno nell'idea di base, prendono spunto dal cervello umano. L'obiettivo non è “copiare” davvero la biologia, ma usare un'ispirazione semplice: tante unità elementari che si scambiano segnali e, insieme, riescono a imparare schemi complessi e a riconoscere regolarità nei dati.

Per capire da dove nasce questa metafora, facciamo un passo indietro e osserviamo com'è fatto un neurone biologico (Figura 16).

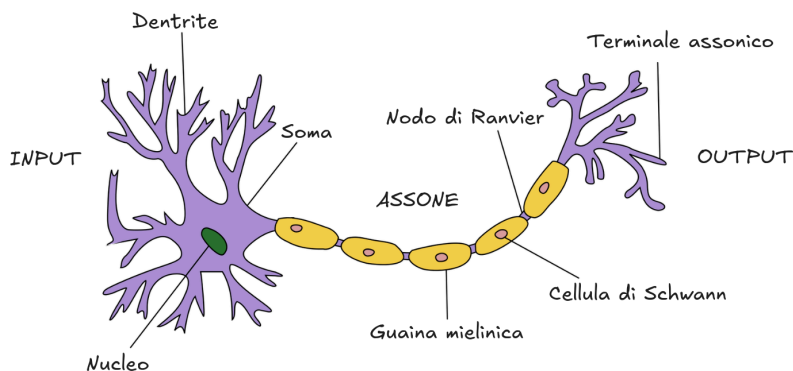


Figura 16: Un modello di neurone

Un neurone è una cellula specializzata del sistema nervoso: riceve informazioni, le combina e decide se trasmette un nuovo segnale ad altre cellule.

I dendriti sono le ramificazioni che funzionano come antenne: raccolgono segnali in arrivo da molti altri neuroni e li convogliano verso il centro della cellula. Le spine dendritiche, piccole protuberanze lungo i dendriti, aumentano la superficie di contatto e facilitano la ricezione dei segnali.

Nel soma (il corpo cellulare) si trova il nucleo, che contiene il DNA e regola le attività fondamentali della cellula. Possiamo immaginarlo come la “sala macchine”: qui i segnali ricevuti vengono integrati e, se l'insieme delle informazioni in entrata supera una certa soglia, il neurone genera un impulso.

Quell'impulso viaggia lungo l'assone, una sorta di cavo che porta il segnale dal soma verso altre cellule. Spesso l'assone è rivestito da mielina, un isolamento che accelera la trasmissione. La mielina non è continua: presenta interruzioni chiamate nodi di Ranvier, che rendono la propagazione più rapida grazie alla cosiddetta conduzione saltatoria.

Alla fine dell'assone troviamo la terminazione sinaptica (o terminale assonico), dove il segnale elettrico viene trasformato in un segnale chimico: vengono rilasciati neurotrasmettitori nello spazio di connessione tra neuroni (la sinapsi). Queste sostanze si legano ai recettori del neurone successivo e permettono all'informazione di proseguire.

In breve: un neurone biologico riceve, integra e trasmette. Ed è proprio questa catena logica, non i dettagli cellulari, che ha ispirato i modelli artificiali.

Un neurone artificiale (originariamente conosciuto come perceptron²³) può essere rappresentato come in Figura 17.

²³ Abbiamo parlato nel Capitolo 1 del perceptron ideato da Rosenblatt.

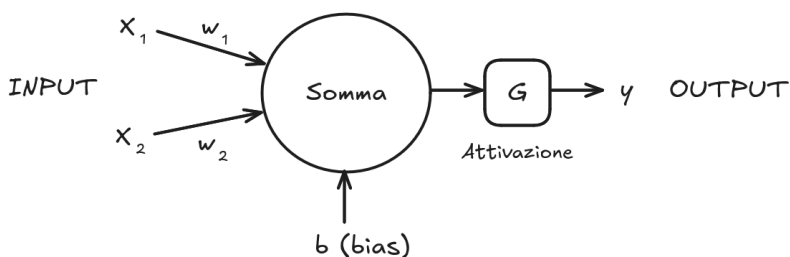


Figura 17: Schema di un neurone artificiale

Immaginiamo, per semplicità, un neurone con due ingressi x_1 e x_2 (input) e un'uscita y (output). A ciascun ingresso è associato un peso: w_1 per x_1 e w_2 per x_2 . I pesi indicano quanto conta quel segnale. Il neurone somma la combinazione degli ingressi con i pesi utilizzando la seguente formula $x_1 * w_1 + x_2 * w_2 + b$.

Il parametro b è indicato con il nome di bias, ossia un valore aggiuntivo che permette di spostare la risposta del neurone, rendendolo più flessibile e più preciso nel rappresentare relazioni tra i dati.

Il risultato di questa somma viene passato a una funzione di attivazione G che introduce una non linearità: è ciò che permette alle reti neurali di andare oltre a una semplice somma e di modellare fenomeni complessi. Una delle attivazioni più diffuse è la ReLU (Rectified Linear Unit): se il valore in ingresso è negativo, l'uscita diventa 0; se è positivo, resta uguale.

La Figura 18 mostra proprio questo comportamento “a soglia morbida”: la ReLU lascia passare i valori positivi e spegne quelli negativi.

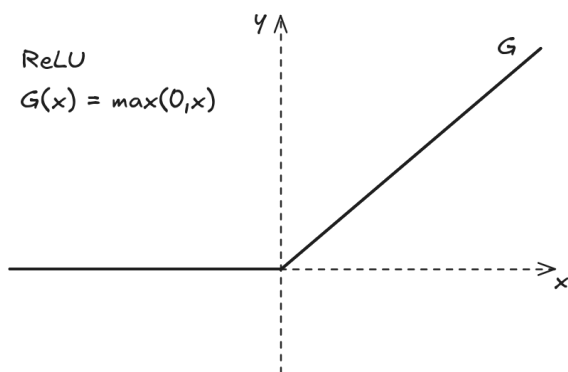


Figura 18: La funzione di attivazione ReLU

Reti neurali profonde

Un neurone da solo è interessante, ma la vera potenza nasce quando molti neuroni vengono collegati tra loro. Le reti neurali sono organizzate in strati (Figura 19):

- **strato di input**, che riceve i dati grezzi;
- **uno o più strati nascosti**, che trasformano progressivamente l'informazione;
- **strato di output**, che produce il risultato finale (una previsione, una classe, un punteggio, etc).

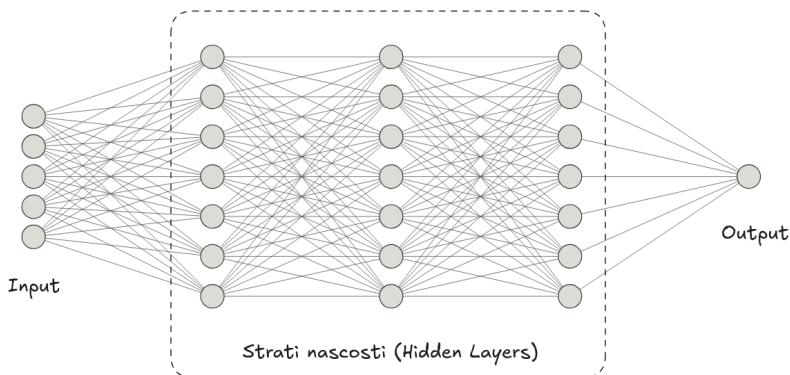


Figura 19: Rete neurale con 3 strati nascosti ognuno composto da 7 neuroni

Nella maggior parte delle architetture, ogni neurone di uno strato è collegato con tutti i neuroni dello strato successivo e ogni collegamento ha un proprio peso.

Quando una rete neurale ha più di tre strati nascosti viene chiamata rete neurale profonda (deep neural network). Ad esempio, il modello GPT-4 utilizza diverse reti neurali strutturate con circa 120 strati nascosti²⁴.

Il funzionamento di una rete neurale si basa su un processo iterativo che cerca di ottimizzare i pesi di connessione per minimizzare l'errore tra l'output prodotto e il risultato atteso. Quando una rete è addestrata, i dati di input vengono trasformati attraverso una serie di operazioni matematiche, principalmente operazioni lineari e non lineari, che permettono al modello di identificare schemi (pattern) o caratteristiche nei dati. Queste operazioni sono ispirate a funzioni biologiche, sebbene la loro implementazione differisca notevolmente dai meccanismi effettivi del cervello umano.

Un aspetto chiave delle reti neurali è la funzione di attivazione, che permette ai neuroni di accendersi o spegnersi a seconda del valore del segnale in input. Abbiamo visto la funzione di attivazione ReLU ma ne esistono di diverse come la funzione softmax, ognuna delle quali svolge un ruolo cruciale nel determinare la natura delle trasformazioni all'interno della rete.

Le reti neurali vengono addestrate mediante algoritmi di ottimizzazione, il più noto dei quali è la discesa del gradiente, che abbiamo già introdotto nel Capitolo 3. Durante l'addestramento, la rete confronta il risultato prodotto con quello atteso e calcola l'errore; successivamente, attraverso un processo di retropropagazione (backpropagation), questo errore viene utilizzato per aggiornare i pesi dei collegamenti in modo che la rete si avvicini progressivamente alla soluzione corretta. La retropropagazione ha rappresentato un'innovazione fondamentale nello sviluppo delle reti neurali, poiché ha reso possibile l'addestramento di reti con più strati nascosti, aprendo così la strada all'uso di reti neurali profonde.

²⁴ <https://medium.com/@daniellefranca96/gpt4-all-details-leaked-48fa20f9a4a>

Le reti neurali possono essere suddivise in vari tipi a seconda dell'architettura e dell'applicazione. Le reti feed-forward, in cui i segnali si muovono in una sola direzione, dall'input all'output, rappresentano la forma più semplice. In contesti più avanzati, si utilizzano reti neurali ricorrenti (Recurrent Neural Network, RNN) e reti neurali convoluzionali (Convolutional Neural Network, CNN).

Recurrent Neural Network

Una RNN (Recurrent Neural Network) è una rete neurale pensata per trattare sequenze: testo, audio, serie temporali, dati in cui l'ordine conta. A differenza di una rete "feed-forward" classica, che elabora ogni input come indipendente, una RNN mantiene una sorta di memoria interna (lo stato nascosto) che viene aggiornata passo dopo passo. In pratica, mentre legge la sequenza, la rete combina l'elemento corrente con ciò che ha già analizzato prima, producendo ogni volta una nuova rappresentazione dello stato. Questo le permette, per esempio, di interpretare una parola in una frase tenendo conto del contesto precedente.

Per addestrare una RNN serve capire quanto ogni decisione precedente abbia contribuito all'errore finale. Qui entra in gioco la backpropagation: la tecnica che calcola in modo efficiente come modificare i pesi della rete per ridurre l'errore (Figura 20).

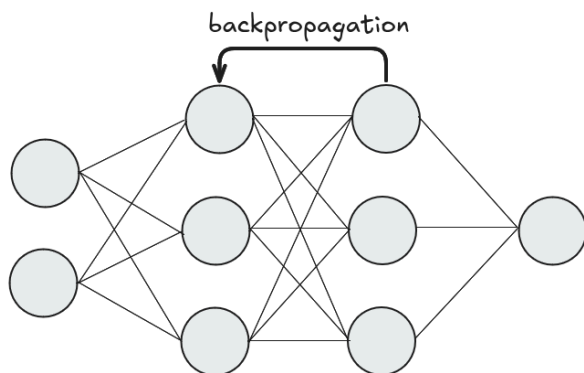


Figura 20: Esempio di rete RNN

Nel caso delle RNN si usa spesso la Backpropagation Through Time²⁵ (BPTT), che “srotola” idealmente la rete nel tempo: la stessa rete viene vista come una catena di copie collegate, una per ogni passo della sequenza. L'errore viene propagato all'indietro lungo questa catena, così da aggiornare i pesi in modo coerente con l'influenza che hanno avuto nei vari istanti. L'importanza della backpropagation, e della sua versione “nel tempo”, è l'elemento fondamentale per imparare dai propri errori, senza di essa la RNN non potrebbe attribuire correttamente il merito o la colpa alle informazioni passate, e quindi non riuscirebbe a riconoscere strutture nella sequenza. Allo stesso tempo, questa procedura rende evidente anche un limite storico delle RNN, su sequenze molto lunghe il segnale di errore può diventare troppo piccolo o troppo grande (vanishing/exploding gradients), motivo per cui in molte applicazioni si usano varianti come Long Short-Term Memory²⁶ (LSTM) o Gated Recurrent Unit²⁷ (GRU), progettate per gestire meglio la memoria a lungo termine.

Convolutional Neural Network

Le reti neurali convoluzionali (CNN) sono un tipo di rete neurale progettata per elaborare e analizzare dati strutturati in griglie, come immagini, utilizzando strati di convoluzione che rilevano automaticamente pattern e caratteristiche spaziali. Le CNN applicano filtri convoluzionali su piccoli gruppi di pixel o regioni dei dati, catturando caratteristiche locali come bordi, forme e texture.

I filtri convoluzionali fungono da rilevatori di caratteristiche, ognuno impara una caratteristica della griglia di dati quando vede un certo pattern. Durante l'addestramento la rete aggiusta i numeri del filtro per renderlo utile al compito, ad esempio distinguere gatti e cani. Un filtro può diventare un rilevatore di

²⁵ https://en.wikipedia.org/wiki/Backpropagation_through_time

²⁶ https://en.wikipedia.org/wiki/Long_short-term_memory

²⁷ https://en.wikipedia.org/wiki/Gated_recurrent_unit

bordi verticali. Un altro può riconoscere bordi orizzontali. Altri ancora catturano texture o motivi ripetitivi. Man mano che i dati passano attraverso più strati convoluzionali, la rete impara a riconoscere pattern più complessi e astratti, come oggetti o volti (Figura 21).

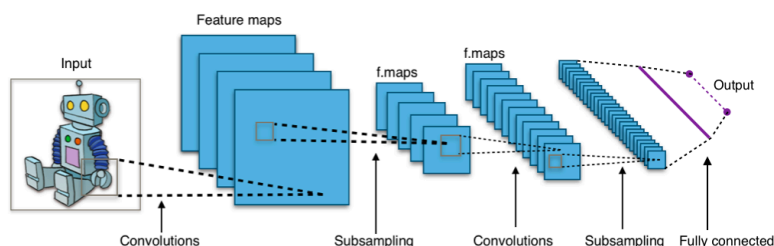


Figura 21: Schema di un'architettura CNN

Le CNN sono particolarmente adottate in scenari che richiedono riconoscimento e classificazione di immagini, come la visione artificiale, la diagnosi medica tramite analisi di immagini, la videosorveglianza, il riconoscimento facciale e la guida autonoma. La capacità di ridurre la dimensionalità mantenendo le informazioni rilevanti rende le CNN efficienti e ideali per questi compiti visivi complessi.

5. DENTRO LA SCATOLA NERA

La crescente complessità e la maggiore capacità delle reti neurali profonde hanno favorito lo sviluppo di sistemi di intelligenza artificiale molto efficaci in compiti come il riconoscimento di immagini, l'elaborazione del linguaggio naturale e la diagnosi medica. Tuttavia, proprio questa potenza computazionale, insieme alla loro architettura a strati, ha reso molte reti neurali (in particolare quelle profonde) dei veri modelli “a scatola nera” (black box). In questi casi, il percorso che porta a una decisione è talmente intricato che per un essere umano diventa difficile, se non addirittura impossibile, ricostruire con precisione le motivazioni che stanno dietro a una previsione o a una classificazione. Questa mancanza di trasparenza crea difficoltà concrete in settori in cui spiegabilità, verificabilità e giustificabilità sono requisiti fondamentali, come la medicina, la finanza o il diritto, dove le decisioni devono poter essere motivate e controllate.

Le reti profonde apprendono infatti in modo gerarchico: gli strati iniziali individuano caratteristiche semplici o di basso livello (per esempio bordi e texture in un'immagine), mentre gli strati successivi combinano progressivamente queste informazioni per riconoscere pattern più complessi, come forme e oggetti. Questa catena di trasformazioni è estremamente potente per la classificazione, ma rende difficile,

per un osservatore umano, capire quale caratteristica o quale combinazione di caratteristiche abbia determinato una specifica previsione. Inoltre, l'ottimizzazione dei parametri lungo i diversi strati può coinvolgere milioni o addirittura miliardi di parametri, producendo decisioni molto accurate ma poco interpretabili. Proprio l'assenza di una logica immediatamente comprensibile nei processi di apprendimento ha alimentato dubbi sulla fiducia e sull'affidabilità di questi sistemi.

Nell'articolo *Why Should I Trust You? Explaining the Predictions of Any Classifier*²⁸ dei ricercatori Ribeiro, Singh e Guestrin viene mostrato un caso "lupo vs husky" in cui un classificatore sembra funzionare bene, ma in realtà ha imparato una scorciatoia sbagliata. Durante l'addestramento le foto dei lupi avevano quasi sempre la neve sullo sfondo, mentre quelle degli husky no. Di conseguenza, il modello finisce per associare "neve" con "lupo" e "assenza di neve" con "husky", indipendentemente dalle caratteristiche dell'animale. Quando poi vede un husky in un paesaggio innevato, lo classifica come lupo e un lupo senza neve può essere classificato come husky. Nell'articolo vengono evidenziate le regioni dell'immagine che hanno contribuito alla decisione, mostrando che il modello sta guardando soprattutto lo sfondo chiaro/neve invece del corpo dell'animale.

È da qui che nasce l'esigenza di progettare modelli di IA che siano il più possibile spiegabili con tecniche di Explainable AI (XAI). In un mondo in cui decisioni automatizzate incidono su aspetti cruciali della vita quotidiana, diventa indispensabile poter comprendere e giustificare le decisioni prese dai modelli di intelligenza artificiale.

Esempi di XAI si trovano soprattutto in ambiti critici come medicina e finanza, dove spiegare il processo decisionale di una rete neurale è essenziale per garantire trasparenza e affidabilità. Nei sistemi di diagnosi assistita da IA, ad esempio, la XAI può aiutare i medici a capire quali caratteristiche cliniche abbiano guidato la diagnosi o una raccomandazione, offrendo quindi un supporto interpretativo che rafforza la validità della decisione

²⁸ <https://arxiv.org/pdf/1602.04938>

clinica. In modo analogo, nel settore finanziario, le tecniche di XAI vengono impiegate per rendere più chiare le decisioni di modelli che stimano il rischio o supportano la concessione di crediti, così che le previsioni risultino motivate e comprensibili anche per gli analisti.

Tra le tecniche più diffuse in XAI rientrano i metodi di attribuzione e di interpretazione locale, come LIME (Local Interpretable Model-agnostic Explanations) e SHAP (SHapley Additive exPlanations). LIME costruisce spiegazioni locali creando, vicino al singolo punto dati, una versione semplificata del modello: in questo modo l'utente può osservare l'effetto specifico delle diverse caratteristiche in quel particolare contesto. SHAP, invece, si basa sui valori di Shapley della teoria dei giochi²⁹ e assegna a ciascuna caratteristica dell'input un contributo per ogni previsione, fornendo quindi una misura quantitativa dell'importanza di ogni variabile nella decisione finale.

In Figura 22 è mostrato un tipico output di un'analisi SHAP applicata a un modello che predice la probabilità che una persona guadagni più di 50'000 \$ l'anno, utilizzando alcune caratteristiche (features) come età, relazione sentimentale, titolo di studio, numero di ore lavorate a settimana, stato civile, sesso, razza, paese di origine, ecc. I dati impiegati per addestrare il modello provengono dall'UCI Adult income dataset³⁰.

²⁹ La teoria dei giochi studia le decisioni strategiche tra individui o gruppi in contesti di interazione, in cui il risultato per ciascun partecipante dipende non solo dalle proprie scelte ma anche da quelle degli altri.

³⁰ <https://archive.ics.uci.edu/dataset/2/adult>

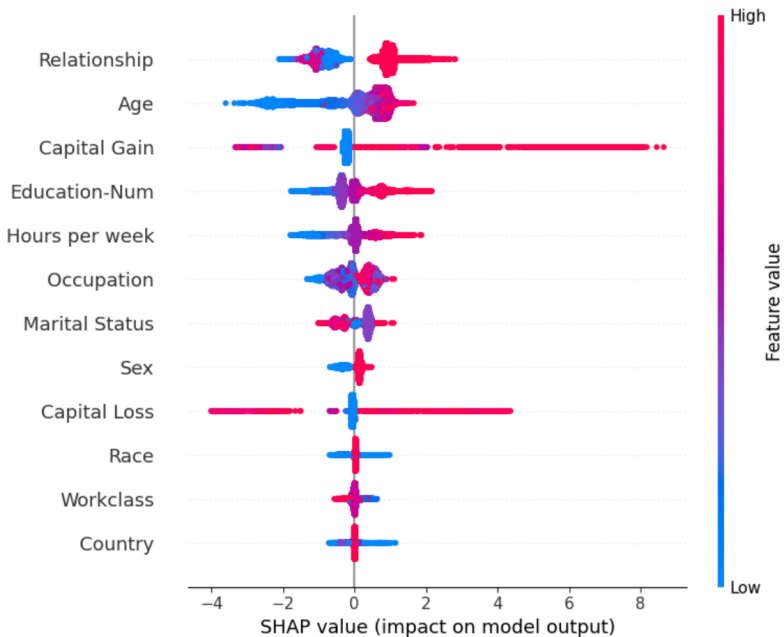


Figura 22: Un esempio di summary plot realizzato con l'analisi SHAP

La caratteristica più influente per questo modello predittivo risulta essere la relazione sentimentale (Relationship). Al secondo posto compare l'età (Age). La terza caratteristica, legata al guadagno netto (Capital gain), ha un impatto su tutto il modello, come ci si aspetterebbe, salvo che per una piccola parte, risulta meno rilevante dell'età e della relazione sentimentale.

Un'ulteriore tecnica molto utilizzata è la visualizzazione delle attivazioni, particolarmente comune nelle reti neurali convoluzionali (CNN) impiegate nella visione artificiale. Questo approccio mette in evidenza le regioni dell'immagine che hanno contribuito maggiormente alla classificazione, spesso tramite mappe di calore o saliency maps (Figura 23).

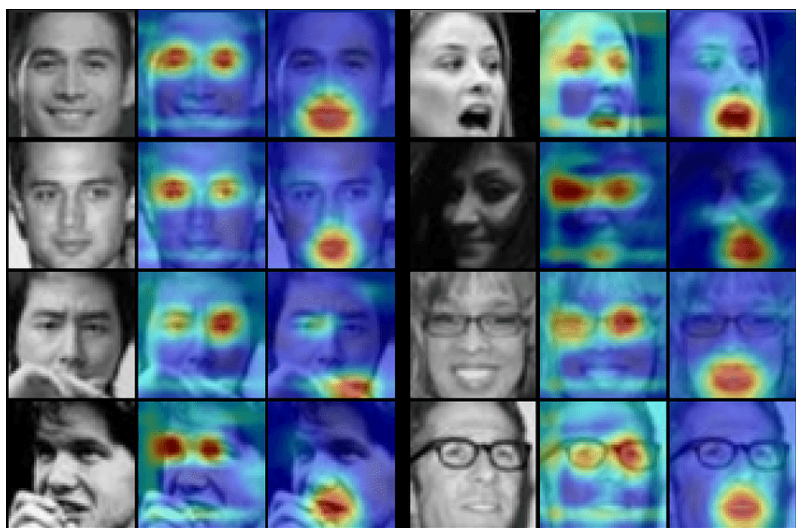


Figura 23: Attention map che si attivano per il riconoscimento degli occhi (a sinistra) e della bocca (a destra) in alcune fotografie

Per esempio, in un sistema che distingue tra polmoni sani e polmoni con segni di polmonite su una radiografia, la mappa può mostrare quali aree del torace hanno guidato la decisione; in un modello che riconosce pedoni e segnali stradali per la guida assistita, può evidenziare se l'attenzione si è concentrata davvero sul soggetto corretto; e in un'app che identifica specie di animali o piante, può indicare se il modello si è concentrato su dettagli rilevanti come il contorno delle foglie o il pattern del mantello. Queste visualizzazioni sono utili anche per scoprire errori o scorciatoie indesiderate: ad esempio, se un classificatore di immagini di cani e gatti si basa soprattutto sullo sfondo (come il divano o l'erba) invece che sull'animale, la saliency map può rivelarlo, aiutando a migliorare i dati e a rendere il modello più affidabile.

Vedremo che in realtà le tecniche di XAI hanno parecchi limiti quando la complessità del modello cresce enormemente. Nel Capitolo 6 introdurremo i Large Language Model e vedremo che con questi modelli le tecniche attuali di XAI riescono a spiegare ben poco del loro funzionamento.

L'attacco one-pixel

L'attacco one-pixel³¹ è un esempio molto noto di attacco basato su tecniche di XAI: modificare in modo impercettibile l'input (per esempio un'immagine) per far sbagliare un modello di machine learning. La particolarità, quasi scioccante dal punto di vista intuitivo, è che qui basta cambiare un solo pixel, cioè scegliere una posizione nell'immagine e alterarne il colore, per spingere la rete neurale a una classificazione errata, talvolta con alta confidenza.

Nel lavoro che ha reso popolare questo attacco, l'obiettivo è trovare quale pixel cambiare e come cambiarlo, facendo solo domande al modello (un'impostazione spesso definita black-box, perché non serve conoscere i pesi interni della rete). Per cercare la modifica “migliore”, gli autori usano un metodo di ottimizzazione ispirato all'evoluzione biologica, la differential evolution: si provano molte possibili modifiche di un pixel, si tengono quelle che aumentano la probabilità della classe desiderata (attacco targeted) o che riducono la classe corretta (attacco non-targeted), e si iterano i tentativi finché l'errore compare (Figura 24).

Un aspetto particolarmente interessante di questi attacchi è che non è necessario conoscere i dettagli interni del modello, come l'architettura, i pesi o il processo di addestramento. È sufficiente poter interrogare la rete neurale osservando le etichette che restituisce per determinati input.

L'idea è relativamente semplice: si inviano al modello diversi input (anche sintetici) e si osservano le classificazioni prodotte. Utilizzando queste coppie input-output è possibile addestrare un modello locale che approssima il comportamento della rete neurale originale. Una volta ottenuta questa approssimazione, il modello locale può essere utilizzato per costruire esempi avversari nello spazio degli input, che con alta probabilità inganneranno anche il modello originale.

³¹ Jiawei Su, Danilo Vasconcellos Vargas, Kouichi Sakurai, *One Pixel Attack for Fooling Deep Neural Networks*, 2019, <https://arxiv.org/pdf/1710.08864>

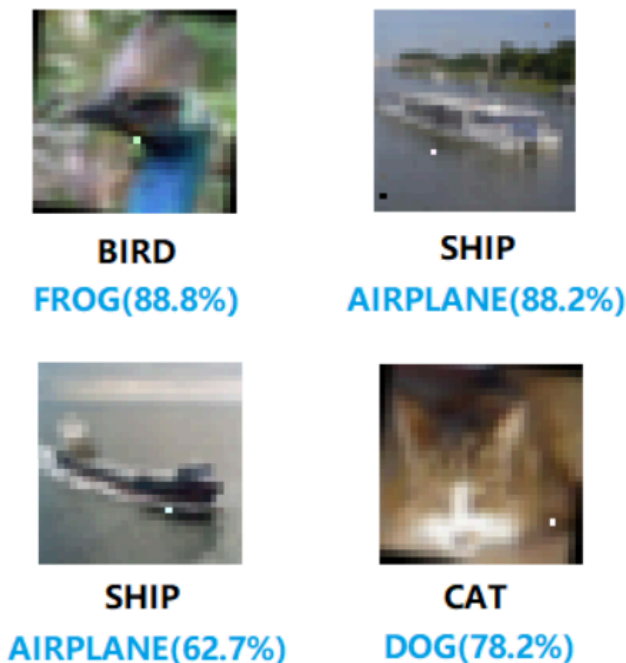


Figura 24: Errori generati con l'attacco one-pixel

In Figura 24 sono riportati alcuni esempi estrapolati dall'articolo originale dell'attacco one-pixel dove variando il semplice colore di un pixel illustrato nelle immagini si provoca l'errore del sistema di machine learning. Ad esempio, nella prima immagine in alto a sinistra un modello di machine learning riconosce correttamente un uccello (bird). La modifica di un singolo pixel, evidenziato in bianco nell'immagine, porta il modello a riconoscere erroneamente il contenuto come una rana (frog) con un grado di affidabilità dell'88,8%.

In un primo momento, l'attacco one-pixel potrebbe sembrare un tema di sicurezza più che di spiegabilità. In realtà è molto utile per la Explainable AI (XAI) perché mette in luce la fragilità del ragionamento del modello. Se un singolo pixel può cambiare la decisione, significa che la frontiera di decisione della rete può essere estremamente sensibile. Nel lavoro originale, gli autori mostrano che su CIFAR-10 una grande frazione di immagini

può essere forzata verso almeno una classe bersaglio con una sola modifica, e che la vulnerabilità si osserva anche su ImageNet (con percentuali più basse, ma comunque non trascurabili).

Inoltre, l'attacco si presta bene a essere letto con strumenti XAI. Con saliency map, activation maps o altre visualizzazioni, si può capire dove il modello “guarda” prima e dopo l'attacco: spesso emerge che la rete attribuisce un peso sproporzionato a dettagli che per un umano sono irrilevanti. L'attacco one-pixel attack può essere considerato un test di stress. Ricorda che spiegare un modello non significa soltanto raccontare perché una predizione è avvenuta, ma anche capire quanto quella predizione sia stabile e affidabile.

Questo tipo di approccio evidenzia anche un altro punto fondamentale per lo studio dell'Explainable AI: anche osservando il modello solo dall'esterno è possibile inferire molto sul suo comportamento. Comprendere quali caratteristiche guidano le decisioni di un modello diventa quindi cruciale sia per migliorarne l'affidabilità sia per individuare vulnerabilità e possibili attacchi.

Inoltre, la letteratura più recente³² mette in guardia dal fatto che non solo i modelli, ma anche le spiegazioni possono essere manipolate: un motivo in più per considerare XAI e robustezza come due facce della stessa medaglia.

³² M. Noppel and C. Wressnegger, *SoK: Explainable Machine Learning in Adversarial Environments*, 2024 IEEE Symposium on Security and Privacy (SP), <https://ieeexplore.ieee.org/document/10646794>

6. LINGUAGGIO ARTIFICIALE

Il linguaggio è una di quelle capacità umane che usiamo in modo così naturale da dimenticare quanto sia sofisticata. Parliamo, ascoltiamo, leggiamo e scriviamo dando per scontato che le parole “portino” il significato, e che il significato si trasferisca da una mente senza nessuna sbavatura. Il linguaggio è una delle espressioni più alte della nostra intelligenza. E’ ancora più sbalorditivo il fatto che nel mondo ci siano attualmente più di 8’000 lingue parlate, quindi tantissimi modi diversi di rappresentare e comunicare concetti anche molto complessi. Eppure, appena proviamo a spiegare a un computer il significato di una frase anche semplice, ci accorgiamo che il linguaggio non è soltanto un insieme di regole grammaticali. È un intreccio di contesto, intenzione, conoscenza del mondo, sottintesi, abitudini culturali e, soprattutto, di ambiguità gestite con disinvoltura.

Basta poco per rendersene conto. Una frase come “Ho visto l’uomo con il cannocchiale” è perfettamente corretta, ma può significare almeno due cose diverse: ero io ad avere il cannocchiale, oppure era l’uomo ad averlo. In una conversazione reale, spesso non ci accorgiamo nemmeno dell’ambiguità, perché il contesto la scioglie al volo: sappiamo dove siamo, che cosa stiamo facendo, che cosa è stato detto prima. Allo stesso modo, espressioni che sembrano banali diventano misteriose se isolate. Dire “È qui” o “È quello” non

basta: occorre sapere a cosa ci si riferisce, dove si trovano gli interlocutori, che cosa stanno guardando. E poi c'è la pragmatica, la parte del linguaggio che riguarda ciò che intendiamo davvero quando parliamo. “Puoi aprire la finestra?” sembra una domanda sulle capacità fisiche dell'interlocutore (ossia sei in grado di aprire una finestra), ma quasi sempre è una richiesta gentile. La frase non cambia, cambia l'intenzione, e l'intenzione la capiamo perché siamo esseri sociali immersi in situazioni condivise.

Il linguaggio, inoltre, è pieno di riferimenti incrociati. In “Maria ha parlato con Giulia perché era preoccupata”, a chi si riferisce quel “era preoccupata”? Maria o Giulia? Anche qui, spesso rispondiamo senza sforzo perché leggiamo tra le righe: immaginiamo scenari plausibili, ricostruiamo motivazioni, attiviamo la nostra conoscenza del mondo. È la stessa conoscenza che ci fa percepire come strana una frase tipo “Ho messo il gelato in tasca”: non c'è nulla di grammaticalmente scorretto, ma sappiamo che il gelato si scioglie e che la tasca non è il posto giusto. Il significato, quindi, non sta solo nelle parole: sta nel legame tra parole e realtà che in ambito tecnico si chiama spesso *grounding*³³.

Questa complessità spiega perché riprodurre il linguaggio con l'intelligenza artificiale sia stato, per decenni, un obiettivo tanto affascinante quanto difficile. La prima grande stagione di ricerca, quella che oggi associamo all'intelligenza artificiale simbolica, partiva da un'idea apparentemente ragionevole: se il linguaggio è governato da regole e strutture, allora basterà rappresentare queste regole in modo esplicito e farle applicare a una macchina. In questa visione, comprendere una frase significava analizzare la sintassi, costruire una rappresentazione formale del significato e poi manipolare simboli seguendo procedure logiche. Nascono così *parser* sintattici, alberi grammaticali, regole di produzione, sistemi che tentano di trasformare frasi in strutture logiche del tipo “se A allora B”. La promessa era seducente: un'intelligenza fatta di trasparenza,

³³ Il *grounding* o messa a terra, è il legame tra linguaggio e evidenze o percezioni esterne, fondamentale per una corretta comprensione del significato di una frase.

spiegabile passo dopo passo, perché ogni passaggio era una regola scritta da esseri umani.

In alcuni casi, i risultati furono degni di nota, come il sistema ELIZA già introdotto nel Capitolo 2.

Un altro esempio, in un certo senso ancora più istruttivo, è SHRDLU³⁴ (1968). Qui non si trattava solo di imitare un dialogo, il sistema operava in un piccolo mondo virtuale di blocchi colorati, con oggetti e regole semplici, e riusciva a collegare istruzioni linguistiche ad azioni e stati del mondo. In un ambiente ristretto, dove il numero di cose possibili è limitato e ben definito, il linguaggio diventa addomesticabile: il sistema può sapere davvero a cosa si riferiscono le parole, perché l'universo di riferimento è chiuso e controllato. SHRDLU mette in luce un fatto cruciale: la comprensione linguistica non è separabile dal mondo. Funziona benissimo quando il mondo è semplice e quando la conoscenza necessaria può essere enumerata. Il problema è che il mondo reale non è così.

Quando si prova a portare l'approccio simbolico fuori da domini ristretti, emergono limiti strutturali. Il primo è la fragilità, basta cambiare leggermente scenario perché regole scritte per un contesto falliscono in un altro. Il secondo è il costo della conoscenza, descrivere a mano tutte le eccezioni, le sfumature, i significati, i casi borderline, richiede uno sforzo enorme, spesso senza fine. Il terzo è la gestione dell'ambiguità e del buon senso, scegliere l'interpretazione giusta non è solo questione di grammatica, ma di probabilità, di contesto e di aspettative sul mondo. In pratica, l'IA simbolica aveva ragione su un punto, il linguaggio ha struttura, ma sottovalutava quanto quella struttura dipenda dall'esperienza e dalla conoscenza implicita.

Large Language Model

Da qui si apre la strada a una transizione che cambia radicalmente la prospettiva. Invece di provare a scrivere esplicitamente tutte le regole del linguaggio, si comincia a

³⁴ <https://en.wikipedia.org/wiki/SHRDLU>

chiedere: e se una macchina potesse impararle osservando tanti esempi? È il passaggio verso approcci statistici e poi verso il machine learning moderno. Il linguaggio, visto così, non è solo un sistema di regole, ma anche un sistema di regolarità probabilistiche: certe parole tendono a seguire altre parole, certi costrutti compaiono in determinati contesti, certe associazioni semantiche sono più frequenti di altre. Modelli più semplici, come quelli basati su n-grammi³⁵, mostrano già un'idea fondamentale: prevedere "la prossima parola" può catturare molta struttura. Se leggo "Nel mezzo del cammin di...", è plausibile che la parola successiva, in un corpus adeguato, sia "nostra". Non perché ci sia una regola grammaticale che imponga "nostra", ma perché un'enorme quantità di esempi suggerisce che quella continuazione è molto probabile.

Questa intuizione, tuttavia, rimane limitata finché i modelli non diventano capaci di gestire contesti lunghi e relazioni complesse tra parti distanti di una frase. Ed è qui che entra in scena uno degli snodi più importanti degli ultimi anni: l'architettura dei Transformer e, con essa, l'avvento dei Large Language Models (LLM). Per raccontarlo in modo semplice, si può dire così: un LLM è un modello che ha imparato a produrre linguaggio assorbendo regolarità da una quantità immensa di testi. La sua abilità di base è predire quali parole siano più probabili come prossimi elementi di una sequenza. Detto così sembra quasi banale, ma la combinazione di architettura e scala rende questo meccanismo sorprendentemente potente. In questo capitolo parleremo in dettaglio dell'architettura Transformer e dei LLM. La branca dell'intelligenza artificiale che si occupa del linguaggio è il Natural Language Processing (NLP). Il suo scopo è consentire ai sistemi di comprendere, interpretare e generare testo in modo paragonabile a quello di una persona. Per riuscirci, l'NLP combina strumenti e conoscenze provenienti da più ambiti come linguistica, statistica e apprendimento automatico, così da affrontare la complessità e l'ambiguità tipiche del linguaggio naturale.

Negli ultimi anni, i modelli di apprendimento automatico di

³⁵ Un n-gramma è una sequenza di n parole. Ad esempio, "andiamo al cinema" è un trigramma e "andiamo al cinema con Luca" è un 5-gramma.

grandi dimensioni (LLM), hanno trasformato radicalmente l'NLP. Modelli come BERT³⁶ e GPT³⁷ vengono addestrati su quantità enormi di dati testuali e includono miliardi di parametri: è proprio questa scala che permette loro di cogliere sfumature linguistiche molto complesse e di produrre testo di alta qualità. Alla base di questi modelli c'è l'architettura dei Transformer, che ha introdotto il meccanismo di auto attenzione (self-attention). Grazie a esso, il modello può considerare in modo più efficace il contesto di ogni parola e gestire dipendenze a lunga distanza all'interno di una frase o di un testo.

L'architettura Transformer è considerata superiore alle reti neurali ricorrenti (RNN³⁸), i modelli precursori dei LLM. Il vantaggio principale è proprio la self-attention, che consente al modello di analizzare l'intera sequenza di testo in parallelo, attribuendo un'importanza diversa a ciascuna parola in base al contesto. Le RNN, invece, elaborano i dati in modo sequenziale: questa caratteristica limita la loro capacità di gestire dipendenze lunghe e rende l'addestramento più lento e meno efficiente (Figura 25).

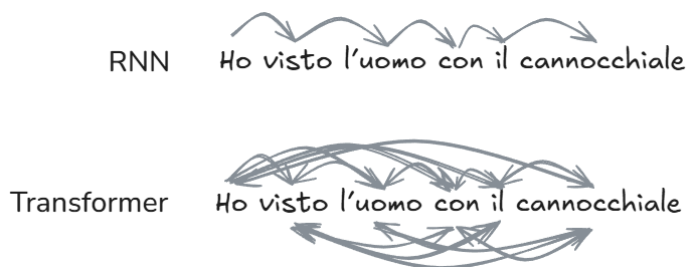


Figura 25: Differenza tra RNN e Transformer

In Figura, la frase “Ho visto l'uomo con il cannocchiale” viene elaborata dal modello RNN una parola alla volta; nel modello Transformer, invece, le parole vengono processate in parallelo

³⁶ Jacob Devlin et al., *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, Google, 2018, <https://arxiv.org/pdf/1810.04805>

³⁷ Alec Radford et al., *Improving Language Understanding by Generative Pre-Training*, OpenAI, 2018, <https://shorturl.at/0GqaG>

³⁸ https://it.wikipedia.org/wiki/Rete_neurale_ricorrente

considerando tutte le possibili combinazioni.

L'idea alla base dell'architettura Transformer può essere spiegata in modo intuitivo come un meccanismo che completa le frasi, selezionando una tra le parole più probabili da aggiungere. Per esempio, se fornissimo a un LLM addestrato sui testi di Dante Alighieri l'inizio "Nel mezzo del cammin di", il modello con alta probabilità proseguirebbe con il celebre verso iniziale dell'*Inferno*, cioè "Nel mezzo del cammin di nostra vita..." (Figura 26).

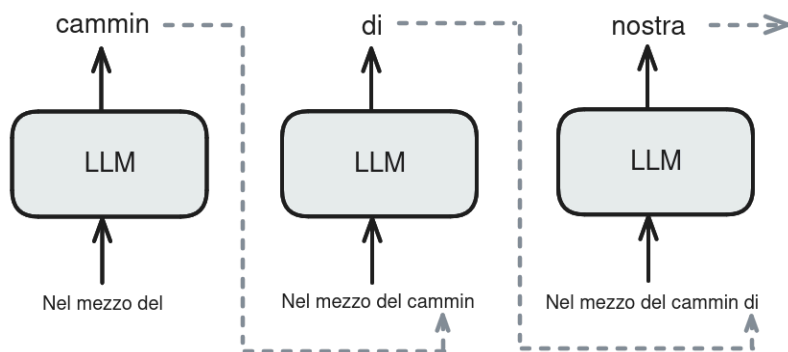


Figura 26: Completamento di una frase con un LLM

Naturalmente, non è garantito che l'LLM scelga esattamente il verso di Dante: la selezione della parola più probabile avviene attraverso un procedimento che include anche una componente di casualità. Al termine del processo dei Transformer, viene generato un elenco di parole candidate con probabilità elevate; da questo elenco si estrae un sottoinsieme, ad esempio le prime k (top- k), e si sceglie a caso una parola tra quelle selezionate. In questo modo, la stessa frase può essere completata in modi differenti: è uno dei motivi per cui si parla di intelligenza artificiale generativa (GenAI). La Figura 27 mostra un esempio concreto.

prob.	parola
0.20	vita
0.10	esistenza
0.02	sorte
0.01	giovinezza

← top-3

Figura 27: Scelta della parola successiva tra le prime 3 (top-3)

In questo caso, il modello restituisce l'elenco delle parole più probabili in ordine decrescente, partendo dalla più probabile (vita). Le prime tre parole (top-3) sono: vita, esistenza e sorte. Il modello può selezionarne casualmente una, producendo quindi frasi diverse a ogni iterazione: ad esempio potrebbe scegliere esistenza e generare “Nel mezzo del cammin di nostra esistenza...”, oppure scegliere sorte e produrre “Nel mezzo del cammin di nostra sorte...”, e così via.

I Large Language Model includono anche un parametro configurabile chiamato temperatura (temperature), che regola il livello di casualità nella scelta delle parole successive. Se la temperatura è impostata a zero, il modello seleziona sempre la parola con probabilità più alta e, a parità di domanda, fornisce sempre la stessa risposta. Aumentando la temperatura oltre lo zero, si introduce variabilità: più la temperatura è alta, maggiore è la probabilità che vengano scelte parole alternative, con risposte più diversificate. In un certo senso, la temperatura controlla il livello di creatività del LLM.

Transformer

L'architettura Transformer costituisce la base dei Large Language Model e rappresenta uno dei cambiamenti più importanti avvenuti negli ultimi anni nell'elaborazione del linguaggio naturale. Il modello originale fu introdotto nel 2017

nell'articolo *Attention Is All You Need*³⁹ pubblicato da ricercatori di Google. In quel contesto il Transformer era stato progettato principalmente per la traduzione automatica tra lingue diverse e utilizzava una struttura composta da due parti distinte, un encoder e un decoder (Figura 28).

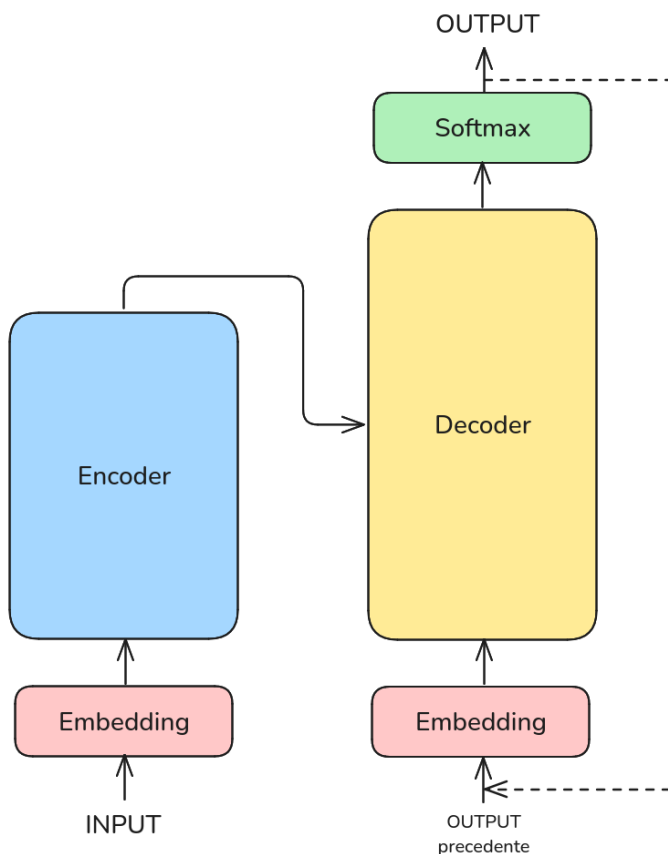


Figura 28: Schema generale di un Transformer

Nel modello originale l'encoder aveva il compito di leggere la frase di input e trasformarla in una rappresentazione numerica che catturasse il significato della sequenza. Il decoder riceveva

³⁹ Ashish Vaswani et al., *Attention Is All You Need*, 31st Conference on Neural Information Processing Systems (NIPS 2017), , Long Beach, CA, USA, <https://arxiv.org/pdf/1706.03762>

questa rappresentazione e generava la frase di output nella lingua di destinazione. Questo schema era adatto ai compiti di trasformazione di sequenze, come la traduzione o il riassunto di testi, ma non era pensato per la generazione libera di linguaggio. I moderni Large Language Model, come la serie GPT, utilizzano invece una variante semplificata dell'architettura chiamata *decoder-only Transformer*. In questo caso il modello elimina completamente il blocco encoder e utilizza soltanto il decoder, adattandolo per prevedere progressivamente il token successivo in una sequenza di testo. Questo cambiamento ha avuto un impatto fondamentale nello sviluppo degli LLM moderni, perché ha reso possibile addestrare modelli linguistici generali su grandi quantità di testo non strutturato e successivamente adattarli a diversi compiti.

Il funzionamento di un modello linguistico basato su Transformer inizia con il testo fornito dall'utente, chiamato **prompt**. Il prompt rappresenta il contesto iniziale da cui il modello deve generare una risposta. Poiché le reti neurali non possono elaborare direttamente parole o frasi in forma testuale, il primo passaggio consiste nella tokenizzazione del testo.

Un token è una parola o una parte di parola. I moderni LLM utilizzano diversi algoritmi di tokenizzazione a seconda della lingua e delle scelte progettuali del modello. Lingue con molte forme flesse, come l'italiano, oppure lingue prive di spazi tra le parole, come il cinese o il giapponese, richiedono strategie di tokenizzazione differenti. Cambiare tokenizzatore modifica il modo in cui il testo viene suddiviso in unità discrete e questo influisce direttamente sui costi computazionali, sulla qualità della rappresentazione linguistica e sul comportamento complessivo del modello.

Un esempio di algoritmo di tokenizzazione è il Byte Pair Encoding⁴⁰ (BPE), utilizzato ad esempio da GPT-2. Questo algoritmo parte dai caratteri elementari del testo e individua progressivamente le sequenze di simboli più frequenti, raggruppandole in unità più grandi. Le sequenze più comuni diventano quindi i token utilizzati dal modello.

⁴⁰ https://en.wikipedia.org/wiki/Byte-pair_encoding

In Figura 29 sono riportati due diversi esempi di tokenizzazione per la frase “Ho visto un uomo con il cannocchiale” utilizzando i modelli GPT-2 e GPT-4⁴¹.

Per GPT-2 i token sono 15 (utilizziamo le parentesi quadre per separare i token):

```
[Ho] [ v] [ist] [o] [ l] ['] [u] [omo] [ con]
[ il] [ cann] [oc] [ch] [ial] [e]
```

mentre per GPT-4 sono 11:

```
[Ho] [ visto] [ l] [ 'u] [omo] [ con] [ il]
[ cann] [oc] [ch] [iale]
```

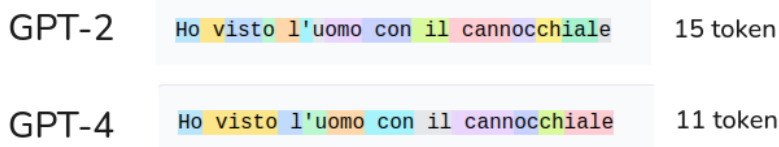


Figura 29: Esempio di tokenizzazione con GPT-2 e GPT-4

Alcuni modelli producono un numero maggiore di token spezzando le parole in frammenti più piccoli, mentre altri riescono a rappresentare unità linguistiche più complete. Differenze di questo tipo influenzano il numero di token necessari per rappresentare una frase e quindi anche la lunghezza del contesto che il modello riesce a gestire.

Dopo la tokenizzazione, ogni token viene trasformato in una rappresentazione numerica chiamata **embedding**. L'embedding è un vettore di numeri che codifica informazioni sul significato del token e sulle sue relazioni con gli altri termini del linguaggio. Parole che compaiono in contesti simili tendono ad avere rappresentazioni vettoriali vicine tra loro, permettendo al modello di catturare relazioni semantiche tra concetti diversi.

Per rappresentare correttamente una frase non è sufficiente conoscere il significato dei singoli token, è necessario anche

⁴¹ Potete verificare l'algoritmo di tokenizzazione (Tiktokenizer) utilizzato in vari modelli sul sito <https://tiktokenizer.vercel.app>

sapere in quale posizione si trovano nella sequenza. I Transformer infatti non leggono le parole una alla volta come facevano i modelli precedenti basati su reti ricorrenti (RNN), ma elaborano l'intera sequenza in parallelo. Per questo motivo viene aggiunta un'informazione chiamata *positional encoding*, che indica la posizione di ogni token nella frase.

Nel Transformer originale il positional encoding era definito tramite una trasformazione fissa e non appresa durante l'addestramento. Nei modelli più recenti, come GPT, la rappresentazione della posizione è invece appresa dal modello insieme agli altri parametri. In architetture ancora più recenti vengono utilizzati metodi come Rotary Positional Embedding⁴², che rappresentano la posizione in modo relativo e permettono di gestire contesti più lunghi con maggiore flessibilità.

Una volta ottenuti embedding e informazioni di posizione, i dati attraversano diversi strati della rete neurale basati sul meccanismo di attenzione. Questo meccanismo è il cuore dell'architettura Transformer e consente al modello di valutare quanto ogni token della sequenza sia rilevante rispetto agli altri.

Nel meccanismo di self-attention ogni token confronta la propria rappresentazione con quella di tutti gli altri token della sequenza. Il modello calcola quindi dei punteggi che indicano quanto ogni parola sia importante per comprendere il significato delle altre. In questo modo il sistema riesce a catturare dipendenze anche molto distanti all'interno della frase, superando i limiti dei modelli precedenti che avevano difficoltà a mantenere relazioni tra parole separate da molte posizioni.

Nei Transformer originali l'encoder utilizzava la self-attention per costruire una rappresentazione completa della frase di input. Il decoder riceveva questa rappresentazione e, durante la generazione dell'output, combinava due tipi di attenzione. Una attenzione guardava ai token già generati, l'altra consultava le informazioni prodotte dall'encoder.

Nei modelli decoder-only questo passaggio tra encoder e decoder non esiste più. L'intero processo avviene all'interno di una singola pila di blocchi di tipo decoder. Ogni blocco

⁴² Jianlin Su et al., *RoFormer: Enhanced Transformer with Rotary Position Embedding*, 2023, <https://arxiv.org/pdf/2104.09864>

contiene un meccanismo di self-attention e una rete feedforward. La self-attention è però di tipo causale, cioè ogni token può osservare soltanto i token che lo precedono nella sequenza. Questo vincolo impedisce al modello di “vedere il futuro” e rende possibile la generazione progressiva del testo.

Durante la generazione il modello riceve il prompt iniziale e calcola la probabilità dei possibili token successivi. Una volta scelto il token più appropriato, questo viene aggiunto alla sequenza e l'intero processo si ripete. Il modello continua quindi a estendere la frase un token alla volta, utilizzando ogni nuova parola come parte del contesto per la previsione successiva.

Il risultato finale di ogni passaggio è un vettore di valori che rappresenta la probabilità associata a ogni token del vocabolario. Per trasformare questi valori in probabilità viene utilizzata una funzione chiamata **softmax**. Questa funzione converte un insieme di numeri reali in una distribuzione di probabilità compresa tra 0 e 1, con somma totale pari a 1.

Il modello non produce necessariamente sempre il token con probabilità più alta. Durante la generazione può esplorare diverse alternative possibili scegliendo tra i token più probabili⁴³. Questo comportamento può essere controllato tramite alcuni parametri di generazione.

Un parametro interessante di questi modelli è la **temperatura**, che consente di regolare la variabilità nella scelta del token, ricalibrando le probabilità generate nella softmax. Questo consente di aumentare la probabilità che il modello scelga token diversi, generando quindi risposte differenti⁴⁴ (Figura 30).

⁴³ Come già spiegato con l'esempio top-3 di Figura 27.

⁴⁴ E' proprio grazie a questo che vengono chiamati modelli generativi.

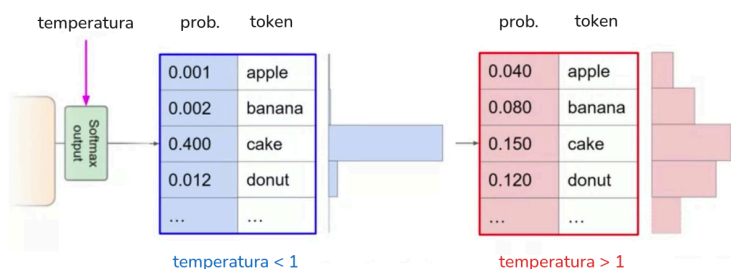


Figura 30: Relazione tra temperatura e distribuzione dei token

In Figura 30 abbiamo due esempi di temperature una “più fredda” con un valore minore di 1 e l’altra “più calda” con un valore maggiore di 1. A temperature basse il modello genera una distribuzione fortemente incentrata nella scelta del token “cake” (con probabilità 0,4), mentre a temperature più alte la distribuzione di probabilità è più uniforme, consentendo una maggiore diversificazione nella scelta del prossimo token, ad esempio “donut” (con probabilità 0,12), “banana” (con probabilità 0,08) o “apple” (con probabilità 0,04).

La temperatura regola di fatto la creatività nella generazione di una risposta, ma allo stesso tempo può anche aumentare la probabilità di allucinazioni. Proprio questa capacità di scegliere la parola successiva in modo probabilistico, basandosi su una rappresentazione profonda del contesto e delle relazioni linguistiche, rende i Transformer strumenti potenti sia per generare testo sia per comprendere in modo avanzato il linguaggio naturale.

Oltre alle differenze architetturali esiste anche una differenza fondamentale nel modo in cui questi modelli vengono addestrati. Il Transformer originale era pensato per essere addestrato direttamente su coppie input-output, come frasi in due lingue diverse. Non prevedeva un pre-training su grandi quantità di testo generico.

La svolta è arrivata nel 2018 con GPT-1⁴⁵, che ha introdotto il paradigma di pre-training su grandi collezioni di testo seguite da

⁴⁵Alec Radford et al., *Improving Language Understanding by Generative Pre-Training*, OpenAI, 2018, <https://openai.com/index/language-unsupervised/>

una fase di adattamento a compiti specifici. In questo approccio il modello impara prima le strutture generali del linguaggio leggendo enormi quantità di dati non annotati. Successivamente può essere specializzato attraverso diverse tecniche di addestramento aggiuntivo.

Nei moderni LLM questa fase di adattamento include tecniche come il supervised fine tuning, il reinforcement learning from human feedback e metodi più recenti come il direct preference optimization. Questi passaggi consentono di allineare il comportamento del modello alle aspettative degli utenti e di migliorare la qualità delle risposte.

L'evoluzione dall'architettura encoder-decoder del Transformer originale al modello decoder-only ha quindi segnato il passaggio dai sistemi progettati per compiti specifici ai grandi modelli linguistici generali. Grazie a questa trasformazione è stato possibile costruire sistemi capaci di comprendere e generare linguaggio naturale in modo estremamente flessibile, aprendo la strada allo sviluppo degli LLM utilizzati oggi.

La Figura 31 mostra uno schema dell'architettura Transformer con maggiori dettagli, inclusi i positional encoding citati in precedenza. Per un approfondimento sul funzionamento dell'architettura Transformer si consiglia la visione del video *L19.5.1 The Transformer Architecture*⁴⁶ di Sebastian Raschka, autore del libro *Building a Large Language Model (From Scratch)*⁴⁷.

⁴⁶ <https://www.youtube.com/watch?v=tstbZXNCfLY>

⁴⁷ <https://www.manning.com/books/build-a-large-language-model-from-scratch>

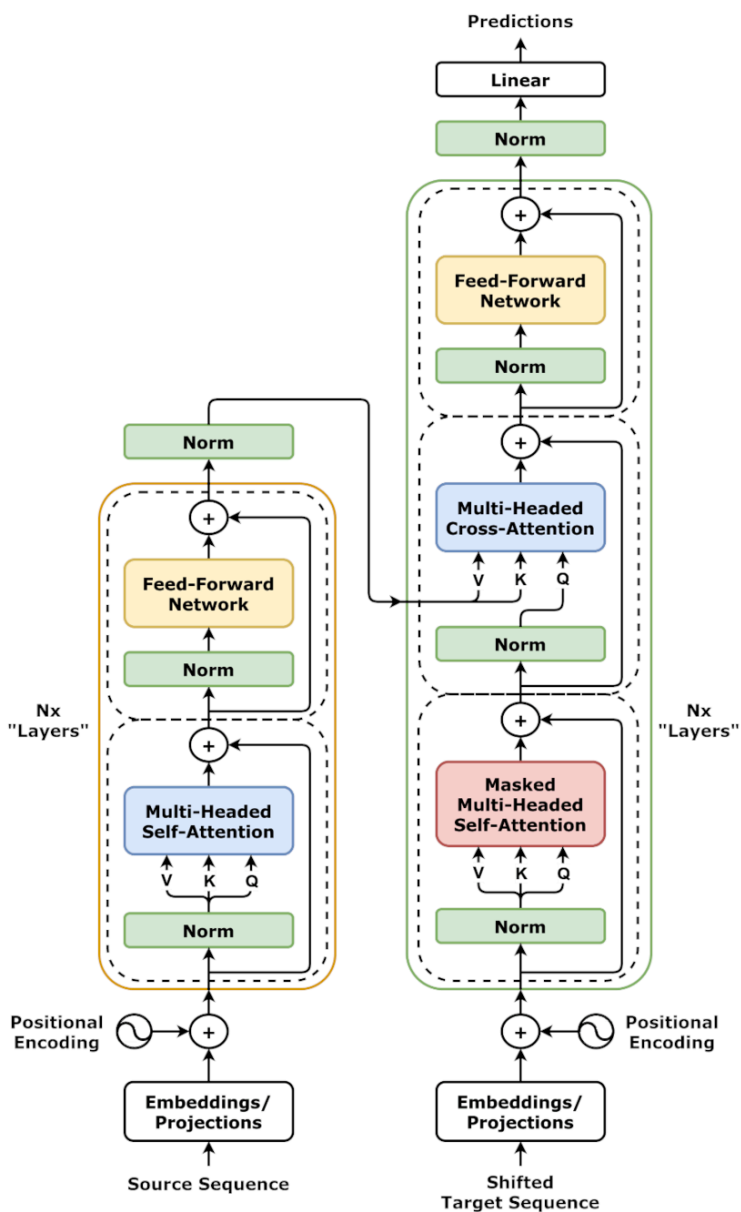


Figura 31: Schema dettagliato dell'architettura Transformer

Addestrare il modello

Nell'architettura Transformer le reti neurali sono presenti in quasi tutti gli step, nei componenti di Encoder e Decoder. Nello specifico nella sezione di Multi-Headed Self-Attention, Feed-Forward Network e Masked Multi-Headed Self-Attention (vedi Figura 31).

Per poter determinare i parametri (i pesi) di queste reti neurali è necessario addestrare il modello. Addestrare un grande modello linguistico (LLM) significa, in sostanza, trasformare una rete neurale inizialmente casuale (con numeri scelti a caso) in una macchina capace di assegnare alta probabilità alle continuazioni di testo più plausibili. Il cuore dell'idea è sorprendentemente semplice: si prende una quantità enorme di testi, li si spezza in unità chiamate token (pezzi di parole, punteggiatura, frammenti di codice), e si fa esercitare il modello su un compito di previsione. Ogni volta che sbaglia, si calcola di quanto ha sbagliato e si correggono leggermente i suoi parametri. Ripetuto miliardi di volte, questo processo porta a una competenza linguistica che non nasce da regole scritte a mano, ma da regolarità statistiche apprese dai dati.

Immagina il training come un gioco del testo incompleto, in cui al modello viene chiesto di ricostruire ciò che manca⁴⁸. Un esempio intuitivo, molto usato per spiegare il meccanismo, è quello in cui si prende una frase e si elimina una parola scelta a caso, poi si chiede al modello di predire la parola mancante. Per esempio: “La capitale dell'Italia è ____”. Se il modello propone “Roma” con alta confidenza, sta andando nella direzione giusta; se invece assegna più probabilità a risposte sbagliate, il sistema misura l'errore e aggiorna i pesi per rendere “Roma” un po' più probabile la prossima volta in un contesto simile. Questa dinamica di predizione, errore, aggiustamento dei pesi è la routine fondamentale dell'apprendimento. Va detto con

⁴⁸ In linguistica e soprattutto nella valutazione glottodidattica questo tipo di esercizio si chiama cloze test (o procedura cloze, spesso in italiano anche “test cloze” o “prova cloze”). È il classico esercizio in cui da un brano vengono eliminate alcune parole e il lettore deve indovinarle usando il contesto.

precisione che molti LLM generativi moderni (in stile GPT) non imparano principalmente togliendo parole “nel mezzo” a caso, ma prevedendo la continuazione da sinistra a destra: il principio però è lo stesso, perché in entrambi i casi il modello viene messo davanti a un pezzo di testo parziale e deve indovinare ciò che manca, e l'errore guida l'aggiornamento dei parametri.

Naturalmente, prima ancora di iniziare ad aggiornare i pesi, c'è un lavoro enorme sulla materia prima: i dati. Non basta scaricare petabyte⁴⁹ di documenti. I corpora vengono raccolti e soprattutto ripuliti: si rimuovono duplicati, spam, pagine piene di elementi ripetitivi, testi di bassa qualità, e spesso si applicano filtri che privilegiano contenuti informativi (manuali, spiegazioni, testi ben formati) rispetto a materiale confuso o ridondante. Questa fase conta moltissimo, perché la qualità media del testo influenza direttamente ciò che il modello interiorizza come normale: stile, coerenza, accuratezza, perfino il modo in cui argomenta.

Quando parte l'addestramento vero e proprio, il modello processa il testo in blocchi. A ogni passaggio produce una previsione (la parola successiva, o la parola mancante, a seconda dell'obiettivo), poi una funzione di perdita quantifica quanto quella previsione sia lontana dal bersaglio corretto. Da lì, tramite backpropagation, si calcola come modificare ciascun peso per ridurre quell'errore. Non è una correzione intelligente nel senso umano: è un aggiustamento numerico minuscolo, ripetuto in modo sistematico, che però su scala gigantesca produce capacità sorprendenti.

Dopo questa fase di pre-training, quella in cui il modello impara il linguaggio in senso ampio, come se avesse letto una biblioteca sterminata, spesso arriva il post-training, cioè l'addestramento che lo rende un assistente più utile e sicuro. L'obiettivo è trasformare il modello da “uno che completa bene i testi” a “uno che sa rispondere come una persona” con istruzioni, conversazione, buone maniere, rifiuti appropriati, attenzione al

⁴⁹ Un petabyte (PB) è l'unità di misura equivalente a un mille terabyte (TB), ossia a un milione di gigabyte (GB), ossia a un biliardo di byte. Un byte può essere considerato come un carattere.

contesto dell'utente. Tipicamente si usa un fine-tuning supervisionato su coppie di domanda/risposta di alta qualità utilizzando anche un feedback umano, tipicamente confrontando due risposte e indicando quale è migliore. In questo modo, il modello impara non solo a essere plausibile, ma anche più utile, chiaro e allineato allo stile desiderato.

Quanto devono essere grandi i dati per ottenere risultati notevoli dell'ordine di GPT-4? Qui è importante parlare per stime, perché i dettagli esatti dei modelli di frontiera non sempre sono pubblici. L'ordine di grandezza, però, è noto: si ragiona in termini di trilioni⁵⁰ di token. Per modelli di fascia altissima oggi è comune parlare di qualche trilione fino a oltre dieci trilioni di token di pre-training, con una componente significativa di dati selezionati (non solo quantità, ma qualità e varietà: prosa, enciclopedie, manualistica, codice, matematica, dialoghi). A questo si aggiungono dataset “curati” per il post-training che possono arrivare a milioni, talvolta decine di milioni, di esempi tra istruzioni e preferenze. E' fondamentale la scelta dei dati, e se essi sono bilanciati e deduplicati, e quanta potenza di calcolo e ottimizzazione si riesce a mettere in campo. A questo punto arriva la domanda cruciale: che cosa produce concretamente il training? Produce i parametri (i pesi) della rete neurale, cioè una collezione enorme di numeri che definiscono come il modello trasforma un input in un output. Un LLM, in pratica, è un grande insieme di numeri che codificano una certa competenza linguistica⁵¹. In Figura 32 sono riportati alcuni dei LLM più noti con il numero di parametri utilizzati, se divulgati dai relativi autori⁵².

⁵⁰ Un trilione è un miliardo di miliardi, in numero 1.000.000.000.000.000.000, ossia 1 seguito da 18 zeri.

⁵¹ In un certo senso un LLM è sistema che comprime i dati di training in una matrice di numeri. Una sorta di file ZIP della conoscenza umana.

⁵² Non sempre il numero di parametri è un dato rilevato pubblicamente dagli autori dei modelli, ad esempio non conosciamo il numero esatto di parametri per GPT-4 e GPT-5 ma solo una loro approssimazione.

Parametri	LLM
<10B	Mistral (7B), Llama 3.2 (3B)
10B-100B	Llama 3.3 (70B), DeepSeek R1 (70B), Claude 3 (70B)
100B-1T	Mistral 3 (675B), GPT-3 (175B), DeepSeek-V3 (671B), Claude 3 (175B)
1T-2T	GPT-4, Gemini 2
>2T	GPT-5, Claude Opus 4, Gemini 3 Pro

Figura 32: Dimensione dei parametri di alcuni LLM⁵³

Nell'architettura Transformer, questi pesi non risiedono in un unico posto: sono distribuiti nei blocchi che compongono la rete. All'ingresso troviamo i pesi degli embedding, la grande tabella che trasforma l'identità discreta di un token nel suo vettore continuo: è la porta d'ingresso del testo nel mondo numerico del modello. Poi c'è la pila di strati Transformer, ripetuti molte volte: in ogni strato ci sono i pesi della parte di attenzione (le trasformazioni che costruiscono le domande, le chiavi e i valori con cui i token si scambiano informazione) e i pesi della rete feed-forward, una piccola rete neurale applicata posizione per posizione che rielabora la rappresentazione dopo che il contesto è stato integrato tramite attenzione. In aggiunta, ogni strato contiene anche parametri di normalizzazione (blocchi Norm in Figura 30), pochi rispetto al resto ma importanti per la stabilità. Se si usa un Transformer encoder-decoder completo, nel decoder compare inoltre un ulteriore blocco di attenzione, la cross-attention, con i suoi pesi dedicati: è il meccanismo con cui il decoder consulta la

⁵³ La lettera B è l'unità di misura in billion, ossia miliardi in italiano, mentre la lettera T indica trillion, ossia bilione in italiano.

rappresentazione prodotta dall'encoder. Infine, all'uscita c'è la testa linguistica, una proiezione che trasforma lo stato interno finale in punteggi sul vocabolario per decidere quale token produrre dopo; anche questa è una grande matrice di pesi, spesso tra le più voluminose.

In altre parole, il training genera il modello nel senso più letterale possibile: non aggiunge regole esplicite, ma scolpisce, numero dopo numero, i parametri di embedding, attenzione, reti feed-forward, normalizzazioni e testa di uscita. È questa distribuzione di pesi dentro i blocchi Transformer che, a fine addestramento, rende il LLM capace di leggere un contesto, cogliere relazioni anche lontane nel testo e generare risposte coerenti e utili.

Finora abbiamo descritto il primo livello di training (detto anche pre-training), quello che consente di creare il modello di base, di fondazione. E' necessario eseguire un secondo passaggio di addestramento, detto anche fine-tuning che consente di migliorare lo stile linguistico del modello, rendendolo "più umano". Questo secondo training è spesso più oneroso perché non può essere automatizzato. E' necessario l'intervento dell'uomo, con esperti di dominio che siano in grado di scrivere esempi di domande e risposte corrette e che valutino successivamente la bontà delle risposte del modello tramite un sistema di votazione⁵⁴.

Senza questa fase di fine-tuning mirata, un LLM tende a produrre testo che può risultare sì grammaticalmente corretto, ma poco credibile nel senso pragmatico del termine: manca di intenzionalità comunicativa stabile, può essere incoerente nel seguire istruzioni, non calibra bene tono e contesto, e rischia di dare risposte eccessivamente generiche o, al contrario, troppo assertive senza adeguate cautele. Il pre-addestramento su grandi quantità di dati insegna al modello soprattutto a prevedere la prossima parola plausibile, non a comportarsi come un interlocutore: per questo, senza un allineamento tramite preferenze umane, il modello non ottimizza davvero qualità

⁵⁴ Quello appena descritto è il principio di funzionamento del Reinforcement Learning From Human Feedback (RLHF), una delle tecniche più utilizzate per eseguire il fine-tuning dei modelli.

conversazionale, utilità, sicurezza e rispetto delle norme implicite del dialogo, ad esempio ammettere incertezze, chiedere chiarimenti quando serve, evitare contenuti inappropriati, etc.

Le aziende come OpenAI, Google, Anthropic e Mistral, investono molto in questa fase di fine-tuning perché è questa che può diversificare realmente il comportamento di un modello e renderlo più adatto a un'interazione umana.

In un recente studio⁵⁵ si è evidenziato come questo aspetto di umanizzazione dei LLM può provocare fenomeni di servilismo (sycophancy) nei confronti degli utenti. Dei LLM analizzati nello studio ossia Claude, Gemini, GPT-4o, Llama e Mistral è emerso che le risposte adulatorie erano del 50% più numerose di quelle che si riscontrano normalmente fra persone.

Questo aspetto di quanto spingersi nell'imitazione dei comportamenti umani è uno dei temi più delicati e complicati da trattare. Il rischio è quello di creare modelli che provochino reazioni inaspettate nell'interazione con delle persone. Purtroppo, sono sempre più numerosi i casi di adolescenti che si sono fatti influenzare dalle conversazioni con LLM provocando in alcuni casi anche dei suicidi⁵⁶.

Large Reasoning Model

Negli ultimi anni i LLM sono diventati molto bravi a produrre testo scorrevole e plausibile. Tuttavia, appena un compito richiede più passaggi logici, un controllo dei vincoli o una pianificazione ordinata, spesso emergono i limiti della semplice previsione della parola successiva. Per superare questi limiti sono stati ideati i Large Reasoning Model (LRM), cioè modelli progettati e addestrati per dedicare più calcolo al ragionamento prima di dare una risposta finale.

L'idea di fondo è abbastanza intuitiva. Invece di rispondere subito, il modello costruisce una sequenza interna di passaggi

⁵⁵ Myra Cheng et al., *Social Sycophancy: A Broader Understanding of LLM Sycophancy*, 2025, <https://arxiv.org/pdf/2505.13995v1>

⁵⁶ Rob Kuznia, Allison Gordon e Ed Lavandera, *"You're not rushing. You're just ready:" Parents say ChatGPT encouraged son to kill himself*, 20 Nov. 2025, CNN, <https://edition.cnn.com/2025/11/06/us/openai-chatgpt-suicide-lawsuit-invs-vis>

intermedi e usa questi passaggi per verificare, correggere e completare la soluzione. In molti prodotti questa sequenza non viene mostrata integralmente, ma il comportamento si vede nei risultati, con risposte più strutturate quando la domanda richiede più passaggi.

Nel funzionamento pratico, un reasoning model esplora una sequenza di passaggi intermedi, che possono includere correzioni, ritorni sui propri passi, revisioni di una scelta fatta all'inizio. Questa fase non coincide necessariamente con ciò che l'utente vede. Nel caso di modelli come la famiglia OpenAI o1⁵⁷, l'azienda ha dichiarato di non mostrare all'utente la catena di ragionamento grezza, preferendo restituire un riassunto prodotto dal modello stesso, una scelta collegata sia al monitoraggio di sicurezza sia a considerazioni di prodotto. In molte implementazioni commerciali compaiono anche concetti come “reasoning tokens” e controlli di “reasoning effort”, cioè meccanismi che regolano quanta computazione dedicare alla fase di ragionamento.

Questa impostazione rende più naturale affrontare problemi in cui una singola risposta immediata tende a fallire. Un esempio intuitivo è la programmazione o la risoluzione di esercizi matematici, dove il modello deve mantenere vincoli, controllare casi limite, verificare se un passaggio precedente era corretto. In questi scenari, spendere più calcolo in inferenza può tradursi in un numero maggiore di tentativi, selezionando quello più coerente, oppure costruendo la soluzione con una logica più simile a quella di un essere umano che ragiona per passi.

Dal punto di vista dell'addestramento, i reasoning model partono dal pre training su larga scala tipico dei modelli di frontiera, poi si differenziano nella fase di post training, dove entrano in gioco tecniche di ottimizzazione orientate al ragionamento. Un fattore importante, descritto per o1, è l'uso di rinforzo su larga scala per insegnare al modello a utilizzare e migliorare una catena di ragionamento prima di rispondere. In questo quadro, due leve si rafforzano a vicenda, più rinforzo durante l'addestramento e più tempo di ragionamento in

⁵⁷ <https://openai.com/o1/>

inferenza, con miglioramenti che crescono in modo regolare man mano che si aumentano entrambe.

Un ingrediente particolarmente significativo è la cosiddetta *process supervision*, cioè un'idea che sposta l'attenzione dal solo risultato finale alla qualità dei passaggi intermedi. Invece di premiare soltanto la risposta corretta, il sistema assegna ricompense anche al percorso, valutando se ogni passo ha senso e conduce verso la soluzione.

La strada verso questi modelli non è stata un singolo salto, ma una somma di scoperte. Un punto di svolta, è stata la *chain of thought prompting*⁵⁸, che ha mostrato come una semplice richiesta di esplicitare i passaggi possa migliorare le prestazioni su problemi complessi, e persino una formula breve come “Let’s think step by step” può stimolare il ragionamento. Da lì sono arrivate estensioni importanti, per esempio la *self consistency*, che campiona molte catene di soluzione e poi sceglie la risposta più stabile, e metodi che combinano ragionamento e azione, come *ReAct*⁵⁹, in cui il modello alterna passi di pensiero e passi operativi.

Questa evoluzione si intreccia anche con una lezione più generale dell'intelligenza artificiale, la *bitter lesson* (lezione amara). E' un'idea formulata da Rich Sutton nel 2019⁶⁰, dopo decenni di storia dell'IA, e dice, in modo molto semplice, che nel lungo periodo vincono quasi sempre i metodi più generali che possono sfruttare sempre più potenza di calcolo, come la ricerca, e l'apprendimento dai dati, mentre gli approcci che cercano di incorporare nel sistema tanta conoscenza e tante regole scritte da esseri umani tendono a funzionare bene all'inizio, poi si fermano, e alla fine vengono superati. Nella ricostruzione proposta, alcuni tentativi di replicare capacità simili a o1 hanno evidenziato che tecniche relativamente

⁵⁸ Jason Wei et al., *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*, 36th Conference on Neural Information Processing Systems (NeurIPS 2022), <https://arxiv.org/pdf/2201.11903>

⁵⁹ Shunyu Yao et al., *ReAct: Synergizing Reasoning and Acting in Language Models*, International Conference on Learning Representations 2023, <https://arxiv.org/pdf/2210.03629>

⁶⁰ https://www.cs.utexas.edu/~eunsol/courses/data/bitter_lesson.pdf

semplici come la distillazione, cioè addestrare un modello più piccolo a imitare le uscite di uno più forte, possono funzionare sorprendentemente bene in questo contesto, rafforzando l'idea che la scalabilità, in addestramento e in inferenza, sia una parte decisiva della storia.

Il caso che ha reso popolare l'etichetta di reasoning model è la serie OpenAI o1, introdotta nel 2024 come modello progettato per spendere più tempo a pensare prima di rispondere, con un posizionamento esplicito su compiti complessi in matematica, codice e scienza. Possiamo citare anche modelli successivi come o3 sempre di OpenAI. Anche altre aziende hanno presentato modelli o varianti orientate al ragionamento, per esempio DeepSeek R1⁶¹, alcune versioni di Qwen⁶², e funzionalità di ricerca multi step come Deep Research⁶³ di Google in ecosistemi concorrenti.

⁶¹ <https://www.deepseek.com/en/>

⁶² <https://qwen.ai/home>

⁶³ <https://shorturl.at/2jf00>

7. IMITAZIONE O COMPRENSIONE?

Nel capitolo precedente abbiamo introdotto il meccanismo di funzionamento dei moderni LLM, in estrema sintesi questi modelli sono progettati per completare delle frasi, utilizzando delle reti neurali addestrate su milioni di documenti.

Emily Bender, una ricercatrice dell'Università di Washington (Seattle, USA) insieme ad altri colleghi, ha pubblicato nel 2021 un famoso articolo dal titolo *On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?*⁶⁴ dove viene data la seguente definizione:

Un modello linguistico è un sistema che, in modo casuale, cuce insieme sequenze di forme linguistiche che ha osservato nei suoi vasti dati di addestramento, sulla base di informazioni probabilistiche su come si combinano, ma senza alcun riferimento al significato: un pappagallo stocastico.

Questa definizione di “pappagallo stocastico” può sembrare molto riduttiva e infatti nella comunità scientifica ci sono pareri discordanti. Se è vero che un LLM è una macchina che predice solo il prossimo token come è possibile che sia in grado di fare il riassunto di un testo, di eseguire un ragionamento o di generare un programma per un computer? Queste attività non

⁶⁴ <https://s10251.pcdn.co/pdf/2021-bender-parrots.pdf>

prevedono una qualche forma di intelligenza? Se avete mai provato ad utilizzare uno di questi modelli come ChatGPT, Gemini o Claude, la sensazione che si prova è quella di dialogare con una forma di intelligenza, forse aliena come quella descritta dal Prof. Guido Vetere nel suo ultimo libro⁶⁵.

In questo capitolo proveremo a indagare su questo aspetto, introducendo alcune proprietà emergenti dei LLM e di come queste possano essere raffinate per imitare (o consolidare?) un comportamento intelligente.

Proprietà emergenti

In Fisica, una proprietà emergente è quando cambiamenti quantitativi in un sistema producono cambiamenti qualitativi nel comportamento. Questa è la definizione introdotta da Philip W. Anderson, premio Nobel per la Fisica nel 1977.

Un esempio di proprietà emergente è la superconduttività: quando in un metallo si abbassa la temperatura sotto una soglia critica, un cambiamento quantitativo nelle condizioni microscopiche (numero di elettroni che si organizzano coerentemente tramite le interazioni) porta a un cambiamento qualitativo del comportamento collettivo. Il sistema entra in una nuova fase in cui compaiono proprietà che non appartengono al singolo elettrone, come la resistenza elettrica nulla e l'effetto Meissner⁶⁶.

Queste proprietà emergenti sembrano essere presenti anche nei LLM. E' stato osservato⁶⁷ che modelli linguistici piccoli non hanno abilità che sono presenti nei modelli grandi della stessa famiglia, con una transizione apparentemente brusca al crescere della scala. In questa prospettiva, la sorpresa non è solo che i modelli grandi siano migliori, è che alcune curve di prestazione

⁶⁵ Guido Vetere, *Intelligenze aliene. Linguaggio e vita degli automi*. Luca Sossella editore, 2025, <https://lucasossellaeditore.it/libro/intelligenze-aliene/>

⁶⁶ L'effetto Meissner è un fenomeno della superconduttività: un superconduttore, raffreddato sotto la sua temperatura critica, espelle il campo magnetico e può respingere un magnete, permettendo levitazione magnetica in dimostrazioni con azoto liquido.

⁶⁷ Jason Wei et al., *Emergent Abilities of Large Language Models*, Transactions on Machine Learning Research (08/2022), <https://arxiv.org/pdf/2206.07682>

sembrano cambiare regime, passando da quasi casuale a chiaramente non casuale in un intervallo ristretto di scala (Figura 33).

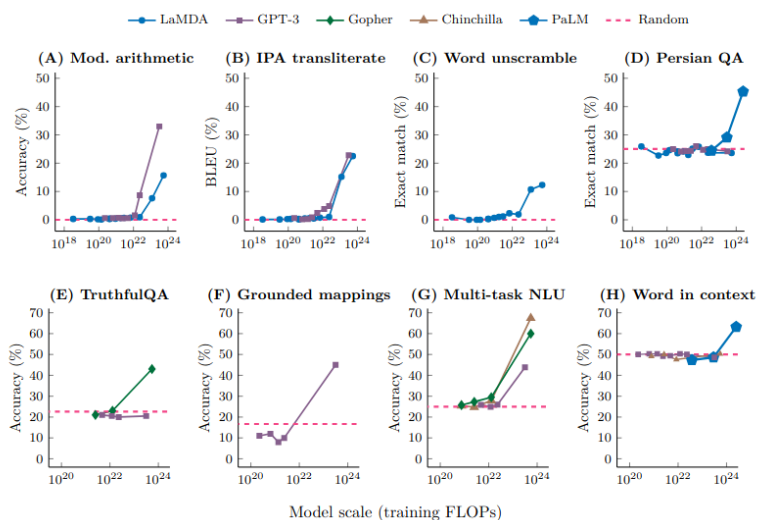


Figura 33: Abilità che emergono all'aumentare delle dimensioni

In Figura 33 è riportato il grafico originale dell'articolo di Jason Wei et al. dove si evince che i modelli manifestano un salto brusco nelle loro abilità dopo una certa soglia. Ad esempio, nel primo grafico in alto a sinistra relativo alle competenze aritmetiche di un modello, si nota che dopo la dimensione di 10^{22} FLOPs⁶⁸, i modelli GPT-3 e LaMDA⁶⁹ manifestano abilità di calcolo inaspettate.

Altri ricercatori, come Rylan Schaeffer, Brando Miranda, e Sanmi Koyejo dell'Università di Stanford hanno un'opinione diversa su queste proprietà emergenti. Secondo il loro studio⁷⁰, queste proprietà sono solo in realtà apparenti perché dipendono da come vengono misurate le prestazioni. In realtà gli errori del

⁶⁸ Un FLOP indica il numero di calcoli al secondo (in virgola mobile) utilizzati per addestrare il modello. Nota: 10^{22} equivale a diecimila miliardi di miliardi.

⁶⁹ <https://en.wikipedia.org/wiki/LaMDA>

⁷⁰ Schaeffer, Miranda, Koyejo, *Are Emergent Abilities of Large Language Models a Mirage?*, 2023, <https://arxiv.org/pdf/2304.15004>

modello possono diminuire in modo graduale man mano che il modello cresce, ma una metrica scelta a priori può far apparire il cambiamento come improvviso. Il tema delle proprietà emergenti è ancora dibattuto ma sta di fatto che questi LLM manifestano delle proprietà inattese, non previste a priori.

Ma quali sono queste proprietà dei LLM? Di seguito proviamo ad analizzare alcune⁷¹: in-context learning, summarization e tool-calling.

In-context learning

Con il termine in-context learning si indica la capacità di un LLM di adattarsi a un compito usando soltanto ciò che trova nel testo che gli forniamo (il prompt): istruzioni, esempi, formato, vincoli.

Questa capacità è estremamente utile perché trasforma il prompt in una specie di laboratorio: possiamo definire al volo una tassonomia, introdurre un'etichetta inventata, imporre un registro di scrittura, o descrivere una trasformazione quasi algoritmica. E allo stesso tempo è delicata: a volte il modello prende troppo alla lettera gli esempi, si confonde se sono ambigui, o imita il formato senza cogliere l'intenzione.

È importante chiarire che LLM non sta di fatto imparando, senso tradizionale del machine learning, perché i suoi parametri interni non vengono aggiornati. È un adattamento locale e temporaneo, come seguire una consegna dentro una conversazione.

Per capire davvero di cosa parliamo, conviene vedere l'in-context learning in azione.

Ad esempio, proviamo a scrivere un prompt per classificare alcune frasi, fornendo alcuni esempi.

Compito: assegna un'etichetta.

A significa: richiesta operativa (fare qualcosa)

B significa: domanda informativa (spiegare qualcosa)

Esempi:

⁷¹ L. Berti, F. Giorgi, G. Kasneci, *Emergent Abilities in Large Language Models: A Survey*, 2025, <https://arxiv.org/pdf/2503.05788>

Input: "Puoi fissare una riunione con Luca domani alle 10?"

Output: A

Input: "Perché il cielo è blu?"

Output: B

Input: "Scrivi una mail di ringraziamento al recruiter."

Output: A

Ora classifica:

Input: "Come funziona la crittografia a chiave pubblica?"

Output:

Quello che stiamo facendo qui è fornire degli esempi per classificare se una frase è una richiesta operativa (A) oppure se è una domanda informatica (B).

Il punto interessante è che A e B non portano significato di per sé. Il modello lo ricava dal contesto e poi lo applica a un caso nuovo.

In particolare, il prompt inizia definendo che cos'è il compito da svolgere, fornisce 3 esempi e infine chiede di classificare la frase "Scrivi una mail di ringraziamento al recruiter".

Inviando questo prompt ad un LLM la risposta che ci aspettiamo è ovviamente B.

Questo modo di scrivere la domanda al LLM segue delle regole particolari, simili più ad un linguaggio di programmazione che a un dialogo in linguaggio naturale.

Questa forma di scrittura è una delle varie tecniche di prompt engineering⁷², ossia l'arte di saper scrivere i prompt per ottenere la miglior risposta possibile da un LLM.

Fornire più esempi per guidare il modello verso la risposta desiderata è una tecnica definita *few-shot prompting*⁷³. Generalmente più gli esempi sono specifici e meglio il LLM riuscirà a fornire la risposta desiderata.

Questa proprietà dell'in-context learning non è stata progettata

⁷² Il prompt engineering è diventato un vero e proprio lavoro, il prompt engineer, ossia l'ingegnere dei prompt. Tra le varie risorse su internet potete consultare il sito <https://www.promptingguide.ai/it>

⁷³ Valentina Peter, Enrico Zimuel, *LLM e didattica: few-shot prompting*, ICT Ed Magazine, Anno VII, N° 3, October 2024, p. 10-14, ISSN 2611-4259

a priori, è emersa nei primi LLM di grandi dimensioni. Quasi tutti i LLM moderni sanno “imparare dal prompt” almeno un po’, ma quanto bene lo fanno dipende da scala e addestramento; e il fine-tuning spesso li rende più bravi a seguire istruzioni ed esempi.

Summarization

Una delle capacità che più colpiscono quando si usa un LLM è la facilità con cui riesce a realizzare riassunti di un testo lungo. Questa abilità, spesso chiamata summarization, non è soltanto un trucco di stile: è una forma di compressione semantica. Invece di limitarsi a tagliare frasi, il modello tenta di identificare i nuclei informativi, stabilire che cosa è centrale e che cosa è accessorio, e poi riscrivere il contenuto in modo più denso e scorrevole. Per questo viene spesso descritta come una proprietà emergente: in molti modelli cresce in modo evidente con la scala e con la qualità del training, fino a diventare un comportamento affidabile in un ampio numero di contesti.

Per capire perché la summarization emerge, conviene partire dal fatto che un LLM non viene addestrato, nella sua forma più semplice, con l’obiettivo esplicito di riassumere. Nel pre-training classico un modello viene esposto a enormi quantità di testo e impara a predire il token successivo. Sembra lontano dall’idea di riassumere, eppure in quel compito apparentemente semplice è nascosta una pressione enorme verso la comprensione delle regolarità del linguaggio. Per predire bene la prossima parola, il modello deve sviluppare rappresentazioni interne che tengano insieme riferimenti, cause ed effetti, ruoli dei personaggi, argomenti principali, definizioni, e anche strutture discorsive come introduzioni, esempi, obiezioni e conclusioni. In altre parole, per continuare coerentemente un testo il modello deve imparare a stimare che cosa conta davvero nel contesto e che cosa no⁷⁴. Questa selezione dell’informazione è già un ingrediente fondamentale del riassunto.

Dal punto di vista del meccanismo, la summarization nasce

⁷⁴ Questa stima dell’importanza delle parole nel contesto è la base del meccanismo della self-attention dell’architettura Transformer.

perché il modello è in grado di costruire, durante la lettura del prompt, una sorta di stato informativo distribuito: non una memoria fatta di frasi copiate, ma una rete di indizi e relazioni che collegano parti diverse del testo. L'attenzione, nelle architetture Transformer, facilita proprio questo: quando il modello genera una parola del riassunto, può collegarsi in modo differenziato a molte parti del testo originale, pesando più i passaggi rilevanti e meno quelli secondari. Non è una lettura umana, ovviamente, e non è un processo simbolico con regole esplicite; è un calcolo di compatibilità tra ciò che sta per scrivere e ciò che ha letto, che permette di richiamare dettagli quando servono e di ignorarli quando non servono. Se il prompt chiede un riassunto breve, il modello tende a privilegiare concetti ad alta copertura informativa, cioè quei punti che spiegano o sintetizzano molte frasi del testo di partenza.

C'è poi un secondo passaggio, spesso decisivo nell'esperienza quotidiana: il fine-tuning durante il post-training. Anche se il modello base può già mostrare capacità di summarization, il comportamento migliora molto quando il modello viene addestrato a rispondere a istruzioni del tipo “riassumi in tre frasi”, “mantieni un tono neutro”, “non aggiungere informazioni”. In questa fase il modello non impara il concetto di riassunto da zero, ma impara a riconoscere meglio l'intento dell'utente, a rispettare vincoli di lunghezza e formato, e a produrre un testo che appare più centrato e leggibile. È come se il pre-training costruisse la competenza linguistica di fondo e il post-training insegnasse quando e come applicarla in modo utile.

Un modo intuitivo per immaginare la summarization in un LLM è questo: quando viene chiesto un riassunto, si sta chiedendo al modello di produrre una nuova sequenza di parole che abbia alta probabilità di essere un buon riassunto dato quel testo. Buon riassunto qui significa una combinazione di fattori che il modello ha interiorizzato dai dati: coerenza, completezza rispetto ai punti principali, brevità, e spesso anche un certo stile espositivo. Durante la generazione il modello non seleziona semplicemente frasi da copiare; può farlo, ma spesso costruisce un testo astrattivo, cioè riformula. Questa riformulazione è

possibile perché il modello ha imparato molte equivalenze linguistiche e molte strutture retoriche tipiche dei riassunti, come l'apertura che dà il tema generale e la chiusura che evidenzia implicazioni o risultati. Se, per esempio, un articolo spiega prima il contesto, poi un problema, poi una soluzione e infine un risultato, il modello riconosce questo schema e tende a riprodurlo in miniatura.

Naturalmente, il fatto che un LLM sappia riassumere non significa che lo faccia sempre in modo fedele. La summarization è un compito che mette in tensione due obiettivi: essere breve e mantenere accuratezza. Quando si comprime informazione, si rischia di perdere dettagli importanti o di omettere sfumature che nel testo originale erano cruciali. Inoltre, poiché un LLM genera testo plausibile, può capitare che riempia vuoti con inferenze non richieste, oppure che introduca dettagli che suonano verosimili ma non sono presenti nel documento. Questo succede soprattutto quando il testo di partenza è ambiguo, quando il prompt non specifica che il riassunto deve essere strettamente ancorato alla fonte, o quando il modello non ha accesso a una porzione del testo per limiti di contesto o di attenzione effettiva. In pratica, un riassunto ben scritto non è detto che sia anche un riassunto fedele.

Per questo, se si vuole descrivere la summarization come proprietà emergente vale la pena sottolineare che essa riguarda la capacità di trasformare testo in testo con compressione semantica e buona forma linguistica, mentre la fedeltà fattuale è un obiettivo più delicato e dipende molto dal modello, dal prompt e dalla natura della fonte. Quando funziona, la summarization sembra quasi un superpotere: leggere pagine e restituire l'ossatura in poche righe. Quando inciampa, mostra il suo lato più fragile: semplifica troppo, interpreta, talvolta si convince di aver visto nel testo qualcosa che invece non c'era.

In definitiva, i LLM riescono a fare riassunti perché, per imparare a predire il linguaggio su scala enorme, hanno dovuto imparare anche a rappresentare ciò che nel linguaggio è più strutturale e più informativo. La summarization non è un modulo separato: è un uso particolare di competenze generali. È la prova che, quando un sistema impara a manipolare il testo

con sufficiente profondità, finisce per sviluppare anche una forma di compressione concettuale. Ed è proprio questa compressione, quando viene guidata da un'istruzione chiara, che dà origine a quel fenomeno che chiamiamo riassunto.

Tool calling

Nelle applicazioni reali, un modello linguistico non viene usato solo per generare del testo, ma per collegare richieste in linguaggio naturale a dati e azioni: interrogare un catalogo prodotti, consultare un calendario, cercare sul web, aggiornare una scheda cliente, avviare un flusso di lavoro. La chiamata di funzioni, oggi spesso chiamata tool calling (o function calling), è il meccanismo che rende questo passaggio praticabile: il modello produce una richiesta strutturata che indica quale funzione invocare e con quali argomenti, mentre l'esecuzione vera e propria avviene nel software che integra il modello. La centralità di questo schema è diventata evidente man mano che la ricerca e l'industria hanno spostato l'attenzione dalla generazione di testo verso sistemi capaci di agire in ambienti esterni.

Immaginiamo un assistente per il meteo integrato in un'app. L'utente scrive: "Che tempo fa a Roma oggi?". Il LLM, invece di rispondere, produce una richiesta di azione del tipo:

```
get_current_weather(location="Roma, IT")
```

Ossia il LLM riconosce di dover richiedere l'esecuzione di un tool esterno, suggerendo di far eseguire all'app la funzione `get_current_weather`, passando come località (location) la città di Roma in Italia.

L'applicazione riceve questa richiesta, esegue davvero la funzione interrogando un servizio meteo e ottiene un risultato, per esempio temperatura 19° e condizione "pioggia leggera". A quel punto l'app invia il risultato al modello, che lo trasforma in una frase naturale: "A Roma ci sono circa 19°C e piove leggermente". Il valore sta tutto nel contratto: il modello decide quale strumento usare e con quali parametri, l'app interagisce

con il servizio meteo, e il modello torna a fare ciò in cui è forte, cioè comunicare e contestualizzare⁷⁵.

Questa capacità, nata come proprietà emergente, è stata consolidata con fine-tuning mirati, supervisionati o con feedback umano. Un esempio storico molto chiaro è WebGPT⁷⁶, dove GPT-3 viene addestrato (fine-tuned) per usare un browser, imparando a cercare e consultare fonti su internet per sostenere le risposte.

Questa idea si perfeziona grazie ai sistemi modulari, come MRKL⁷⁷, che descrivono un'architettura in cui il modello linguistico funge da collante e instrada azioni che emergono dal prompt verso moduli esterni specializzati, ad esempio conoscenza, calcolo, ragionamento discreto. L'idea è che un LLM può diventare un controllore, capace di scegliere quale strumento (tool) esterno attivare per compensare i suoi limiti intrinseci.

Il passo successivo è rendere l'azione parte integrante della strategia di soluzione. Questo tema è formalizzato nel framework ReAct⁷⁸ che propone un approccio in cui ragionamento e azioni, verso strumenti esterni, si intrecciano: il modello alterna momenti di pianificazione e momenti di interazione con servizi esterni, riducendo allucinazioni e propagazione degli errori proprio perché può verificare, recuperare dati e aggiornare il piano mentre procede. E' una sorta di ragionamento forzato ispirato al nostro problem solving: scomporre il problema in azioni e pianificare un ordine di esecuzione. Il modello impara a riconoscere la necessità di consultare una fonte esterna evitando di inventare (allucinare)

⁷⁵ Il dettaglio del funzionamento dell'esempio del meteo con il tool calling è disponibile nell'articolo *Tools calling in Agentic AI. How LLMs power Agentic AI systems*, https://www.zimuel.it/blog/tool_calling_AI_agents

⁷⁶ Reiichiro Nakano et al., *WebGPT: Browser-assisted question-answering with human feedback*, OpenAI, 2022, <https://arxiv.org/pdf/2112.09332>

⁷⁷ Ehud Karpas et al., *MRKL Systems. A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning*, AI21 Labs, 2022, <https://arxiv.org/pdf/2205.00445>

⁷⁸ Shunyu Yao, *ReAct: Synergizing Reasoning and Acting in Language Models*, Negli atti della conferenza International Conference on Learning Representations, ICLR 2023, <https://arxiv.org/pdf/2210.03629>

una risposta.

Con il progetto Toolformer⁷⁹ è stata formalizzata la tecnica di addestramento dei modelli per decidere quali strumenti (tool) chiamare, quando farlo, quali argomenti passare e come incorporare il risultato, usando un'impostazione in larga parte auto-supervisionata con poche dimostrazioni per strumento. Il risultato è evidente, l'affidabilità del tool calling cresce quando il comportamento viene allenato esplicitamente, non soltanto richiesto via prompt.

OpenAI ha reso questa capacità un elemento strutturale dell'ecosistema quando, a giugno 2023, ha introdotto la funzione di function calling nella Chat Completions API⁸⁰: gli sviluppatori descrivono funzioni disponibili e il modello può scegliere di restituire gli argomenti necessari per invocarle in modo affidabile, con un formato pensato proprio per essere eseguito da un'applicazione.

Questa funzionalità del tool calling rappresenta la proprietà fondamentale sulla quale si basano tutte le applicazioni agentiche che stanno spopolando in questi ultimi anni. Ad esempio, se oggi ChatGPT è in grado di effettuare delle ricerche in tempo reale su internet è grazie al tool calling. Un altro esempio è il servizio Perplexity⁸¹ che è basato interamente sull'utilizzo di tool esterni per effettuare delle ricerche online su svariati argomenti.

Imitare più che comprendere?

Non c'è dubbio che i LLM abbiano raggiunto livelli sorprendenti. Basta dialogare con i recenti modelli come GPT-5, Gemini, Claude per rendersene conto. La sensazione netta è quella di interagire con una qualche forma di intelligenza. Abbiamo visto che in realtà questi modelli sono stati progettati con metodologie stocastiche, ossia in grado di

⁷⁹ Timo Schick et al., *Toolformer: Language Models Can Teach Themselves to Use Tools*, Meta AI, Universitat Pompeu Fabra, 2023, <https://arxiv.org/pdf/2302.04761>

⁸⁰ <https://platform.openai.com/docs/api-reference/chat>

⁸¹ <https://www.perplexity.ai/>

predire la parola successiva in base al contesto della frase, in base a un addestramento su miliardi di libri, articoli e materiale online scritto dall'uomo. In un certo senso è come se questi modelli avessero imparato a imitare la scrittura umana da tutte queste fonti. E' sorprendente che questi modelli funzionino così bene ed è difficile etichettare il comportamento come una mera elaborazione stocastica. Forse c'è di più.

Nel Capitolo 1 abbiamo parlato del Test di Turing. Nel Marzo 2025 questo test è stato superato dai modelli GPT-4.5 e LLaMa-3.1⁸². Questo vuol dire che l'intelligenza artificiale ha raggiunto l'intelligenza umana? In realtà, questo test non lo dimostra. Visto che non abbiamo una definizione precisa di cosa sia l'intelligenza, Turing nel 1950 ideò questo gioco per capire se un sistema artificiale potesse dialogare al pari di un umano. Se si comporta come un'anatra, nuota come un'anatra e starnazza come un'anatra, allora probabilmente è un'anatra⁸³.

Nell'ambito della filosofia della mente e della scienza cognitiva, il Prof. John Searle⁸⁴ ha invece criticato la possibilità che un'intelligenza artificiale possa realmente comprendere, sostenendo con il celebre esperimento mentale della "stanza cinese" (1980) che una macchina potrebbe simulare la comprensione senza possedere una vera intelligenza.

Immaginiamo una stanza chiusa, all'interno della quale si trova una persona che non conosce il cinese. Fuori dalla stanza, ci sono dei madrelingua cinesi che inseriscono delle domande scritte in cinese attraverso una fessura.

Dentro la stanza, la persona ha a disposizione un enorme manuale con istruzioni dettagliate su come manipolare i simboli cinesi. Questo manuale le dice esattamente quali caratteri rispondere in base ai caratteri ricevuti, senza che la persona abbia alcuna idea del significato dei simboli. Seguendo le istruzioni, la persona riesce a produrre risposte in cinese che sembrano del tutto corrette agli interlocutori esterni (Figura 34).

⁸² Cameron R. Jones, Benjamin K. Bergen, *Large Language Models Pass the Turing Test*, 2025, <https://arxiv.org/pdf/2503.23674>

⁸³ https://it.wikipedia.org/wiki/Test_dell'anatra

⁸⁴ [https://en.wikipedia.org/wiki/John_Searle_\(philosopher\)](https://en.wikipedia.org/wiki/John_Searle_(philosopher))

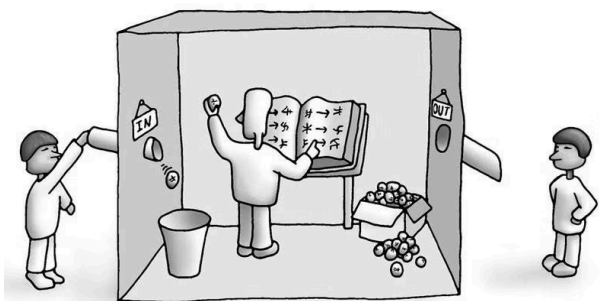


Figura 34: Schema dell'esperimento della stanza cinese di Searle

Da fuori, chi riceve le risposte potrebbe pensare che all'interno della stanza ci sia qualcuno che comprende realmente il cinese, quando in realtà la persona all'interno sta solo eseguendo regole formali, senza alcuna comprensione del linguaggio o del significato delle parole.

Secondo Searle, ciò che chiamiamo "intelligenza artificiale" potrebbe essere, nella migliore delle ipotesi, una sofisticata elaborazione simbolica priva di coscienza e intenzionalità.

Con questo esperimento, Searle vuole dimostrare che elaborare simboli secondo regole predefinite non equivale a comprendere realmente qualcosa. Questo è un attacco diretto alla concezione dell'Intelligenza Artificiale Forte, secondo cui una macchina potrebbe non solo simulare l'intelligenza umana, ma anche sviluppare una vera comprensione e coscienza.

Per Searle, un computer che esegue operazioni su dati (come la persona nella stanza che segue istruzioni senza capire) può imitare il comportamento umano, ma non sviluppa una vera comprensione, perché manca di intenzionalità, ovvero la capacità di attribuire significato ai simboli. L'IA, per quanto avanzata, resterebbe dunque un sistema di manipolazione sintattica, incapace di vera semantica.

Avendo introdotto l'architettura Transformer dei LLM nel Capitolo 6, sappiamo che utilizzano reti neurali profonde per imparare a comprendere gli schemi tra le parole. Questa architettura non può in qualche modo manifestare una forma di comprensione, più che di imitazione?

Uno studio del 2025 pubblicato da Anthropic⁸⁵ ha cercato di fare luce su alcuni aspetti dei meccanismi interni di funzionamento di Claude, il LLM realizzato dall'azienda. E' stata ideata una tecnica chiamata *circuit tracing* in grado di analizzare, tramite una rete di connessioni, i componenti dell'architettura che contribuiscono alla scelta di un token in output e in quale misura.

Nello studio sono stati osservati alcuni fenomeni interessanti:

- **Ragionamento a più passi:** quando il modello pensa a “Austin, Texas” sceglie “Texas” come passo intermedio prima di arrivare ad “Austin”.
- **Pianificazione in avanti nella scrittura creativa:** nella generazione di una poesia, il modello tende a preselezionare possibili rime prima di completare la riga.
- **Astrazione multilingue:** il sistema sembra che lavori a due livelli, uno specifico della lingua e l'altro più astratto, indipendente dalla lingua⁸⁶.
- **Meccanismi legati ad allucinazioni e metacognizione:** circuiti che stimano se un'entità è nota o meno, spingendo il modello a inventare (allucinare) una risposta.

I risultati emersi da questo studio sono interessanti dal punto di vista scientifico ma evidenziano la difficoltà di questa tipologia di analisi che è limitata solo ad alcune zone del modello, data la vastità e la complessità delle possibili interazioni.

Dallo studio di Anthropic emergono comportamenti complessi all'interno dell'architettura Transformer che evidenziano fenomeni locali di ragionamento, pianificazione e autocontrollo. Alla luce di quanto detto, definire un LLM come un “pappagallo stocastico” è sicuramente riduttivo.

⁸⁵ Emmanuel Ameisen et al., *Circuit Tracing: Revealing Computational Graphs in Language Models*, Anthropic, 2025, <https://transformer-circuits.pub/2025/attribution-graphs/methods.html>

⁸⁶ I LLM manifestano comportamenti di trasferimento cross-linguistico, ossia la capacità di eseguire azioni in una lingua, avendo impartito istruzioni in un'altra lingua. Per maggiori informazioni A. D. Farashah et al., *AdaMergeX: Cross-Lingual Transfer with Large Language Models via Adaptive Adapter Merging*, 2024, <https://arxiv.org/abs/2402.18913>

Allo stato attuale delle conoscenze, l'unica certezza è che siamo ancora lontani dal comprenderne davvero il funzionamento interno dei LLM. Conosciamo nel dettaglio l'architettura, ma ci sfugge la comprensione dell'interazione tra le parti. E' questa complessità, oltre alla fase di *post training* con l'iterazione umana, che fa sì che questi modelli riescano a ottenere risultati così sorprendenti.

Benchmark

Per ora, lo strumento principale che abbiamo per cercare di valutare e confrontare le capacità dei LLM sono i benchmark, ossia raccolte di domande e risposte predefinite, messe insieme per testare competenze diverse in vari settori. Negli ultimi anni ne sono stati proposti a centinaia. In un certo senso, si possono considerare come dei test psicoattitudinali, con la differenza che, invece di essere somministrati a persone, vengono somministrati ai LLM.

Alcuni di questi benchmark hanno evidenziato capacità che eguagliano e a volte superano quelle umane.

Ad esempio, il benchmark MathArena⁸⁷ che valuta i modelli su problemi presi da competizioni matematiche pubblicate di recente, così da ridurre il rischio che i quesiti siano già presenti nei dati di addestramento. In questo benchmark gli autori riportano che, sulle prove di risoluzione di problemi, i modelli migliori come GPT-5, Grok-4, DeepSeek-3.2, Gemini-2.5-Pro e Claude-Sonnet-4.5 ottengono risultati tali da superare le prestazioni del top 1% dei partecipanti umani. Le prestazioni scendono drasticamente se si considerano però problemi basati su dimostrazioni, dove i LLM restano ben sotto il 40%, ad esempio sui testi delle Olimpiadi Internazionali della Matematica (IMO) 2025.

Altri benchmark invece mostrano dei livelli molto al di sotto delle capacità umane. Ad esempio il recente benchmark

⁸⁷ Mislav Balunovic et al., *MathArena: Evaluating LLMs on Uncontaminated Math Competitions*, 39th Conference on Neural Information Processing Systems (NeurIPS 2025), <https://arxiv.org/pdf/2505.23281>

APEX–Agents⁸⁸ ha evidenziato che i più avanzati LLM, in modalità di ragionamento (thinking) e con l'utilizzo di agenti, non sono in grado di sostituire le attività umane in progetti complessi a lungo termine. Nello specifico il benchmark ha analizzato alcune attività legate al mondo dell'analisi finanziaria (*investment banking*), della consulenza manageriale (*management consultants*) e delle attività legali in un'azienda (*corporate lawyer*). Il miglior risultato è stato quello di Gemini 3 Flash con appena un 24% di compiti portati a termine (Figura 35).

Esistono benchmark che dimostrano capacità di risoluzione di problemi oltre il livello umano e altri, dove i risultati sono ben al di sotto delle nostre capacità. Come è giustificabile questa differenza?

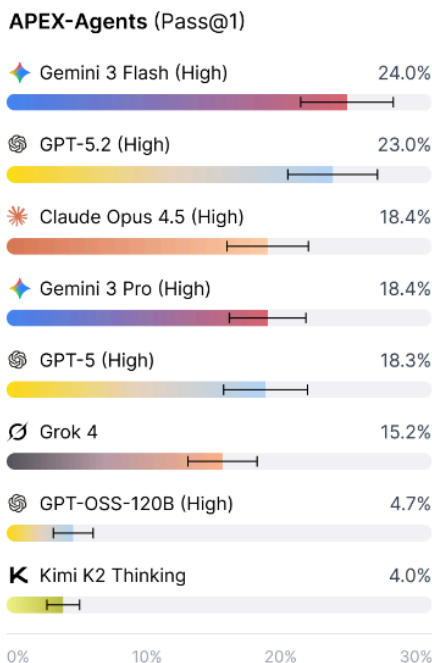


Figura 35: I risultati del benchmark APEX-Agents

⁸⁸ Bertie Vidgen et al., *APEX–Agents*, MERCOR Research, 20 Gennaio 2026, <https://arxiv.org/pdf/2601.14242>

Sistemi a intelligenza zero

L'espressione intelligenza artificiale evoca l'idea di una mente creata dall'uomo, un robot capace di ragionare come una persona. Il termine IA è stato inventato come operazione di marketing, per attrarre finanziamenti per l'avvio di questa nuova disciplina di studio nel 1956. In realtà, nei sistemi di IA non c'è nessuna mente, nessuna intelligenza nel senso umano del termine, per questo il paragone con l'uomo è fuorviante.

Il Prof. Luciano Floridi, uno dei massimi esperti al mondo di etica digitale, ha ideato il termine *artificial agency* per definire il comportamento di questi modelli di IA. Un *artificial agency* è un sistema a intelligenza zero ma con capacità di agire (*agency*).

Questa apparente contraddizione può essere spiegata con numerosi esempi tratti dalla nostra vita quotidiana. I robot aspiratori utilizzati per pulire le nostre abitazioni ne sono un esempio. Questi robot sono in grado di agire pulendo un appartamento in maniera autonoma, non hanno bisogno di essere guidati. Sembrano agire con intelligenza ma non per questo sono dotati di una mente artificiale. Il navigatore di Google Maps è un altro esempio. Questo software è in grado di considerare il traffico del nostro percorso e suggerire strade alternative ma lo fa in base a delle regole prestabilite. Sono queste regole, è l'algoritmo che risolve il nostro problema specifico come pulire casa o arrivare a destinazione. Il sistema agisce senza intelligenza.

I modelli di IA sono in grado di svolgere compiti complessi, anche oltre le nostre capacità, ma in un modo profondamente diverso da noi esseri umani.

Il Prof. Walter Quattrociocchi dell'Università di Roma ha coniato il neologismo *epistemia*, recentemente aggiunto nel vocabolario italiano della Treccani⁸⁹, per indicare la confortevole illusione di conoscenza prodotta dall'interazione con l'IA generativa dei grandi modelli linguistici (LLM), là dove la plausibilità simulativa del discorso fluente e la coerenza narrativa sostituiscono l'efficienza cognitiva e l'affidabilità dei

⁸⁹ [https://www.treccani.it/vocabolario/neo-epistemia_\(Neologismi\)/](https://www.treccani.it/vocabolario/neo-epistemia_(Neologismi)/)

dati. A parità di ignoranza, l'output generato da un LLM risulta più fluido, articolato, persuasivo di quello prodotto da un essere umano privo di competenze. Ma è un'efficienza simulativa, non cognitiva. E proprio per questo, più l'utente è inconsapevole dei limiti strutturali del modello, più è esposto al fascino della forma e all'equivoco dell'affidabilità.

Il Prof. Quattrociochi, insieme ad altri ricercatori, ha pubblicato un articolo⁹⁰ dove vengono analizzate le differenze tra il processo cognitivo umano nel formulare un giudizio con quello dei Large Language Model (Figura 36). Il modo di produrre giudizi di questi sistemi IA è profondamente diverso dalla cognizione umana. Invece di formare credenze o costruire modelli del mondo, gli LLM funzionano come sistemi stocastici che completano pattern linguistici, formalmente descrivibili come movimenti su grafi ad alta dimensionalità di transizioni linguistiche. Gli autori della ricerca individuano sette principali fratture epistemiche tra esseri umani e LLM, tra cui grounding, esperienza, motivazione, ragionamento causale e metacognizione. Definiscono questa condizione strutturale *epistemia*, una situazione in cui la plausibilità linguistica sostituisce la valutazione epistémica autentica, producendo un'illusione di conoscenza.

Nel discutere l'epistemologia dei Large Language Models è utile tornare al tema delle proprietà emergenti introdotto in precedenza. Il punto filosofico centrale riguarda infatti il tipo di emergenza di cui stiamo parlando. Nei sistemi fisici si parla spesso di emergenza ontologica, cioè della comparsa di nuove proprietà causali che esistono realmente nel mondo.

Nel caso degli LLM, però, il rischio è diverso: potremmo confondere una vera emergenza con una sorpresa epistémica. Quando un modello mostra una capacità inattesa, tendiamo a definirla "emergente", ma questa sorpresa potrebbe dipendere semplicemente dal fatto che non siamo in grado di comprendere completamente sistemi con miliardi di parametri.

90

Walter Quattrociochi, Valerio Capraro, Matjaž Perc, *Epistemological Fault Lines Between Human and Artificial Intelligence*, 2025, <https://arxiv.org/pdf/2512.19466>

In altre parole, ciò che appare come una nuova proprietà del modello potrebbe riflettere soprattutto i limiti della nostra comprensione, più che una reale novità ontologica.

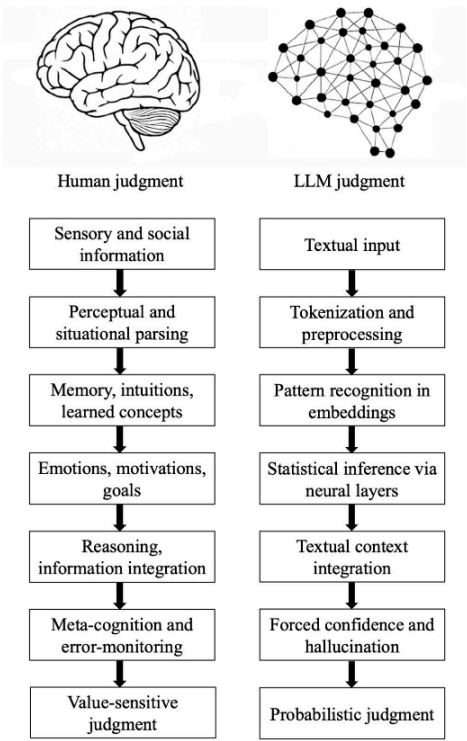


Figura 36: Differenze tra giudizio umano e LLM

Dalle ricerche presentate in questo capitolo emerge un quadro tutt'altro che chiaro sulle capacità reali dei LLM. Alcuni benchmark mostrano prestazioni che superano quelle umane, mentre altri mettono in luce limiti significativi, soprattutto nei compiti più complessi e in quelli che richiedono una conoscenza articolata del mondo. Una parte degli studiosi guarda con preoccupazione alla possibile evoluzione di questi modelli, ipotizzando che, con l'aumento di scala e di potenza computazionale, possano raggiungere o persino superare le capacità umane. Altri ricercatori, invece, ridimensionano queste

interpretazioni, sottolineando la natura stocastica dei modelli e mettendo in guardia rispetto all'apparente affidabilità delle loro risposte.

Ciò che appare evidente è che ci troviamo in una fase storica ancora esplorativa. Stiamo ancora cercando di comprendere quali siano le reali capacità di questi sistemi e quali i loro limiti strutturali. Forse l'errore più comune è continuare a misurarli esclusivamente in rapporto all'intelligenza umana. Il punto, più che stabilire una competizione, è capire con precisione che cosa abbiamo effettivamente costruito. Siamo di fronte a una nuova forma di intelligenza oppure a un sofisticato effetto di superficie che produce l'impressione di comprendere senza realmente comprendere?

8. IA GENERATIVA: NON SOLO PAROLE

Negli ultimi anni l'intelligenza artificiale ha compiuto un salto qualitativo che ha cambiato radicalmente il modo in cui interagiamo con le macchine. Questo cambiamento è legato all'affermazione dell'IA generativa, una famiglia di modelli capaci non solo di analizzare dati o fare previsioni, ma anche di produrre nuovi contenuti testuali, visivi, sonori e persino di codice. A differenza dei sistemi del passato, progettati per svolgere compiti specifici e ben delimitati, l'IA generativa si presenta come uno strumento flessibile, in grado di adattarsi a contesti molto diversi tra loro.

Ma come fanno questi modelli a generare dei risultati diversi anche a parità di input? Se provate a porre la stessa domanda a uno di questi modelli, otterrete una risposta sempre diversa.

Il motivo di questa diversità nelle risposte è legato al caso, o meglio all'utilizzo di una funzione pseudo casuale all'interno dell'algoritmo. Nel capitolo 6 del libro abbiamo introdotto l'architettura Transformer dei LLM e in particolare l'algoritmo top-k per la scelta del prossimo token. Ebbene la scelta casuale della prossima parola (token) da utilizzare è la parte generativa del modello⁹¹. Grazie a questa tecnica, il modello è in grado di generare delle risposte diverse, più o meno simili tra di loro⁹².

⁹¹ Qualcuno potrebbe anche definirla come la parte “creativa” del modello.

⁹² Il grado di similitudine delle risposte può essere variato utilizzando il parametro della temperatura. Per maggiori informazioni si rimanda al

Una delle caratteristiche innovative dei modelli recenti di IA è il fatto che possono operare non soltanto sul testo ma anche su immagini, video, file, programmi, etc.

Modelli multimodali

Per molti anni i modelli linguistici hanno vissuto in un mondo fatto solo di parole, ricevevano testo in ingresso e producevano testo in uscita, oggi una parte crescente dei sistemi più avanzati è multimodale, cioè capace di capire e generare informazioni che non sono soltanto linguistiche, come immagini, audio, video, diagrammi, schermate di software, fotografie di oggetti, questo passaggio cambia la natura stessa dell'interazione, perché non si tratta più di conversare soltanto su ciò che è scritto, si tratta di ragionare su ciò che è visto e, in alcuni casi, di creare contenuti visivi coerenti con un'idea espressa a parole.

L'espressione LLM multimodale può indicare due famiglie di soluzioni, la prima è un modello linguistico a cui viene collegato un modulo visivo o sonoro, la seconda è un modello progettato fin dall'inizio per lavorare con più tipi di dati. In entrambi i casi le informazioni ingresso, parole, immagini e suoni vengono trasformate in sequenze di numeri, così come le frasi vengono trasformate in token. Per esempio un'immagine può essere divisa in piccole porzioni, chiamate patch, e ogni patch viene convertita in una tabella di numeri che cattura colori, contorni, texture, relazioni spaziali, etc. Questa serie di numeri rappresenta un vettore⁹³, ossia una proiezione del dato in uno spazio multidimensionale in cui vive il linguaggio, a quel punto il modello può collegare ciò che vede con il significato delle parole.

Quando un LLM multimodale interpreta un'immagine, di solito entra in gioco un encoder⁹⁴ visivo, spesso basato su architetture

Capitolo 6.

⁹³ Un vettore è una lista di numeri che rappresenta un punto in uno spazio a n dimensioni (multidimensionale). Ad esempio, in uno spazio a 2 dimensioni, il vettore (1,1) rappresenta il punto che si trova alle coordinate $x=1$ e $y=1$.

⁹⁴ Un encoder è un sistema in grado di codificare, di rappresentare un dato in

tipo Vision Transformer⁹⁵. L'encoder produce una sequenza di numeri, detti embedding, che rappresentano l'immagine.

Questi embedding che rappresentano l'immagine vengono passati al modello linguistico tramite un componente che permette ai due sistemi di comunicare. In questo modo il modello può, per così dire, guardare l'immagine mentre genera il testo.

Le frasi prodotte da un modello multimodale, non nascono da una semplice descrizione automatica, il modello collega ciò che vede nell'immagine con le parole, con le conoscenze generali che possiede, con il contesto della conversazione e con le istruzioni date dall'utente.

Il caso più affascinante, e spesso più frainteso, è la generazione di immagini che non vengono dipinte punto per punto (pixel per pixel). I modelli si comportano più da registi, traducendo l'intenzione espressa a parole in una descrizione interna ben strutturata, affidando successivamente la parte pittorica a un generatore di immagini. Questa divisione del lavoro è importante perché il linguaggio e i pixel hanno statistiche molto diverse, e richiedono competenze diverse. I modelli linguistici ragionano sulla coerenza delle frasi, i modelli di immagine trasformano questa intenzione semantica in una struttura visiva.

Per capire il meccanismo, immaginiamo di scrivere il seguente prompt: “Genera un’illustrazione realistica di una piazza italiana al tramonto, ci sono tavolini di un bar, luci calde, una fontana al centro, sullo sfondo una chiesa romanica, in primo piano una Vespa parcheggiata, atmosfera cinematografica, profondità di campo” (Figura 37).

una serie di numeri leggibili da un sistema di IA.

⁹⁵ https://en.wikipedia.org/wiki/Vision_transformer



Figura 37: Immagine generata con GPT-5.2 utilizzando il prompt riportato nel libro

Il sistema fa un primo passaggio, il testo viene convertito in embedding, cioè in una sequenza numerica che cattura significato, stile, relazioni tra oggetti, e soprattutto vincoli, come il tramonto, la luce calda, la fontana centrale, la Vespa in primo piano. Se è presente un modello che funge da regista, può anche riscrivere internamente il prompt in una forma più precisa, può risolvere ambiguità, per esempio decidere che la fontana deve

essere leggermente decentrata per una composizione più naturale, può stabilire un'inquadratura, può fissare un'ottica virtuale, può anche evitare richieste in conflitto, come ombre impossibili o prospettive incoerenti.

Poi entra in gioco il modello di generazione dell'immagine, basato sul concetto di diffusione. La diffusione funziona in modo controintuitivo, non parte da una tela bianca, parte da insieme casuale di punti e l'obiettivo del modello è trasformare quel rumore in un'immagine che rispetti gli embedding del testo (il significato delle parole). Il processo avviene per passi, a ogni passo il modello osserva l'immagine ancora rumorosa, stima quale parte del rumore è di troppo e la rimuove leggermente, ripete questa operazione decine o centinaia di volte, finché emergono forme stabili (Figura 38).

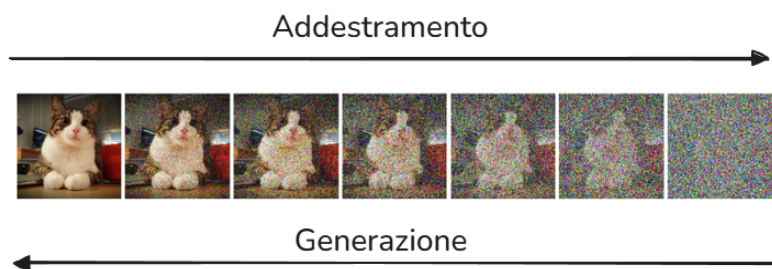


Figura 38: Fase di addestramento (sinistra verso destra) e fase di generazione (destra verso sinistra) di un modello di diffusione

La condizione testuale guida ogni fase, spesso con un meccanismo chiamato *guidance*, che amplifica la fedeltà al prompt, se la *guidance* è troppo bassa l'immagine può diventare generica, se è troppo alta può risultare artificiale o eccessivamente letterale.

Per fare un paragone, potremmo pensare a un processo di scultura, il rumore è un blocco di marmo pieno di imperfezioni, il testo è l'idea della statua, la diffusione è la sequenza di colpi che toglie materia superflua, finché resta solo ciò che assomiglia all'idea, non è una metafora perfetta, ma rende bene la natura iterativa del procedimento, e soprattutto spiega perché spesso i primi passaggi producono immagini caotiche e i passaggi finali

rendono tutto nitido. Questo meccanismo aiuta anche a capire alcuni fenomeni tipici, se nel prompt chiedi “una Vespa rossa” e “un tramonto con luce calda”, l’immagine potrebbe rispettare entrambi, ma in alcuni casi un vincolo vince sull’altro, per esempio la Vespa diventa arancione perché la luce calda influenza i colori, oppure la Vespa è rossa ma la luce sembra meno dorata. Qui il ruolo del componente linguistico e del sistema di controllo diventa importante, perché può introdurre vincoli più espliciti, può insistere su attributi chiave, può scegliere formulazioni che riducono le ambiguità, può anche proporre una variante, “Vespa rosso acceso, illuminata da luce dorata, ma mantenendo il rosso saturo”, così il generatore ha una guida più chiara.

Gli LLM multimodali non sono soltanto generatori di immagini, sono soprattutto strumenti di ragionamento su contenuti visivi. La stessa architettura che collega patch visive a token linguistici permette di leggere un grafico, interpretare una radiografia, seguire i passaggi di un problema di geometria disegnato su carta, riconoscere un errore in uno screenshot di codice, questo amplia l’idea di assistente, non più un correttore di testi, ma un sistema che può identificare l’oggetto del discorso. Allo stesso tempo, questa potenza porta nuove responsabilità, perché la multimodalità aumenta la superficie di rischio, sia sul piano della privacy, perché una foto può contenere dati personali involontari, sia sul piano della sicurezza, perché immagini e testi possono essere usati per ingannare, manipolare, falsificare.

In prospettiva, l’aspetto più interessante di questi modelli non riguarda tanto la qualità delle immagini generate, quanto la capacità di mantenere coerenza tra le parole e le immagini, usando il mondo visivo come una memoria esterna. Un diagramma, una bozza, una schermata, una foto, diventano parte del ragionamento, e il linguaggio diventa il collante che coordina percezione, pianificazione e creatività. In questa fusione tra parola e immagine si intravede una nuova fase dell’informatica, in cui chiedere, spiegare, creare e correggere avvengono nello stesso spazio conversazionale, e l’intelligenza artificiale smette di essere un semplice completatore di frasi, per

diventare un interprete e un costruttore di rappresentazioni del mondo.

L'IA generativa non è quindi soltanto una nuova tecnologia, ma un cambiamento culturale. Rende più sottile il confine tra produzione umana e produzione automatica, costringendoci a ripensare il valore della creatività, della conoscenza e dell'autorialità. Comprenderne il funzionamento, i limiti e le potenzialità è essenziale per usarla in modo consapevole. Come spesso accade nella storia dell'innovazione, non è lo strumento in sé a determinare il futuro, ma il modo in cui la società sceglie di integrarlo nella vita quotidiana.

Esempi di modelli

ChatGPT

ChatGPT⁹⁶ è il prodotto che ha reso popolare il dialogo (chat) con un modello generativo che simula il comportamento di un assistente umano. È sviluppato da OpenAI⁹⁷, nata come fondazione no-profit poi diventata un'azienda statunitense con sede a San Francisco e viene offerto come servizio accessibile via web e tramite app.

La sua peculiarità storica non è solo la qualità linguistica, ma la scelta dell'interazione conversazionale come formato principale: la possibilità di fare domande successive, correggere il tiro, chiedere spiegazioni, far emergere i passaggi intermedi di un compito complesso. L'architettura e l'addestramento dietro le quinte sono pensati per trasformare un modello generativo in un sistema capace di seguire istruzioni e mantenere un filo nel dialogo, riducendo risposte incoerenti o fuori tema. Come la maggior parte delle aziende che producono modelli, OpenAI dedica molte risorse economiche nella fase di addestramento del modello con persone in carne e ossa. Questo è l'elemento distintivo in grado di dare un tocco umano alle conversazioni. La pagina di presentazione originale insiste proprio su questo punto: ChatGPT è stato addestrato per interagire in forma di

⁹⁶ <https://chatgpt.com/>

⁹⁷ <https://openai.com/>

dialogo e gestire domande di follow-up, ammettere errori⁹⁸ e correggersi.

Nel tempo, ChatGPT si è anche trasformato in una sorta di contenitore di strumenti: non solo scrittura e sintesi, ma supporto allo studio, al lavoro d'ufficio e alla programmazione. È una scelta di prodotto, prima ancora che di modello. L'utente deve essere guidato per la risoluzione di un compito tramite una conversazione a iterazioni successive, come in un laboratorio di scrittura o in una sessione di pair programming⁹⁹.

Gemini

Gemini¹⁰⁰ è una famiglia di modelli sviluppata da Google DeepMind, laboratorio di ricerca anglo-statunitense con sede a Londra e centri di ricerca anche negli Stati Uniti. In pratica, quando si parla di Gemini si parla insieme di un modello e di un assistente: la famiglia di modelli e l'interfaccia che li rende accessibili.

La peculiarità più citata di Gemini è l'idea di multimodalità come caratteristica progettuale: non solo testo, ma la capacità di lavorare con più tipi di input nello stesso contesto, combinando informazioni in modo più fluido. La documentazione descrive Gemini come un modello multimodale e la voce enciclopedica sottolinea che nel contesto possono coesistere forme diverse di input. Questa scelta ha un impatto pratico: invece di trattare immagini, audio o video come funzioni separate, il sistema tende a incorporarli come parti di una conversazione unificata, utile per compiti come analisi di materiali misti, spiegazioni a partire da fonti eterogenee e assistenza nella produzione di contenuti che attraversano più media.

Un'altra peculiarità è l'integrazione: Gemini nasce dentro un ecosistema di servizi, applicazioni e flussi di lavoro già esistenti.

⁹⁸ In ChatGPT e in altri servizi è presente un messaggio che avvisa che il modello può commettere errori e che è necessario assicurarsi di verificare le risposte, nel caso di informazioni importanti.

⁹⁹ Il pair programming è lo sviluppo di codice a quattro mani, con una supervisione contemporanea di due persone (o di una persona e di un modello di IA).

¹⁰⁰ <https://gemini.google.com>

Questa capacità di integrazione è importante perché sposta il focus dall'idea di modello come cervello isolato all'idea di modello come collante tra strumenti: scrivere un testo, riassumere un documento, trasformare appunti in un output più ordinato, oppure passare con naturalezza dalla domanda generale al lavoro su una fonte specifica.

Claude

Claude¹⁰¹ è sviluppato da Anthropic, società statunitense con sede a San Francisco. È una famiglia di modelli che, fin dalle origini, ha comunicato una forte attenzione a sicurezza e controllabilità: non solo prestazioni, ma comportamento, affidabilità e limiti.

La peculiarità più riconoscibile, per molti utenti, è lo stile: spesso orientato a risposte strutturate, caute nei punti delicati, con un'enfasi sulla trasparenza quando qualcosa è incerto. A livello di impostazione, Anthropic descrive se stessa come azienda di ricerca sulla sicurezza dell'IA, impegnata a costruire sistemi affidabili e governabili. Questo si riflette anche nei documenti pubblici sulla governance del comportamento del modello, come l'idea di una costituzione che esplicita principi e vincoli, un tentativo di rendere più prevedibile come il sistema decide cosa fare e cosa rifiutare.

Un secondo elemento pratico è il lavoro sul contesto lungo, cioè la quantità di testo e informazioni che il modello può tenere a mente durante una sessione. Più contesto significa poter lavorare su documenti lunghi, progetti articolati, conversazioni che durano, senza dimenticare parti essenziali. Nel tempo la famiglia Claude è stata spesso citata proprio per questa capacità e per l'uso in compiti professionali dove la continuità conta, come revisione di testi estesi, analisi di requisiti, ragionamento su specifiche tecniche.

Mistral

Mistral¹⁰² identifica sia una società sia un insieme di modelli.

¹⁰¹ <https://claude.ai/>

¹⁰² <https://mistral.ai/>

Mistral AI è un'azienda francese con sede a Parigi. Nel panorama dominato da attori statunitensi, la presenza di un produttore europeo con ambizioni di frontiera è diventata rapidamente un riferimento, anche per ragioni culturali e regolatorie: sovranità digitale, ecosistema industriale europeo, capacità di innovare senza dipendere interamente da infrastrutture esterne.

La peculiarità tecnica che Mistral ha reso popolare è l'attenzione all'efficienza: ottenere prestazioni competitive con modelli relativamente compatti e ottimizzati. Un esempio molto citato è Mistral 7B, presentato come modello rilasciato con licenza Apache 2.0¹⁰³ e con scelte architetture mirate a velocità e gestione di contesti più lunghi a costo contenuto. In parallelo, la società ha pubblicato una gamma di modelli descritti come open weight¹⁰⁴ e orientati sia al cloud sia all'esecuzione su dispositivi e infrastrutture più vicine all'utente.

Un aspetto interessante di questi open weight è il potenziale effetto sul mercato, modelli disponibili con pesi scaricabili e licenze permissive rendono più facile sperimentare, costruire prototipi, integrare l'IA in un prodotto senza consegnare tutto a un servizio remoto. È un cambio di prospettiva che riduce la distanza tra ricerca, industria e comunità di sviluppatori.

DeepSeek

DeepSeek¹⁰⁵ sono una famiglia di modelli sviluppati da Hangzhou DeepSeek Artificial Intelligence Basic Technology Research, azienda cinese con sede a Hangzhou, nella provincia di Zhejiang. Negli ultimi anni DeepSeek ha attirato l'attenzione per due motivi che spesso vanno insieme: buone prestazioni su

¹⁰³ Apache 2.0 è una tipologia di licenza open source che permette l'uso, la modifica e la redistribuzione anche commerciale di un software. Inoltre, riduce il rischio legato ai brevetti, che in ambito software può essere molto rilevante.

¹⁰⁴ Per open weight si intende un modello che è possibile scaricare liberamente (dove i pesi del modello vengono resi noti). Ne parleremo in maniera più diffusa tra qualche pagina.

¹⁰⁵ <https://www.deepseek.com/>

compiti di ragionamento e programmazione, e una narrativa di efficienza nei costi di addestramento e utilizzo.

Dal punto di vista della disponibilità, una caratteristica notevole è la scelta di rilasciare versioni con pesi open weight sotto licenze permissive. Nelle note di rilascio, DeepSeek dichiara che alcuni modelli, inclusi DeepSeek-R1 e release della linea V3, sono distribuiti con licenza MIT¹⁰⁶ e con pesi resi disponibili alla comunità. Anche repository pubblici collegati al progetto descrivono addestramento su grandi quantità di token e un posizionamento esplicito come alternativa open rispetto a modelli solo via API.

Qui emerge una differenza interessante rispetto ad altri attori, ossia l'apertura dei pesi, quando è accompagnata da licenze davvero permissive, può accelerare l'adozione globale perché permette adattamenti locali, fine-tuning, compressione e distillazione. In altre parole, un modello non resta un monolite, diventa una base da cui derivano versioni specializzate, più leggere o più controllate, adatte a contesti diversi.

Open weight e open source: una distinzione che conta

Nel linguaggio comune, soprattutto fuori dagli ambienti legali e delle comunità open source¹⁰⁷, si tende a chiamare open source qualsiasi modello di cui si possano scaricare i pesi. Se posso scaricarlo e usarlo, sembra aperto. Tuttavia la distinzione tra open weight e open source, in senso rigoroso, è importante.

Per open weight si intende un modello di cui i parametri addestrati, cioè i pesi, sono pubblicamente disponibili e quindi utilizzabili e modificabili con tecniche come il fine-tuning. Questo però non implica automaticamente che siano disponibili anche gli elementi necessari per studiare e riprodurre il sistema in modo completo, come dati di addestramento, codice, documentazione estesa e vincoli di licenza realmente

¹⁰⁶ La licenza MIT è una tra le licenze open source più permissive.

¹⁰⁷ Per open source si intende un programma che viene rilasciato con i codici sorgenti. In questo modo chiunque è in grado di capire il funzionamento del programma e, a seconda della licenza utilizzata, avere la possibilità di modificarli all'occorrenza.

compatibili con la definizione classica di open source.

L'Open Source Initiative¹⁰⁸ ha lavorato proprio per chiarire questo punto, da un lato spiega cosa siano gli open weights e perché non coincidano con l'idea di open source applicata all'IA, dall'altro propone una Open Source AI Definition che definisce requisiti più ampi rispetto al semplice rilascio dei pesi. In questa prospettiva, un modello può essere open weight senza essere open source, e un'etichetta confusa rischia di creare aspettative sbagliate su trasparenza, riusabilità e diritti.

Nel panorama reale, molte organizzazioni pubblicano pesi e parte del codice, ma mantengono opachi o non distribuibili i dati di addestramento, oppure impongono restrizioni d'uso incompatibili con i criteri *open source* più tradizionali. Il risultato non è qualcosa che si possa ridurre a un semplice alternativa tra chiuso o aperto, ma una gamma di possibilità che va dalla chiusura totale all'apertura dei pesi, fino a forme più complete che implicano una totale trasparenza tecnologica, nel rispetto del termine open source.

Eseguire modelli sul proprio computer

Quando i pesi sono disponibili, sorge spontanea una domanda: come si esegue davvero un modello sul proprio computer, senza passare ogni volta da un servizio remoto? Qui entrano in scena strumenti che distribuiscono modelli e runtime in modo accessibile.

Ollama¹⁰⁹ è uno dei progetti più citati perché punta a rendere semplice avviare modelli linguistici in locale, con installazioni e comandi pensati per ridurre la complessità di dipendenze, formati e accelerazione hardware. Il repository ufficiale descrive Ollama come un modo per partire rapidamente con i large language models, con supporto a sistemi operativi comuni e modalità di installazione dirette.

L'esecuzione locale è una scelta legata a diversi aspetti. Primo

¹⁰⁸ L'Open Source Initiative è un'organizzazione no profit che lavora per aiutare i progetti open source. Per informazioni: <https://opensource.org/>

¹⁰⁹ <https://ollama.com/>

tra tutti l'aspetto della privacy, i dati non viaggiano su internet e non vengono memorizzati in server remoti. Altri fattori sono il controllo totale del modello e dei suoi parametri che spesso non sono modificabili in servizi cloud. Anche l'aspetto delle performance può essere un fattore determinante, la latenza può diminuire non dovendo eseguire delle chiamate su internet. D'altro canto, l'esecuzione in locale richiede dei prerequisiti (costi): memoria, potenza di calcolo, gestione della GPU¹¹⁰, e la necessità di scegliere modelli compatibili con l'hardware. Infine, c'è da dire che in locale si possono far girare modelli non troppo grandi con un grado di accuratezza vicino ma non paragonabile a quello dei modelli più grandi come GPT e Claude Sonnet che richiedono un'enorme potenza di calcolo. I modelli open weight stanno diventando sempre più potenti e nel giro di un anno hanno quasi raggiunto la potenza di quelli commerciali in alcuni ambiti applicativi.

Non solo modelli, alcuni servizi di IA

Accanto ai modelli di IA, stanno crescendo servizi che non si presentano come un singolo chatbot generalista, ma come ambienti di lavoro costruiti attorno a un compito specifico. In questi casi il punto non è solo cosa sa fare il modello, ma come il prodotto lo guida a usare fonti, citazioni, documenti e flussi di ricerca.

NotebookLM¹¹¹, di Google, è uno strumento di ricerca e di supporto al pensiero che lavora sulle fonti caricate dall'utente, con l'idea di trasformare materiali complessi in appunti, sintesi e spiegazioni più maneggevoli. Ad esempio, è in grado di generare una presentazione in formato PDF partendo da uno o più fonti caricate dall'utente o da un elenco di pagine web.

¹¹⁰ Una GPU (Graphics Processing Unit) è un processore specializzato nella gestione ed elaborazione parallela di grandi quantità di dati, originariamente progettato per la grafica computerizzata ma oggi ampiamente utilizzato anche per calcoli ad alte prestazioni, intelligenza artificiale e machine learning. Il suo principale vantaggio è la capacità di eseguire simultaneamente migliaia di operazioni in parallelo, risultando molto più efficiente e veloce di una CPU nei compiti altamente ripetitivi e computazionalmente intensivi.

¹¹¹ <https://notebooklm.google/>

Perplexity¹¹², invece, rappresenta la categoria dei motori di risposta che uniscono modello generativo e ricerca sul web in tempo reale, con un'enfasi sulle fonti consultate. In questo caso il modello non è solo un autore, ma anche un mediatore tra domanda e documenti pubblici, con un comportamento che ricorda la consultazione di una bibliografia.

Ci sono molti altri servizi che nascono ogni giorno, ad esempio possiamo citare strumenti orientati alla ricerca accademica, come Elicit¹¹³ e SciSpace¹¹⁴, che puntano a cercare, riassumere e organizzare letteratura scientifica, spesso con funzioni di estrazione dati e lettura assistita. Insieme, questi servizi mostrano una tendenza chiara, il futuro dei modelli generativi non è solo un modello più grande, ma prodotti che incanalano la generazione dentro procedure, fonti e contesti verificabili.

¹¹² <https://www.perplexity.ai/>

¹¹³ <https://elicit.com/>

¹¹⁴ <https://scispace.com/>

9. ETICA E REGOLAMENTAZIONE DELL'IA

L'etica dell'intelligenza artificiale è un tema centrale nello sviluppo e nell'applicazione di tecnologie avanzate, poiché l'IA ha un impatto significativo su molti aspetti della vita sociale, economica e politica. L'intelligenza artificiale non è solo uno strumento tecnico, ma una tecnologia che interagisce con la società in modi complessi e spesso imprevedibili, influenzando decisioni che riguardano la privacy, l'equità e l'autonomia delle persone. Per questo motivo, è essenziale che le aziende e i ricercatori si interrogino sulle implicazioni morali delle loro scelte progettuali e implementative, adottando un approccio etico che metta al centro i valori umani.

Il Prof. Luciano Floridi ha esplorato questi temi nel suo libro *Etica dell'Intelligenza Artificiale*¹¹⁵. Floridi sottolinea come l'intelligenza artificiale debba essere progettata non solo per massimizzare l'efficienza o la precisione, ma anche per rispettare principi di trasparenza, responsabilità e inclusività. Uno degli esempi citati riguarda l'utilizzo di algoritmi decisionali nei sistemi giudiziari, dove l'IA può aiutare a identificare pattern di recidiva o suggerire sentenze, ma al contempo rischia di incorporare comportamenti discriminatori (bias), influenzando così l'equità dei processi legali.

¹¹⁵ Luciano Floridi, *Etica dell'intelligenza artificiale. Sviluppi, opportunità, sfide*, Raffaello Cortina Editore, 2022

Floridi evidenzia anche le opportunità offerte dall'IA in ambito medico, dove può migliorare la diagnosi precoce e personalizzare i trattamenti. Tuttavia, il rischio è che la dipendenza da algoritmi opachi possa ridurre l'autonomia decisionale dei medici e compromettere la relazione di fiducia con i pazienti. In questo contesto, Floridi propone una visione dell'intelligenza artificiale come tecnologia assistiva, che supporti l'essere umano senza sostituirlo nelle decisioni critiche. Attraverso esempi come questi, Floridi invita le aziende e i ricercatori a considerare attentamente l'impatto sociale dell'intelligenza artificiale e a sviluppare modelli che rispettino la dignità e i diritti fondamentali degli individui. L'etica dell'IA non è solo una questione di regole da seguire, ma un impegno continuo a garantire che la tecnologia contribuisca al bene comune, mitigando i rischi e promuovendo opportunità reali per una società più equa e inclusiva.

AI Act

Negli ultimi anni l'Unione Europea ha lavorato su una proposta di regolamentazione dell'intelligenza artificiale. Il 13 giugno 2024 è stata approvata la regolamentazione 1689/2024 sull'intelligenza artificiale, denominata **AI Act**. Questa regolamentazione pionieristica, la prima della storia, stabilisce standard di sicurezza, trasparenza e affidabilità per garantire uno sviluppo etico e responsabile. Questo regolamento classifica i sistemi di IA in base al livello di rischio associato al loro uso, differenziando tra rischi inaccettabili, elevati, limitati e minori (Figura 39).

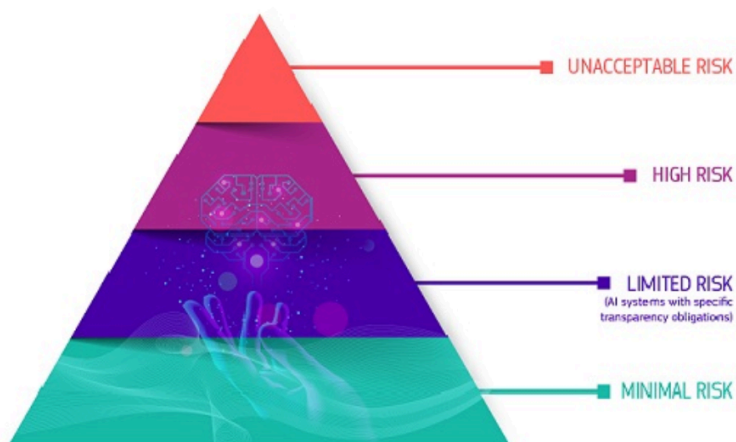


Figura 39: La piramide dei rischi presente nell'AI Act

L'AI Act adotta un approccio basato sul rischio: quanto maggiore è il potenziale impatto di un sistema di intelligenza artificiale su sicurezza, salute e diritti fondamentali, tanto più severi sono gli obblighi normativi. I sistemi vietati perché ritenuti a rischio inaccettabile comprendono, tra l'altro, pratiche manipolative o ingannevoli, forme di sfruttamento delle vulnerabilità delle persone, il social scoring¹¹⁶, alcune forme di categorizzazione biometrica basata su dati sensibili, il riconoscimento delle emozioni in contesti sensibili come scuola e lavoro, nonché determinati usi dell'identificazione biometrica remota in tempo reale da parte delle forze dell'ordine, salvo eccezioni molto restrittive.

I sistemi ad alto rischio, invece, sono ammessi ma sottoposti a requisiti rigorosi. Rientrano in questa categoria i sistemi usati, ad esempio, in biometria, infrastrutture critiche, istruzione, occupazione e selezione del personale, accesso a servizi essenziali, attività di law enforcement, gestione delle frontiere, giustizia e processi democratici. Per tali sistemi il regolamento impone, tra l'altro, gestione del rischio, qualità e governance dei dati, documentazione tecnica, tracciabilità, supervisione umana,

¹¹⁶ Il social scoring è un sistema che valuta o classifica le persone in base al loro comportamento sociale, o a caratteristiche personali note, inferite o previste, per attribuire loro un punteggio di affidabilità o merito.

requisiti di accuratezza, robustezza e cybersicurezza, oltre a valutazioni di conformità prima dell'immissione sul mercato o dell'uso.

Un elemento particolarmente innovativo dell'AI Act è l'introduzione di obblighi specifici anche per i modelli di IA a finalità generale (general-purpose AI models), inclusi i modelli generativi che possono essere integrati in molteplici applicazioni. Per questi modelli sono previsti obblighi di trasparenza, documentazione tecnica, informazioni per gli sviluppatori a valle e misure collegate ai dati di addestramento; per i modelli più potenti, suscettibili di comportare rischi sistemici, si aggiungono obblighi più stringenti di valutazione e mitigazione del rischio e di cybersicurezza.

L'AI Act prevede inoltre obblighi di trasparenza verso gli utenti. Per esempio, i chatbot devono informare chiaramente la persona di interagire con una macchina, mentre determinati contenuti generati o manipolati artificialmente, come i deepfake¹¹⁷, devono essere etichettati o resi riconoscibili come tali. Questa dimensione della trasparenza è centrale perché mira a ridurre i rischi di inganno, manipolazione informativa e opacità decisionale.

Sul piano istituzionale, il regolamento istituisce una struttura di governance multilivello composta dalle autorità nazionali competenti, dall'AI Office della Commissione europea, dall'AI Board, da un panel scientifico di esperti indipendenti e da un forum consultivo degli stakeholder. Tale architettura serve a garantire un'applicazione uniforme delle regole nel mercato unico europeo e una vigilanza specifica sui modelli di IA a finalità generale.

Infine, il regolamento prevede un sistema sanzionatorio molto incisivo, modellato su logiche simili a quelle del GDPR: per alcune violazioni, in particolare delle pratiche vietate, le sanzioni possono arrivare fino a 35 milioni di euro o al 7% del fatturato annuo mondiale; per altre violazioni gravi fino a 15 milioni o al

¹¹⁷ Per deepfake si intendono le tecniche di manipolazioni delle immagini e volte ad impersonificare altre persone. Le tecnologie attuali di deepfake sono arrivate a un livello tale che è sempre più difficile distinguere un'immagine vera da una falsa. Esempio: <https://www.youtube.com/watch?v=B4jNttRvbpU>

3%; per informazioni scorrette o fuorvianti fino a 7,5 milioni o all'1%, con un regime proporzionato per PMI e start-up.

L'importanza di un approccio etico nello sviluppo dell'intelligenza artificiale va oltre la conformità normativa; ogni azienda impegnata nella creazione di modelli di IA ha una responsabilità morale di garantire che i propri prodotti siano sicuri, giusti e trasparenti. L'IA può influenzare aspetti cruciali della vita umana, come l'accesso a servizi essenziali, l'equità nei processi decisionali e la tutela della privacy, ed è quindi fondamentale che le aziende implementino misure per garantire che gli algoritmi siano imparziali, spiegabili e privi di bias. Adottare un approccio etico contribuisce inoltre alla fiducia degli utenti e alla legittimità dell'intelligenza artificiale, evitando che tecnologie potenzialmente invasive o discriminatorie minino i diritti e le libertà individuali.

L'AI Act rappresenta un tentativo di bilanciare innovazione tecnologica e tutela dei diritti fondamentali, introducendo un modello regolatorio che non vieta l'intelligenza artificiale in quanto tale, ma ne differenzia la disciplina in funzione del rischio. La portata storica del regolamento risiede proprio nell'aver trasformato principi etici come trasparenza, responsabilità, non discriminazione e supervisione umana in obblighi giuridici concreti, destinati a incidere non solo sul mercato europeo, ma anche sugli standard internazionali di sviluppo e impiego dell'IA.

La legge italiana sull'IA

Il Senato ha approvato in via definitiva la legge italiana sull'Intelligenza Artificiale (Legge 23 settembre 2025, n. 132). È un passaggio rilevante perché introduce, per la prima volta in Europa, un quadro nazionale che regola sviluppo, adozione e governo dei sistemi di IA, tenendo insieme innovazione, principi costituzionali e tutela dei diritti fondamentali, in coerenza con l'AI Act dell'Unione europea.

Il principio cardine della normativa è che l'intelligenza artificiale debba essere sviluppata e utilizzata in modo da restare al servizio della persona. Questo significa promuovere un uso

antropocentrico, trasparente e sicuro, con attenzione a quattro snodi decisivi: innovazione, cybersicurezza, accessibilità e protezione della riservatezza. La norma interviene in modo organico in ambiti dove l'IA può portare benefici concreti in ambiti come sanità, lavoro, pubblica amministrazione e giustizia, formazione e sport, introducendo garanzie come tracciabilità dei processi, responsabilità umana e, soprattutto, centralità della decisione finale di una persona fisica.

Sul piano della governance, la legge individua due Autorità nazionali competenti con ruoli complementari. L'Agenzia per la Cybersicurezza Nazionale¹¹⁸ (ACN) vigila sull'adeguatezza e sulla sicurezza dei sistemi di IA, anche con poteri ispettivi. L'Agenzia per l'Italia Digitale¹¹⁹ (AgID) gestisce le notifiche e promuove casi d'uso affidabili per cittadini e imprese.

Il provvedimento introduce anche un meccanismo di programmazione strategica: la Strategia nazionale per l'IA sarà predisposta e aggiornata ogni due anni dal Dipartimento per la trasformazione digitale della Presidenza del Consiglio, con il supporto di ACN e AgID e il coinvolgimento delle principali autorità di settore. Per rafforzare la trasparenza, è previsto inoltre un monitoraggio annuale al Parlamento.

Per accelerare competitività e diffusione dell'IA, la legge attiva infine un programma di investimenti da 1 miliardo di euro a favore di startup e PMI attive in IA, cybersicurezza e tecnologie emergenti, con l'obiettivo di sostenere trasferimento tecnologico e filiere strategiche.

I punti chiave della legge sono:

- **Principi e tutele:** uso antropocentrico e sicuro; trasparenza, accessibilità, protezione dei dati e della riservatezza.
- **Ambiti di applicazione:** sanità (ruolo centrale del medico e uso dei dati per ricerca), lavoro (dignità del lavoratore e osservatorio), Pubblica Amministrazione (PA) e giustizia (supporto decisionale tracciabile), scuola e sport (formazione, inclusione).

¹¹⁸ <https://www.acn.gov.it>

¹¹⁹ <https://www.agid.gov.it>

- **Governance nazionale:** ACN e AgID come Autorità competenti; coordinamento presso la Presidenza del Consiglio; Strategia IA aggiornata ogni due anni e relazione annuale al Parlamento.
- **Sostegno all'innovazione:** 1 miliardo di euro per startup e PMI in IA e tecnologie correlate.

In altre parole, con questa legge l'Italia si dota di una cornice stabile che punta a conciliare adozione rapida e regole chiare, per portare l'IA dentro i settori chiave del Paese con responsabilità, sicurezza e tutela dei cittadini, in linea con le migliori pratiche europee e valorizzando competenze e imprese nazionali.

Aspetti applicativi

Il testo evidenzia obblighi e adeguamenti immediati soprattutto nel lavoro, nella PA e nel diritto d'autore, con ricadute operative di governance, formazione e sicurezza.

1) Lavoro (Art. 11)

L'IA va usata per migliorare condizioni e prestazioni dei lavoratori, secondo un approccio antropocentrico e rispettoso dei diritti.

Il datore di lavoro ha l'obbligo di informare i lavoratori quando usa IA, richiamando alle modalità già previste dalla L. 152/1997.

Se l'IA è impiegata per gestire personale (selezione, valutazione, turni, performance, ecc.), servono adeguamenti tipici di compliance:

- aggiornare informativa privacy e registro dei trattamenti;
- aggiornare l'eventuale DPIA¹²⁰ (valutazione d'impatto ex GDPR¹²¹ art. 35) se già svolta;

¹²⁰ La valutazione d'impatto (DPIA – Data Protection Impact Assessment) è uno strumento previsto dall'art. 35 del GDPR, utile per delineare un trattamento di dati e per valutare la necessità, la proporzionalità e i relativi rischi.

¹²¹ Il GDPR (General Data Protection Regulation) è il regolamento europeo

- attenzione alle implicazioni di controllo a distanza (Statuto dei Lavoratori, art. 4).

A tal proposito, è stato attivato a Dicembre 2025 un Osservatorio sul Mondo del Lavoro presso il Ministero¹²².

2) Professionisti (Art. 13)

Introduce regole per avvocati, commercialisti, consulenti del lavoro, notai: obiettivo trasparenza, tutela del pensiero critico umano e del rapporto fiduciario col cliente.

L'IA deve supportare e non sostituire il giudizio professionale.

Obbligo di informare il cliente sull'uso di sistemi automatizzati, chiarendo finalità e limiti.

Sono richiesti principi di responsabilità e proporzionalità, l'IA non deve condizionare indebitamente le decisioni.

Previsti obblighi formativi per i professionisti, al fine di capire strumenti e rischi. In caso di danni da uso improprio, la responsabilità resta al professionista, che deve poter giustificare scelte e decisioni.

Per studi legali e notarili risulta necessario l'aggiornamento delle procedure, dei controlli, della sicurezza dei dati e della qualità del servizio.

3) Pubblica Amministrazione (Art. 14)

L'IA è confermata come supporto all'azione amministrativa, ma la responsabilità resta al funzionario/RUP, sia con decisione algoritmica sia con strumenti di mero supporto.

Se c'è decisione algoritmica, richiamo ai principi amministrativi: conoscibilità, non esclusività (decisione non totalmente delegata), non discriminazione.

La PA deve attivare misure tecniche, organizzative e formative: non solo cybersecurity (richiami a NIS2¹²³ recepita e normativa

sulla protezione dei dati personali, entrato in vigore nel 2018, che armonizza le leggi sulla privacy nell'UE, protegge i dati dei cittadini europei e stabilisce regole più rigorose per la raccolta, l'elaborazione e la conservazione dei dati personali da parte di aziende e organizzazioni, sia all'interno che all'esterno dell'UE, garantendo maggiore controllo agli individui e prevedendo sanzioni severe per le violazioni.

¹²² <https://shorturl.at/oSbvC>

¹²³ NIS2 (Network and Information Security Directive 2) è una nuova

nazionale), ma anche formazione su strumenti e su gestione dell'intero ciclo dei dati (archivi e documenti inclusi).

Sono annunciate linee guida AgID sull'adozione dell'IA nella PA, che enfatizzano la necessità di un aggiornamento continuo.

4) Diritto d'Autore (Art. 25)

Modifica la disciplina per:

- tutelare le opere dell'ingegno anche quando create con ausilio di IA;
- introdurre una tutela differenziata anche per le fonti.

Questo articolo ha un impatto trasversale non solo per l'editoria, ma anche attività quotidiane come video, slide, testi.

Criticità principali

Uno dei punti critici di questa legge è l'aspetto dei finanziamenti. La legge viene descritta come indirizzo senza risorse immediate; si citano possibili dotazioni future (esteri, cybersecurity, quantistico). Il rischio è che l'innovazione rimanga soltanto un'idea senza finanziamenti adeguati.

Un altro punto critico è la delega al Governo con molte parti del testo che demandano a decreti futuri, con il rischio di un impianto ancora incompleto o velleitario.

Tra i temi da disciplinare:

- dati e algoritmi per addestramento;
- inibizioni e sanzioni;
- poteri ispettivi;
- banking/finanza/pagamenti;
- alfabetizzazione e formazione (scuola e alta formazione);
- processo e diritto di difesa.

La legge tocca settori sensibili come sanità e ricerca, giustizia, difesa, sviluppo economico e introduce profili penali (es. il deepfake). Il documento nota che il Garante Privacy appare meno centrale (salvo richiami sui dati particolari), mentre in altri

direttiva dell'Unione Europea che rafforza le norme sulla cybersicurezza per proteggere le infrastrutture digitali critiche.

Paesi si tende a concentrare la responsabilità su un'unica autorità indipendente.

Oltre alla legge nazionale, restano gli obblighi dell'AI Act e una scadenza già citata nel documento (2 agosto 2025).

Tra gli obblighi utili da integrare nelle organizzazioni: alfabetizzazione/AI literacy, da inserire in strumenti di governance come MOG 231¹²⁴ o PIAO¹²⁵ (dove presenti), più un regolamento interno per dipendenti.

Più complessa la mappatura dei processi e la costruzione di una vera cultura del dato. L'intelligenza artificiale non è un semplice aggiornamento tecnologico e richiede coerenza, competenze e implementazione concreta nelle organizzazioni con nuove sfide etiche da affrontare.

¹²⁴ Il MOG 231 (Modello di Organizzazione, Gestione e Controllo) è un insieme di regole, procedure e protocolli previsti dal Decreto Legislativo n. 231/2001 italiano, adottato volontariamente dalle aziende per prevenire la commissione di reati da parte di dipendenti o apicali.

¹²⁵ Il PIAO (Piano Integrato di Attività e Organizzazione) è un documento unico e fondamentale per le Pubbliche Amministrazioni italiane.

10. L'IA DEL FUTURO

L'intelligenza artificiale sta entrando in una fase in cui smette di essere considerata solo una tecnologia, e diventa una infrastruttura, un po' come l'elettricità o internet. La vediamo nei motori di ricerca, negli strumenti di scrittura, nei sistemi di diagnosi, nelle catene logistiche, nelle piattaforme che decidono cosa leggiamo e cosa compriamo. La novità, rispetto alle ondate precedenti, è che l'IA generativa ha reso visibile a tutti un tipo di potere che prima restava dietro le quinte, la capacità di produrre testi, immagini, codice e sintesi, e di farlo in modo abbastanza fluido da sembrare competenza. Questa sensazione, a volte, è più importante della realtà tecnica, perché guida l'adozione, sposta investimenti, cambia aspettative, crea pressioni competitive. Quando una tecnologia diventa narrazione collettiva, il suo futuro non dipende solo da ciò che può fare, ma da ciò che il mondo decide di farle fare, e da quanto capitale, lavoro, energia, fiducia sociale, le viene consegnato.

In questo scenario, parlare di futuro significa distinguere tre linee che si intrecciano, una linea economica e finanziaria, una linea del lavoro e della distribuzione del valore, una linea scientifica e filosofica, cioè che cosa capiamo davvero dell'intelligenza, e che tipo di sistemi vogliamo costruire. È

proprio qui che le posizioni dei professori Yann LeCun, Geoffrey Hinton e Luciano Floridi diventano complementari, perché puntano ciascuno un riflettore su un punto cieco diverso, e insieme raccontano la stessa storia, l'IA non è solo un prodotto, è un nuovo ambiente.

I limiti dei grandi modelli linguistici

Il successo dei grandi modelli linguistici ha creato una tentazione, scambiare la competenza linguistica per comprensione del mondo. Il Prof. Yann LeCun, che da anni guida una linea di ricerca alternativa, insiste sul fatto che prevedere la prossima parola non equivale a costruire una rappresentazione causale e utilizzabile della realtà. In un suo famoso articolo¹²⁶ sulla strada verso macchine più autonome, LeCun descrive la necessità di agenti capaci di apprendere come animali e umani, con la capacità di ragionare e pianificare, e propone un'architettura che combina un modello predittivo del mondo, motivazioni intrinseche, e una famiglia di approcci chiamata Joint Embedding Predictive Architecture (JEPA), pensata per imparare rappresentazioni astratte attraverso apprendimento auto supervisionato.

L'idea di un World Model è per LeCun cruciale. In un recentissimo articolo LeCun e altri ricercatori esaminano i limiti degli attuali modelli di intelligenza artificiale nel raggiungere un apprendimento davvero autonomo, proponendo un'architettura di apprendimento ispirata al modo in cui imparano esseri umani e animali. Il modello proposto unisce due modi di apprendere: da una parte l'apprendimento attraverso l'osservazione (Sistema A), dall'altra l'apprendimento attraverso l'azione diretta e l'esperienza pratica (Sistema B). Questi due approcci possono alternarsi in modo flessibile grazie a segnali interni di controllo e coordinamento (Sistema M). L'idea è quella si spiegare come questo modello potrebbe essere sviluppato prendendo spunto da come gli organismi si adattano ad ambienti reali e in

¹²⁶ Yann LeCun, *A Path Towards Autonomous Machine Intelligence*, Courant Institute of Mathematical Sciences, New York University, 2022, <https://openreview.net/pdf?id=BZ5a1r-kVsf>

continuo cambiamento, sia nel corso dell'evoluzione sia durante la loro crescita.

Un agente di intelligenza artificiale davvero utile non deve solo parlare bene, deve anticipare conseguenze, capire permanenza degli oggetti, causalità fisica, intenzioni degli altri agenti, deve poter fare piani a più orizzonti temporali, deve avere memoria operativa, deve distinguere ciò che è possibile da ciò che è solo grammaticalmente plausibile. Per LeCun, una parte della risposta è imparare non tanto a generare pixel o token, quanto a prevedere in uno spazio di rappresentazioni, cioè a fare previsioni interne su come il mondo potrebbe evolvere. I lavori su JEPA mostrano un approccio non generativo per imparare rappresentazioni visive predicendo parti mancanti dell'immagine a livello di embedding, con l'obiettivo di catturare semantica e struttura, non solo dettagli superficiali. In modo analogo, filoni come VJEPA¹²⁷ cercano di portare questo paradigma al video, puntando a una comprensione temporale che avvicini l'IA alle regolarità del mondo fisico e dell'azione.

La conseguenza, per il futuro, è che potremmo vedere una biforcazione. Da una parte LLM sempre più integrati in prodotti, utili come interfaccia universale, bravi a scrivere, sintetizzare, conversare. Dall'altra parte sistemi più immersi nella realtà, multimodali, capaci di apprendere da video, da sensori, da interazione, più vicini a un'idea di intelligenza come capacità di agire, non solo di descrivere. In questa prospettiva, chiedersi se l'intelligenza generale possa emergere semplicemente scalando i modelli linguistici non è soltanto una questione tecnica, ma una vera e propria scommessa scientifica. Ed è proprio qui che LeCun richiama l'attenzione su un punto decisivo: questa scommessa potrebbe anche rivelarsi sbagliata.

¹²⁷ L'approccio V-JEPA di LeCun è un modo di addestrare l'IA a capire cosa sta succedendo in un video senza dover ricostruire ogni pixel. La sigla significa Video Joint-Embedding Predictive Architecture. In pratica, invece di chiedere al modello di generare il futuro fotogramma, gli si chiede di prevedere, in uno spazio interno di rappresentazioni, cosa è probabile che accada nelle parti mancanti o future della scena.

Potenza, rischio, controllo dei sistemi IA

Se LeCun mette in guardia dall'illusione linguistica, Geoffrey Hinton tende a prendere più sul serio la traiettoria di potenza dei sistemi attuali. Nel 2024 il Nobel per la Fisica è stato assegnato congiuntamente a John J. Hopfield e Geoffrey Hinton “per scoperte e invenzioni fondamentali che rendono possibile il machine learning con reti neurali artificiali”, riconoscendo quanto la storia dell'IA moderna sia intrecciata con idee nate anche nella fisica statistica.

Hinton, però, non si limita a celebrare il progresso. Nella sua Nobel Banquet Speech ha elencato rischi concreti già presenti e rischi imminenti, dalla creazione di camere dell'eco e manipolazione dell'informazione, all'uso per sorveglianza di massa, phishing e cybercrimine, fino alla possibilità di armi autonome, e persino di strumenti capaci di progettare agenti patogeni o armi sempre più letali¹²⁸. Il cuore della sua preoccupazione è il controllo, non nel senso fantascientifico di un robot ribelle, ma nel senso politico e tecnico di sistemi che diventano così utili e così diffusi da rendere difficile limitarli, auditarli, spegnerli, o anche solo capire davvero perché fanno ciò che fanno. Su questo tema, preoccupa l'atteggiamento delle Big Tech, come OpenAI che ha recentemente siglato un accordo con il dipartimento della Difesa statunitense per l'utilizzo delle sue tecnologie di IA¹²⁹.

Questa non è una disputa tra ottimisti e pessimisti, è una disputa su quali rischi sono più urgenti. Hinton tende a enfatizzare la possibilità che le capacità emergenti superino le nostre difese istituzionali, LeCun tende a enfatizzare che manca ancora un'architettura capace di senso comune e pianificazione robusta. Bisogna tenere conto di entrambe queste prospettive, così da evitare atteggiamenti poco consapevoli o eccessivamente fatalisti.

¹²⁸ https://en.wikipedia.org/wiki/Artificial_intelligence_arms_race

¹²⁹ <https://www.ilpost.it/2026/02/28/accordo-openai-difesa-stati-uniti/>

Il futuro dell'IA come problema di governance

Il Prof. Luciano Floridi porta il discorso su un piano diverso, non si interroga solo su cosa possa fare l'IA, ma sulla trasformazione dell'uomo sommerso in un ambiente digitale in continua trasformazione. In un lavoro sul futuro prossimo dell'IA, Floridi propone di interpretare questi sistemi non come una copia dell'intelligenza biologica, ma come una separazione tra agency e intelligence, cioè la capacità di ottenere risultati e la necessità di essere intelligenti per ottenerli, una macchina può agire con successo senza la capacità di comprendere, nel senso umano del termine. Questa idea è liberatoria e inquietante allo stesso tempo, liberatoria perché riduce l'antropomorfismo e ci aiuta a progettare strumenti, inquietante perché sposta l'attenzione sulla responsabilità, se un agente artificiale fa accadere cose nel mondo, chi risponde, chi decide obiettivi, chi controlla i contesti d'uso.

Floridi insiste anche sul fatto che l'etica dell'IA non può essere solo una lista di buone intenzioni, e introduce la distinzione tra hard ethics, che contribuisce a plasmare norme e leggi, e soft ethics, che si applica dopo la conformità legale, definendo lo spazio delle scelte morali responsabili dentro ciò che è permesso¹³⁰. È un modo elegante per dire che la legge è necessaria ma non sufficiente, anche quando un uso dell'IA è legale, può essere comunque sbagliato, dannoso, o socialmente corrosivo. E qui torna il tema del lavoro, un'azienda può legalmente automatizzare, ma può farlo in un modo che umilia, sorveglia, precarizza, o in un modo che sostiene transizioni, riqualifica, riduce orari, protegge dignità e autonomia.

Nel futuro che Floridi immagina, la domanda da porsi non è se useremo l'IA oppure no, perché la stiamo già utilizzando, ma chi controllerà chi. Chi controllerà l'IA, chi ne diventerà dipendente, chi avrà accesso a competenze e dati, chi sarà costretto a subire decisioni automatizzate senza trasparenza.

¹³⁰ Luciano Floridi, *Soft Ethics and the Governance of the Digital*, Philosophy & Technology, Volume 31, pag. 1-8, 2018, <https://link.springer.com/article/10.1007/s13347-018-0303-9>

Questa è una questione di cittadinanza nell'infosfera¹³¹, più che una questione di strumento tecnologico.

Lavoro, quali mestieri a rischio

Quando si parla di lavori a rischio, bisogna evitare due errori simmetrici, il primo è immaginare un'apocalisse occupazionale uniforme, il secondo è pensare che non cambierà quasi nulla. L'IA non colpisce i settori, colpisce le attività all'interno dei settori. Per questo il confine non è tra lavori manuali e lavori intellettuali, ma tra compiti ripetitivi e compiti situati, tra produzione standardizzata e relazione umana, tra decisioni basate su testi e procedure e decisioni che richiedono responsabilità concreta in un contesto reale.

I compiti più esposti sono quelli che assomigliano a una trasformazione di informazioni, riassumere documenti, redigere comunicazioni standard, compilare report, gestire ticket, produrre bozze di contratti, fare ricerche preliminari, tradurre testi non specialistici, scrivere contenuti di marketing a basso valore, generare codice di routine, preparare presentazioni, classificare richieste. In molti uffici, questi compiti sono una parte significativa del lavoro di segreteria evoluta, back office, customer care, amministrazione, contabilità di base, supporto legale e compliance, analisi junior, redazione. Non significa che queste professioni spariranno, significa che la loro "parte automatizzabile" diventerà più grande, e che la competenza richiesta si sposterà verso controllo di qualità, supervisione, gestione dei casi anomali, relazione con clienti e colleghi, responsabilità delle scelte.

Un'implicazione etica spesso sottovalutata è la qualità dell'occupazione. Se l'IA prende i compiti più semplici, le persone junior rischiano di perdere la palestra dove imparavano, e i senior rischiano di diventare supervisori di flussi automatizzati, con un lavoro più veloce ma anche più alienante.

¹³¹ Per Floridi, l'infosfera è l'ambiente complessivo costituito da tutte le entità, relazioni, processi e scambi informativi, sia online sia offline. In altre parole, è lo spazio in cui vivono e interagiscono dati, tecnologie, esseri umani e agenti artificiali.

Il potere contrattuale può spostarsi verso chi controlla i sistemi e i dati, lasciando altri in una posizione di sostituibilità. Se poi il risparmio di costi finisce soprattutto in profitti e non in redistribuzione, formazione, riduzione dell'orario, nasce un problema di giustizia sociale, non di tecnologia.

Geoffrey Hinton ha parlato esplicitamente del rischio che l'IA produca forti squilibri, con profitti che si concentrano e con molte persone che possono essere spinte fuori dal mercato o verso lavori peggiori, e in generale ha insistito sul fatto che la velocità del cambiamento rende difficile per le società adattarsi con gradualità. La questione etica, quindi, non è solo quanti posti, ma che tipo di vita lavorativa, chi decide, chi beneficia, chi paga i costi della transizione.

Il rischio di una “bolla dell’IA”

Ogni grande trasformazione tecnologica porta con sé un doppio movimento, l'innovazione reale e la sua versione finanziaria. La seconda è fatta di aspettative, di storytelling, di corsa alla posizione, di aziende che promettono, di investitori che inseguono, di mercati che si concentrano. Negli ultimi anni l'IA ha alimentato una forte crescita di alcune grandi società tecnologiche, e istituzioni come la Banca dei Regolamenti Internazionali hanno osservato che il boom legato all'IA ha continuato a influenzare i mercati azionari, con segnali di cautela quando le valutazioni appaiono distorte e la fiducia può cambiare rapidamente.

Il punto non è decidere se siamo in una bolla poiché queste si riconoscono davvero solo quando scoppiano. Il punto è capire la struttura del rischio di una possibile bolla. Da un lato ci sono investimenti giganteschi in infrastrutture, data center, energia, rete, chip, e la finanza tende a premiare chi sembra più vicino al rubinetto dell'IA. Dall'altro lato c'è la monetizzazione, cioè quanto valore economico stabile si riesce a estrarre da questi sistemi, e in quali settori, e con che margini. Analisi di mercato hanno sottolineato che l'aumento delle azioni legate all'IA ha riaccessato timori di dinamiche speculative, e hanno evidenziato l'ampiezza della spesa in data center, con cifre dell'ordine di

centinaia di miliardi e previsioni di ulteriore crescita, insieme al rischio di sovrainvestimento, la difficoltà di monetizzare e l'eccessiva concentrazione possano rendere fragile il quadro se i fondamentali peggiorano.

Un elemento tipico delle fasi euforiche è che l'investimento precede la domanda, si costruisce capacità prima di sapere se verrà usata con profitto. E' emblematico l'esempio della startup Thinking Machine¹³² fondata a Febbraio 2025 da Mira Murati, ex CTO di OpenAI, che ha ricevuto un finanziamento di 2 miliardi di dollari senza avere un business plan ben definito¹³³.

Reuters ha riportato la preoccupazione di molti gestori sul rischio di indigestione da investimenti e sull'incertezza dei ritorni, con l'idea che una parte del mercato tema una sovraspesa che ricorda, per dinamica, la stagione dot com, pur in un contesto diverso. Anche il Fondo Monetario Internazionale, in dichiarazioni riportate da Reuters, ha richiamato la possibilità che un boom di investimenti possa sgonfiarsi, senza per forza trasformarsi in una crisi sistemica globale, ma con effetti rilevanti su aziende e mercati troppo esposti¹³⁴.

Se una bolla dovesse materializzarsi, il danno non sarebbe solo borsistico. Potrebbe produrre una reazione opposta, tagli improvvisi, progetti cancellati, riduzione della ricerca, licenziamenti nel settore, sfiducia verso applicazioni utili. Il futuro dell'IA non dipende solo dalla qualità dei modelli, ma anche dalla stabilità del ciclo di investimento che li sostiene, e dalla capacità di trasformare spesa in infrastruttura in valore sociale, non solo in valore azionario.

Intelligenza artificiale generale (AGI)

“L'intelligenza artificiale (IA) riprodurrà l'intelligenza umana. L'IA sconfiggerà le malattie. L'IA è la più grande e più

¹³² <https://thinkingmachines.ai/>

¹³³ <https://shorturl.at/d8z3l>

¹³⁴ Mike Dolan, *Global funds fear AI investment indigestion*, Reuters, 2025, <https://www.reuters.com/markets/europe/global-funds-fear-ai-investment-indigestion-2025-11-19/>

importante invenzione nella storia dell'umanità. Lo avete sentito ripetere mille volte ma probabilmente nulla di tutto ciò è vero". Si apre così il numero speciale intitolato Hype Correction, con il sottotitolo *It's time to reset expectations*, pubblicato nel dicembre 2025 dalla MIT Technology Review, la rivista del Massachusetts Institute of Technology (MIT), considerata una delle fonti più autorevoli nel panorama della ricerca scientifica e tecnologica negli Stati Uniti e a livello internazionale.

Sebbene i recenti progressi dei sistemi di IA generativa, in particolare in ambiti come lo sviluppo software, siano sorprendenti, l'ipotesi dell'imminente arrivo di un'intelligenza artificiale generale (AGI) appare, nel complesso, ancora poco credibile.

Una parte della comunità scientifica interpreta l'evoluzione degli LLM come il segnale che, aumentando scala e capacità, emergano competenze sempre più generali.

Un riferimento è il lavoro *Sparks of Artificial General Intelligence: Early experiments with GPT-4*¹³⁵ di Bubeck, Horvitz, Kamar e colleghi, che descrive prestazioni ampie e trasversali e propone che modelli come GPT possano essere visti come una forma precoce ma incompleta di AGI, pur riconoscendo limiti importanti e la possibile necessità di andare oltre la sola predizione della parola successiva.

Altri studiosi ritengono che i LLM, per come sono costruiti oggi, manchino di ingredienti essenziali per parlare seriamente di AGI.

Oltre al già citato Prof. Yann LeCun che sostiene l'assenza di un modello del mondo indispensabile per ottenere una forma di intelligenza ci sono anche altri articoli come quello di François Chollet, ricercatore presso Google e vincitore dell'ARC Prize¹³⁶, che contesta l'idea che più scala equivalga a più intelligenza. Per lui, gran parte delle performance attuali misura abilità addestrate su distribuzioni note, non intelligenza intesa come adattamento efficiente a situazioni nuove e incerte. La sua critica è legata

¹³⁵ <https://arxiv.org/pdf/2303.12712>

¹³⁶ La ARC Prize Foundation è un'organizzazione senza scopo di lucro con la missione di orientare ricercatori, industria e autorità di regolamentazione verso l'intelligenza artificiale generale attraverso benchmark solidi e duraturi.

proprio al tema della generalizzazione fuori distribuzione, considerata cruciale per l'AGI¹³⁷.

Diversi critici insistono su difetti strutturali che oggi vediamo anche empiricamente: incoerenze, fragilità ai cambi di contesto, difficoltà con ragionamento robusto e novità. Gary Marcus¹³⁸, ad esempio, ha ripetutamente sostenuto che la somiglianza con l'output umano non implica la stessa architettura cognitiva né affidabilità generalista, e che gli LLM non siano AGI¹³⁹.

Anche la Prof.ssa Melanie Mitchell, docente al Santa Fe Institute, ha discusso la questione dell'apparente robustezza del ragionamento negli LLM, sottolineando come molti risultati dipendono fortemente dai benchmark e non garantiscano generalità stabile¹⁴⁰.

Tra ottimismo e scetticismo c'è un'area di convergenza crescente: gli LLM potrebbero essere una parte di un sistema più ampio, ma non l'AGI da soli.

Un esempio spesso citato è quello di Demis Hassabis, Premio Nobel per Chimica nel 2024 e cofondatore di Google DeepMind¹⁴¹, che ha sottolineato come i sistemi attuali non abbiano una comprensione robusta di fisica, causalità e conseguenze delle azioni nel tempo; secondo questa visione, servono capacità aggiuntive (ragionamento, pianificazione gerarchica, memoria, coerenza) e, soprattutto, qualche forma di modello del mondo che vada oltre la sola manipolazione del linguaggio.

La domanda se gli LLM porteranno all'AGI è meno semplice di quanto sembri perché contiene almeno tre questioni diverse:

- Che cos'è AGI, in modo operativo e misurabile?
- Gli LLM mostrano generalità vera o generalità apparente dovuta a dati e benchmark?
- Quanto di ciò che manca può essere aggiunto con

¹³⁷ <https://time.com/collections/time100-ai-2024/7012823>

¹³⁸ Gary Marcus è uno psicologo cognitivo e scienziato statunitense, professore emerito alla New York University, noto per i suoi studi sul linguaggio, sull'intelligenza artificiale e sui limiti dei modelli di deep learning.

¹³⁹ <https://garymarcus.substack.com/p/rumors-of-agis-arrival-have-been>

¹⁴⁰ <https://aiguide.substack.com/p/the-llm-reasoning-debate-heats-up>

¹⁴¹ <https://deepmind.google/>

scalabilità più integrazione, e quanto richiede un cambio di paradigma?

Oggi, la letteratura e le posizioni degli studiosi indicano che non esiste consenso. C'è chi vede negli LLM scintille di generalità e chi ritiene che, senza un World Model e nuove architetture, la via sia sbagliata.

La traiettoria più condivisa, sembra essere quella dei sistemi composti. L'AGI, se arriverà, potrebbe assomigliare meno a un unico modello e più a un insieme integrato di moduli come linguaggio, percezione, memoria, pianificazione, modelli del mondo, con capacità di apprendimento e controllo più profonde.

Stiamo vivendo un momento storico segnato da profondi cambiamenti tecnologici, che stanno modificando il modo in cui lavoriamo, comunichiamo, studiamo e prendiamo decisioni. In altre parole, questa tecnologia sta trasformando la nostra società. Spetta a tutti noi scegliere se e come utilizzarla. L'unico strumento che abbiamo per cercare di governare l'intelligenza artificiale è la conoscenza. Senza comprendere come funziona, quali sono i suoi limiti e quali effetti collaterali può produrre, rischiamo di prendere decisioni sbagliate.

BIBLIOGRAFIA

- Pierluigi Adami, *Oltre l'umano. Intelligenza artificiale e futuro della civiltà*, Bordeaux, 2025, ISBN 9791259632500
- Anil Ananthaswamy, *Perché le macchine imparano. L'eleganza della matematica dietro all'AI*, Apogeo, 2024, 9788850337385
- Daniel Andler, *Il duplice enigma. Intelligenza artificiale e intelligenza umana*, Einaudi, 2024, ISBN 9788806263430
- Paolo Benanti, *Human in the loop. Decisioni umane e intelligenze artificiali*, Mondadori, 2022, ISBN 9791220600477
- Paolo Benanti, *Oracoli. Tra algoretica e algocrazia*, Luca Sossella editore, 2018, ISBN 9788897356905
- Nick Bostrom, *Superintelligenza. Tendenze, pericoli, strategie*, Bollati Boringhieri, 2023, ISBN 9788833939964
- Paolo Caressa, *Ignoranza artificiale. Quello che le macchine non sanno*, Apogeo, 2025, ISBN 9788850339297
- Massimo Chiriatti, *Incoscienza artificiale. Come fanno le macchine a prevedere per noi*, Luiss University Press, 2021, ISBN 9788861056312
- Massimo Chiriatti, *#Humanless. L'algoritmo egoista*, Hoepli, 2019, ISBN 9788820390921
- Qiufan Chen, Kai-Fu Lee, *Ai 2041. Scenari dal futuro dell'intelligenza artificiale*, Luiss University Press, 2023, ISBN 9788861058606
- Andrea Colamedici, Simone Arcagni, *L'algoritmo di Babele. Storie e miti dell'intelligenza artificiale*, Solferino, 2024, ISBN 978-8828215813
- Kate Crawford, *Né intelligente né artificiale. Il lato oscuro dell'IA*, Il Mulino, 2021, ISBN 9788815294197
- Nello Cristianini, *Machina sapiens. L'algoritmo che ci ha rubato il segreto della conoscenza*, Il Mulino, 2024, ISBN 9788815384461
- Nello Cristianini, *Sovrumano. Oltre i limiti della nostra intelligenza*, Il Mulino, 2025, ISBN 9788815392107
- Pedro Domingos, *L'algoritmo definitivo. La macchina che*

impara da sola e il futuro del nostro mondo, Bollati Boringhieri, 2016, ISBN 9788833927060

- Alfio Ferrara, *Le macchine del linguaggio. L'uomo allo specchio dell'intelligenza artificiale*, Einaudi, 2025, ISBN 9788806268428
- Giusella Finocchiaro, *Intelligenza artificiale. Quali regole?*, Il Mulino, 2024, ISBN 9788815388353
- Luciano Floridi, *Etica dell'intelligenza artificiale. Sviluppi, opportunità, sfide*, Raffaello Cortina Editore, 2022, ISBN 9788832854091
- Luciano Floridi, *La differenza fondamentale. Artificial Agency: una nuova filosofia dell'intelligenza artificiale*, Mondadori, 2025, ISBN 9788804805854
- Benedetta Giovanola, *Etica e intelligenza artificiale per l'impresa*, ROI Edizioni, 2025, ISBN 9788836202492
- Francesco D'Isa, *La rivoluzione algoritmica. Arte e intelligenza artificiale*, Luca Sossella Editore, 2024, ISBN 9791259980601
- Jerry Kaplan, *Generative A.I. Conoscere, capire e usare l'intelligenza artificiale generativa*, Luiss University Press, 2024, ISBN 9791255960911
- Henry Kissinger, Daniel Huttenlocher, Eric Schmidt, *L'era dell'intelligenza artificiale. Il futuro dell'identità umana*, Mondadori, 2023, ISBN 9788804752950
- George Lakoff, Srini Narayanan, *La mente neurale. Come pensa il nostro cervello*, ROI edizioni, 2025, ISBN 9788836202508
- Paolo Legrenzi, *L'intelligenza del futuro. Perché gli algoritmi non ci sostituiranno*, Mondadori, 2024, ISBN 9788804783060
- Riccardo Manzotti, Simone Rossi, *IO & IA. Mente, cervello e GPT*, Rubbettino, 2023, ISBN 9788849876642
- Melanie Mitchell, *L'intelligenza artificiale. Una guida per esseri umani pensanti*, Einaudi, 2022, ISBN 9788806253301
- Niccolò Monti, *Prompting. Poetiche e politiche dell'intelligenza artificiale*, Tlon, 2025, ISBN 9791255540892

- Enrico Panai, *L'etica dell'intelligenza artificiale spiegata a mio figlio*, Mimesis, 2024, ISBN 9791222312194
- Giorgio Parisi, *Le simmetrie nascoste. Perché la fisica è alle radici dell'intelligenza artificiale di oggi e di domani*, Rizzoli editore, 2026, ISBN 9788817181440
- Alfio Quarteroni, *Algoritmi per un nuovo mondo*, Edizioni Dedalo, 2021, ISBN 9788822016089
- Alfio Quarteroni, *L'intelligenza creata. L'AI e il nostro futuro*, Hoepli, 2025, ISBN 9788836020089
- Stefano Quintarelli (curatore), *Intelligenza artificiale. Cos'è davvero, come funziona, che effetti avrà*, Bollati Boringhieri, 2020, ISBN 9788833935423
- Francesca Rossi, *Intelligenza artificiale. Come funziona e dove ci porta la tecnologia che sta trasformando il mondo*, Laterza, 2024, ISBN 9788858155394
- Dennis Yi Tenen, *Teoria letteraria per robot. Come i computer hanno imparato a scrivere*, Bollati Boringhieri, 2024, ISBN 9788833943299
- Alan Turing, *Intelligenza meccanica*, Bollati Boringhieri, 1994, ISBN 9788833908809
- Gennaro Vessio, *Intelligenza artificiale per curiosi. Una mini-guida scientifica per tutti*, Edizioni Dedalo, 2025, ISBN 9788822069276
- Guido Vetere, *Intelligenze aliene. Linguaggio e vita degli automi*, Luca Sossella editore, 2025, ISBN 9791259980878
- Enrico Zimuel, *Strumenti di informatica. Un approccio moderno per corsi universitari*, Tecniche Nuove, 2025, ISBN 9788848147613

CREDITI SULLE IMMAGINI

Figura 1, Chess Programming Wiki,
<https://www.chessprogramming.org/File:JohnMcCarthy.jpg>

Figura 2, Wikipedia,
https://en.wikipedia.org/wiki/Turing_test#/media/File:Turing_test_diagram.png

Figura 3, Wikipedia,
https://it.wikipedia.org/wiki/File:Theseus_Maze_by_Claude_Shannon,_1952_-_MIT_Museum_-_DSC03702.JPG

Figura 4, Wikipedia,
https://en.wikipedia.org/wiki/Perceptron#/media/File:Organization_of_a_biological_brain_and_a_perceptron.png

Figura 5, Wikipedia,
[https://en.wikipedia.org/wiki/Perceptron#/media/File:330-PSA-80-60_\(U_SN_710739\)_ \(20897323365\).jpg](https://en.wikipedia.org/wiki/Perceptron#/media/File:330-PSA-80-60_(U_SN_710739)_ (20897323365).jpg)

Figura 6, Fermo immagine del video di Yann LeCun su
https://www.youtube.com/watch?v=FwFduRA_L6Q

Figura 7, Wikipedia,
[https://it.wikipedia.org/wiki/ELIZA_\(chat_bot\)#/media/File:ELIZA_con_versation.png](https://it.wikipedia.org/wiki/ELIZA_(chat_bot)#/media/File:ELIZA_con_versation.png)

Figura 8, IBM Deep Blue, <https://www.ibm.com/history/deep-blue>

Figura 9, PressKit del documentario AlphaGo,
<https://www.alphagomovie.com/alphago-press-kit.zip>

Figura 10, Diagnosi automatizzate: dalla COVID-19 ai tumori al seno, Scienza in rete,
<https://www.scienzainrete.it/articolo/diagnosi-automatizzate-dalla-covid-19-ai-tumori-al-seno/chiara-sabelli/2020-10-23>

Figura 11, Screenshot della pagina ChatGPT, <https://chatgpt.com/>

Figure 12, 13, 15, 17, 18, 19, 20, 25, 26, 27, 28, 29, 30, 32, 38 realizzate da Enrico Zimuel

Figura 14, Shuvendu Roy; Md. Sakif Rahman, Emergency Vehicle Detection on Heavy Traffic Road from CCTV Footage Using Deep Convolutional

Neural Network, <https://ieeexplore.ieee.org/document/8679295>

Figura 16, Rielaborazione dell'immagine su Wikipedia, https://it.wikipedia.org/wiki/Cono_d'emergenza#/media/File:Neuron_Hand-tuned2.svg

Figura 21, Wikimedia, <https://commons.wikimedia.org/w/index.php?curid=45679374>

Figura 22, Adam Lineberry, Getting Up to Speed with SHAP for Model Interpretability, <https://adamlineberry.io/shap/>

Figura 23, Haifeng Zhang et al., Weakly Supervised Local-Global Attention Network for Facial Expression Recognition, https://www.researchgate.net/figure/Attention-maps-that-indicate-crucial-facial-regions-For-a-given-input-image-left_fig1_339477370

Figura 24, Jiawei Su et al., One Pixel Attack for Fooling Deep Neural Networks, <https://arxiv.org/pdf/1710.08864>

Figura 31, Research Gate, Illustration of the Transformer architecture, https://www.researchgate.net/figure/Illustration-of-the-transformer-architecture-showcasing-the-encoder-decoder-structure-The_fig4_390063429

Figura 33, Wikimedia, LLM emergent benchmarks, https://commons.wikimedia.org/wiki/File:LLM_emergent_benchmarks.png

Figura 34, Wikicommons, Chat GPT in a Chinese room!, <https://www.craigbellamy.net/2023/01/15/chat-gpt-in-a-chinese-room/>

Figura 35, APEX-Agents benchmark, <https://arxiv.org/pdf/2601.14242>

Figura 36, Walter Quattrociocchi, Epistemological Fault Lines Between Human and Artificial Intelligence, <https://arxiv.org/pdf/2512.19466>

Figura 37, Immagine generata da Enrico Zimuel con GPT-5.2

Figura 39, AI Act, <https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai>

INFORMAZIONI SULL'AUTORE



Enrico Zimuel è docente di Intelligenza Artificiale e Machine Learning presso l'Università di Torino e docente di Agentic AI e Architetture RAG presso l'Università di Roma Tre.

Lavora da più di 15 anni nel settore della Ricerca e Sviluppo con aziende della Silicon Valley e si occupa di sviluppo software dal 1996. Ha lavorato come *Tech Lead* e *Principal Software Engineer* presso Elastic.

Ha sviluppato diversi progetti open source, come il progetto *Zend Framework* con più di 570 milioni di installazioni che nel 2010 ha vinto il premio InfoWorld Bossie Award.

Ha svolto attività di ricerca come *Research Programmer* presso l'Informatics Institute dell'Università di Amsterdam.

E' uno speaker TEDx e ha partecipato come relatore in più di 140 conferenze internazionali. Ha pubblicato diversi libri di informatica, tra i più recenti "*Strumenti di Informatica. Un approccio moderno per corsi universitari*" edito da Tecniche Nuove nel 2025.

Ha conseguito il Master in Data Analytics presso l'Università di Roma Tre e la laurea cum laude in Economia Informatica presso l'Università "G. D'Annunzio" di Chieti-Pescara. Si è specializzato presso la Brown University (USA) su algoritmi e automi cellulari sotto la guida del Prof. Stephen Wolfram.

Vive e lavora (da remoto) a Torino dal 2009. Per maggiori informazioni è possibile consultare il sito www.zimuel.it.