CORSO AI

Entriamo nel mondo delle AI

COME NASCONO QUESTE SLIDE

In risposta alla rapida diffusione di prodotti basati sull'Intelligenza Artificiale, ho elaborato una presentazione che ripercorre l'evoluzione di questa tecnologia e illustra i termini chiave utilizzati nel settore.

Nel corso della mia attività professionale, ho inoltre sperimentato diverse soluzioni AI che mi hanno permesso di ottimizzare i processi lavorativi, aumentando sia l'efficienza che la qualità dei risultati.

Ho quindi arricchito la presentazione con una sezione pratica dedicata ai vari strumenti AI, specificando per ciascuno il campo di applicazione ideale.

L'obiettivo di questo lavoro è duplice: da un lato, far conoscere i benefici concreti che l'Intelligenza Artificiale può apportare nella vita professionale, dall'altro, fornire una guida pratica per la scelta degli strumenti AI più adatti alle diverse esigenze lavorative quotidiane.

CHI SONO ?

Matteo Baccan è un ingegnere del software e formatore professionista con oltre 30 anni di esperienza nel settore IT.
Ha lavorato per diverse aziende e organizzazioni, occupandosi di progettazione, sviluppo, testing e gestione di applicazioni web e desktop, utilizzando vari linguaggi e tecnologie. È anche un appassionato divulgatore e insegnante di informatica, autore di numerosi articoli, libri e corsi online rivolti a tutti i livelli di competenza.
Gestisce un sito internet e un canale YouTube dove condivide video tutorial, interviste, recensioni e consigli sulla programmazione.
Attivo nelle community open source, partecipa regolarmente a eventi e concorsi di programmazione.
Si definisce un "sognatore realista" che ama sperimentare, innovare e condividere le sue conoscenze e passioni, seguendo il motto: "Non smettere mai di imparare, perché la vita non smette mai di insegnare".

COSA È L'INTELLIGENZA ARTIFICIALE?

L'**Intelligenza Artificiale** (AI) è una branca dell'informatica che abbraccia diverse discipline, tra cui l'apprendimento automatico, la visione artificiale, l'elaborazione del linguaggio naturale e la robotica.

Il suo scopo principale è sviluppare algoritmi che permettano ai computer di svolgere attività tradizionalmente eseguibili solo dall'intelletto umano. Si tratta di una scienza interdisciplinare che integra teoria, metodologia, analisi del linguaggio e ingegneria per creare sistemi intelligenti capaci di prendere decisioni articolate in scenari complessi.

Nel contesto aziendale, l'Intelligenza Artificiale si rivela uno strumento prezioso per ottimizzare i processi lavorativi, incrementando sia la produttività che l'efficienza operativa.

LE ORIGINI: LA CONFERENZA DI DARTMOUTH (1956)

L'Intelligenza Artificiale come disciplina scientifica venne formalizzata nel **1956** durante la storica **Conferenza di Dartmouth**, dove quattro pionieri gettarono le basi per l'automazione del pensiero:

- **John McCarthy**: coniò il termine "Intelligenza Artificiale"
- **Marvin Minsky**: pioniere delle reti neurali e della robotica
- **Nathaniel Rochester**: progettò il primo computer a transistor (IBM 704)
- **Claude Shannon**: padre della teoria dell'informazione

La loro visione: determinare se le macchine potessero essere istruite a utilizzare il linguaggio, formare astrazioni e risolvere problemi tipicamente umani.

COME SI SVILUPPA L'INTELLIGENZA ARTIFICIALE?

Task e Algoritmo

- **Task**: Problema da risolvere.
- **Algoritmo**: Serie di passi per risolvere un task.

Esempi di Task

- **Object Recognition**: Riconoscere oggetti in immagini.
- **Sentiment Analysis**: Analizzare sentimenti nei testi.

COSA CONTRIBUISCE ALL'AI?

Esistono più discipline in grado di contribuire alla creazione di una AI: informatica, biologia, psicologia, linguistica, matematica e ingegneria.

EVOLUZIONE DELL'INTELLIGENZA ARTIFICIALE

La storia dell'AI non è una crescita lineare, ma un susseguirsi di **entusiasmi e "inverni"**: fasi di grandi promesse seguite da brusche frenate quando la tecnologia non riusciva a mantenerle, fino alla svolta degli ultimi 15 anni.

1948-1958: I PRIMI PASSI

Tutto comincia con il programma di scacchi di Alan Turing (1948, Turochamp), il primo a usare un algoritmo di ricerca per scegliere la mossa migliore. Seguono i programmi di scacchi di Claude Shannon (1950) e John McCarthy (1951).

Nel 1958 Frank Rosenblatt presenta il **Perceptron**, il primo modello matematico ispirato al neurone biologico: le premesse per le reti neurali sono gettate.

1969-1980: IL PRIMO INVERNO DELL'AI

Nel 1969 Minsky e Papert dimostrano matematicamente i limiti del Perceptron (non riesce a risolvere problemi semplici come lo XOR). L'entusiasmo crolla, i finanziamenti si prosciugano: è il **primo AI Winter**.

La ricerca non si ferma, ma rallenta e si sposta su approcci simbolici e sistemi a regole.

1980-1990: SISTEMI ESPERTI E SECONDO INVERNO

Gli anni '80 vedono il boom dei **Sistemi Esperti** (regole scritte a mano da specialisti di dominio), adottati in massa dalle aziende.

Nel 1986 Rumelhart, Hinton e Williams rilanciano le reti neurali con l'algoritmo di **Backpropagation**, che risolve il problema dell'addestramento su più livelli. Ma i Sistemi Esperti si rivelano troppo rigidi e costosi da mantenere: fine anni '80, **secondo AI Winter**.

1990-2012: LA LUNGA PREPARAZIONE

Con più dati e più potenza di calcolo disponibile, l'apprendimento automatico statistico prende piede silenziosamente: riconoscimento vocale, motori di raccomandazione, primi sistemi di visione artificiale.

Sono le fondamenta su cui poggerà l'esplosione successiva.

2012: ALEXNET E LA RIVOLUZIONE DEEP LEARNING

Nel 2012 la rete neurale **AlexNet** vince nettamente la competizione ImageNet di riconoscimento immagini, staccando ogni altro approccio. È il momento in cui il mondo accademico e industriale si convince che le reti neurali profonde (Deep Learning), allenate su GPU, sono la strada giusta.

Da qui parte la corsa che porterà ai modelli di oggi.

2017: IL TRANSFORMER

Nel paper **"Attention Is All You Need"** (Google, 2017) viene introdotta l'architettura **Transformer**, che elabora intere sequenze in parallelo invece che parola per parola come le RNN (ne parleremo più avanti).

È l'architettura su cui si basano tutti i moderni Large Language Models: GPT, Claude, Gemini, Llama.

2022: CHATGPT E L'ADOZIONE DI MASSA

Il 30 novembre 2022 OpenAI rilascia **ChatGPT**: la tecnologia LLM esisteva già da anni, ma un'interfaccia conversazionale semplice la rende accessibile a centinaia di milioni di persone in pochi mesi.

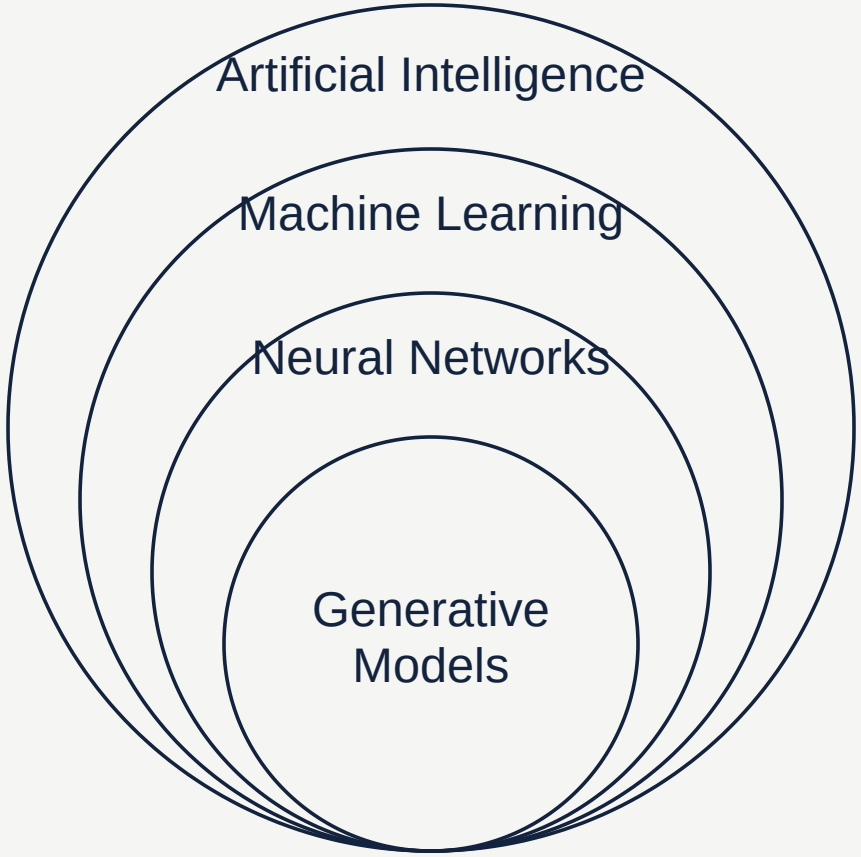
È il momento che ha innescato la rapida diffusione di prodotti AI a cui accenniamo in apertura di questo corso.

CAPACITÀ DI CALCOLO

Con l'aumento delle capacità di calcolo, la ricerca si è mossa verso la robotica, il ragionamento avanzato e l'analisi dei dati su larga scala. Progressi significativi in visione artificiale, produzione automatizzata e guida autonoma.

L'AI è oggi parte della vita quotidiana, con applicazioni che vanno dal riconoscimento vocale alla diagnostica medica, fino agli agenti che eseguono compiti in autonomia.

AI - DA COSA È COMPOSTA



DAI SISTEMI ESPERTI AL MACHINE LEARNING

Nella storia dell'IA abbiamo assistito a un'inversione fondamentale nel modo in cui le macchine acquisiscono competenza.

Sistemi Esperti: un programmatore collaborava con un esperto di dominio per codificare manualmente regole logiche. Se l'obiettivo era identificare lo spam, l'umano scriveva: "Se il messaggio contiene 'offerta' e 'urgente', allora segnala come spam". Metodo fragile, non scalabile, con manutenzione costante.

Machine Learning: l'algoritmo estrae regole autonomamente dai dati. L'esperto non scrive più codice, ma etichetta milioni di esempi. L'algoritmo analizza i dati per individuare pattern statistici e costruire i propri criteri di decisione.

Caratteristica	Sistemi Esperti	Machine Learning
Origine regole	Scritte manualmente	Estratte dai dati
Flessibilità	Molto bassa	Alta
Manutenzione	Elevata	Ridotta (servono dati)
Scalabilità	Limitata	Elevata

COS'È IL MACHINE LEARNING?

In principio esistevano algoritmi hard-coded e regole fisse che si sono dimostrate inadeguate per compiti come riconoscimento di immagini e trattamento del testo

La soluzione è stata quella di simulare il processo di apprendimento umano attraverso il machine learning, in questo modo gli algoritmi imparano da una vasta quantità di dati: vengono addestrati dei modelli in base a dei dati di input.

Questo processo è analogo all'apprendimento umano, dove si inizia con conoscenze basilari e si progredisce verso livelli più avanzati. Inoltre, per affrontare compiti ancora più complessi, è stata introdotta l'idea di imitare il cervello umano attraverso reti neurali.

Queste reti consentono alle macchine di apprendere in modo simile all'elaborazione dei dati da parte dell'essere umano, ampliando così le capacità dell'IA.

LE FASI DEL MACHINE LEARNING

Training: Il sistema apprende regole dai dati.
Utilizzo: Il sistema applica ciò che ha imparato.

ELIZA

ELIZA è un chatbot scritto nel 1966 da Joseph Weizenbaum. Il bot consiste in un analizzatore lessicale e un insieme di regole che permettono di simulare una conversazione in inglese, gallese o tedesco.

Lo script più noto, spesso erroneamente identificato con ELIZA, è DOCTOR, che imita un terapeuta rogersiano.

ELIZA procedeva analizzando e sostituendo semplici parole chiave in frasi preconfezionate. A seconda delle parole che l'utente immetteva nel programma, l'illusione di un interlocutore umano veniva smascherata o poteva continuare per diverse battute. Talvolta risultava talmente convincente che alcune persone erano così convinte di comunicare con un essere umano, da insistere per parecchi minuti.

https://it.wikipedia.org/wiki/ELIZA_(chatterbot)

ELIZA - FOTO

```
  Welcome to
                 EEEEEE  LL      IIII   ZZZZZZ   AAAAA
                 EE      LL       II       ZZ   AA   AA
                 EEEEE   LL       II     ZZZ    AAAAAAA
                 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:
```

TIPOLOGIE DI MACHINE LEARNING

- Unsupervised Learning
  I dati non sono etichettati e il modello deve trovare da solo i pattern e relazioni.
- Supervised Learning
  I dati sono catalogati ed etichettati: ed esempio immagini e loro rappresentazione. Il modello acquisisce competenza in base a quanto ha appreso. Questo tipo di apprendimento è utilizzato per la classificazione e la regressione (predizione dei dati).
- Reinforcement Learning
  Un agente apprende a prende decisioni in un ambiente dinamico interagendo con esso. L'agente riceve feedback sotto forma di ricompense o punizioni in base alle sua azioni. L'obiettivo è massimizzare la ricompensa totale nel lungo termine.

RETI NEURALI

Il machine learning aveva difficoltà con problemi di semplice comprensione, fino a quando si è capito che la limitazione non risiedeva nel concetto di machine learning o nell'imitare il cervello umano.

Il problema era che le reti neurali semplici, con un numero limitato di neuroni collegati in modo basilare, non potevano replicare le capacità del cervello umano.

Questo non dovrebbe sorprendere, dato che il cervello umano contiene circa 86 miliardi di neuroni e una rete di connessioni estremamente complessa.

COME SI MISURANO LE RETI NEURALI?

Per capire la grandezza di una rete neurale possiamo valutare il numero di parametri, ovvero i pesi che vengono adattati durante il processo di apprendimento.

Esempio

- GPT-3: 175 miliardi di parametri (dato ufficiale).
- GPT-4: stimato attorno a 1.76 trilioni di parametri (cifra trapelata via leak, mai confermata ufficialmente da OpenAI).

Dai modelli GPT-5 in poi, così come per la maggior parte dei laboratori (Anthropic, Google, xAI), il numero di parametri non viene più dichiarato ufficialmente. Questo dato sta diventando sempre meno rilevante rispetto all'architettura, alla qualità dei dati di addestramento e alle tecniche di post-training (reasoning, RLHF, agentic tuning).

LE SCALING LAWS

Perché le aziende hanno continuato ad aumentare parametri e dati? Le **scaling laws** (leggi di scala) osservate empiricamente mostrano che le prestazioni di un modello migliorano in modo prevedibile aumentando insieme tre fattori:

1. **Numero di parametri** del modello
2. **Quantità di dati** di addestramento
3. **Potenza di calcolo** (compute) impiegata nel training

Per anni la ricetta è stata "più grande è, meglio è". Dal 2024 in poi si è aggiunto un quarto asse, indipendente dal training: il **calcolo impiegato durante l'inferenza** (vedi slide sui Reasoning Model).

IL PROBLEMA DELL'OPACITÀ

A differenza dei Sistemi Esperti, dove ogni regola è leggibile e spiegabile, una rete neurale è una **"matassa di numeri"**.

I parametri sono come i **"pirulini" (manopole) di un enorme mixer audio**: ruotandoli, si cambia il modo in cui il segnale fluisce, e quindi il risultato finale. Ma con miliardi di manopole, è impossibile per un umano tracciare perché una specifica combinazione produca un certo output.

Conseguenze:

- L'**Inferenza** (l'applicazione di quanto appreso) è matematicamente certa, ma logicamente oscura
- Difficile diagnosticare errori o bias
- Nasce il campo dell'**XAI** (Explainable AI) per rendere le decisioni più trasparenti

COS'È IL DEEP LEARNING?

Il deep learning consiste nell'utilizzo di reti neurali con una maggiore complessità, comprensive di più neuroni, livelli e interconnessioni.

Sebbene non siamo ancora in grado di replicare completamente la complessità del cervello umano, stiamo facendo progressi in questa direzione.

Il deep learning è fondamentale per numerosi sviluppi nell'informatica, come le auto a guida autonoma e il riconoscimento vocale, che sfruttano forme di intelligenza artificiale basate su questa tecnologia.

COSA SONO GLI ALGORITMI GENERATIVI?

Gli algoritmi generativi sono una classe di algoritmi di apprendimento automatico che generano dati sintetici, come immagini, suoni o testo, che sono simili a quelli reali.

Questi algoritmi utilizzano una rete neurale artificiale per apprendere i modelli di dati reali e quindi generare nuovi dati sintetici.

Riceve dei contenuti non strutturati e il modello cerca dei pattern per organizzare questi contenuti e per produrre dei nuovi dati e contenuti.

Gli algoritmi generativi sono utilizzati in diverse applicazioni, come la generazione di immagini per i videogiochi, la sintesi di voci per gli assistenti vocali e la generazione di testo per la scrittura assistita.

COME FUNZIONANO GLI ALGORITMI GENERATIVI?

- Generano contenuti in maniera probabilistica
- Non sono intelligenti e non comprendono: sono ben addestrati su come ci esprimiamo e su come scriviamo
- Non sono dei motori di ricerca: possono produrre delle allucinazioni

DATI DI ADDESTRAMENTO

I dati di addestramento sono un insieme di dati utilizzati per addestrare un modello di machine learning. Questi dati sono utilizzati per insegnare al modello come eseguire un compito specifico, come la classificazione delle immagini o la traduzione automatica.

In rete esistono vari dataset di dati di addestramento, sia a pagamento che gratuiti.

DA DOVE PROVENGONO I DATI DI ADDESTRAMENTO

Alcuni dei più famosi sono rappresentati da

- Common Crawl https://commoncrawl.org/ : più di 250 miliardi di pagine internet nell'arco di 16 anni
- Laion https://laion.ai/projects/ : contiene vari dataset, come LAION5B un dataset di 5.86 miliardi di coppie di immagini/testo
- The Pile https://pile.eleuther.ai/ : 825 GiB di dati di alta qualità, come ad esempio PubMed Central (PMC) un archivio di articoli accademici pubblicati su riviste biomediche

TIPI DI INTELLIGENZA ARTIFICIALE

Esistono varie tipologie di intelligenza artificiale, a seconda del loro scopo e delle loro funzionalità. Alcune delle tipologie più comuni includono:

- ANI (Artificial Narrow Intelligence)
- AGI (Artificial General Intelligence)
- ASI (Artificial Super Intelligence)

ANI (ARTIFICIAL NARROW INTELLIGENCE)

L'intelligenza artificiale stretta è un tipo di IA in cui una tecnologia ha la capacità di superare gli esseri umani in un compito ben definito. Si concentra su un singolo sottoinsieme di abilità cognitive, come la guida autonoma o il riconoscimento facciale.

AGI (ARTIFICIAL GENERAL INTELLIGENCE)

L'intelligenza artificiale forte o intelligenza artificiale generale è la capacità di un agente intelligente di apprendere e capire un qualsiasi compito intellettuale che può imparare un essere umano.

È l'obiettivo principale di alcune delle ricerche nell'intelligenza artificiale e un argomento comune nella fantascienza e nella futurologia.

Alcune fonti accademiche riservano il termine "IA forte" (strong AI) a quei programmi informatici in grado di essere senzienti e di avere una coscienza.

https://it.wikipedia.org/wiki/Intelligenza_artificiale_forte

ASI (ARTIFICIAL SUPER INTELLIGENCE)

AI che è più intelligente dei migliori esseri umani in termini di velocità di elaborazione, abilità di apprendimento e capacità di risolvere problemi.

AI GENERATIVA

L'AI generativa è un tipo di AI che utilizza algoritmi di apprendimento automatico per creare nuovi contenuti, come immagini, video, testi e suoni.

Esempi di architetture alla base dell'AI generativa:

- Reti generative avversariali (GAN)
- Diffusion Model
- Reti neurali ricorrenti (RNN)
- Reti neurali convoluzionali (CNN)
- Transformer (alla base dei moderni LLM)

IL ROVESCIO DELLA MEDAGLIA: AI SLOP

La stessa facilità con cui l'AI genera contenuti ha un lato oscuro: l'**AI Slop** (letteralmente "poltiglia AI").

Il termine indica contenuti (testi, immagini, video) generati in massa, a bassissimo costo e senza revisione umana, con l'unico scopo di riempire il web (per SEO, engagement o pura quantità), non di essere davvero utili o accurati.

Alcuni esempi tipici: articoli-farm che riscrivono notizie senza verifica, immagini "sensazionalistiche" virali sui social, video pubblicitari a basso sforzo, recensioni o commenti generati in blocco.

Il rischio concreto è duplice: **inquinamento dell'informazione** (sempre più difficile distinguere contenuto umano da generato) e **collasso del modello** (model collapse), se i futuri LLM vengono addestrati sempre più su dati generati da altri LLM invece che su contenuti umani originali.

RETI GENERATIVE AVVERSARIALI

Le Reti Generative Avversariali (GAN - Generative Adversarial Networks) sono un'architettura di apprendimento automatico introdotta da Ian Goodfellow nel 2014.

La loro struttura si basa su due reti neurali che competono tra loro in un "gioco" a somma zero:

1. **Il Generatore (G)**: Produce dati sintetici cercando di imitare dei dati reali. Il suo obiettivo è creare esempi così convincenti da "ingannare" il Discriminatore.
2. **Il Discriminatore (D)**: Agisce come un "giudice", cercando di distinguere tra dati reali e generati. Deve classificare correttamente i dati come autentici o falsi.

Le due reti si allenano simultaneamente:

- Il Generatore migliora progressivamente la qualità dei dati sintetici
- Il Discriminatore affina la sua capacità di rilevare le falsificazioni

RETI GENERATIVE AVVERSARIALI - ESEMPI

Alcuni esempi pratici di applicazione delle GAN:

- Generazione di immagini fotorealistiche
- Conversione di schizzi in fotografie
- Invecchiamento/ringiovanimento di volti nelle foto
- Creazione di opere d'arte
- Sintesi di video
- Generazione di dati per aumentare dataset esistenti

Il processo di allenamento è spesso complesso e può presentare instabilità, ma le GAN hanno rivoluzionato il campo della generazione di contenuti artificiali.

Oggi le GAN sono in gran parte storiche: la generazione di immagini e video è dominata dai **Diffusion Model** (vedi slide successiva).

DIFFUSION MODEL

I Diffusion Model sono l'architettura alla base della generazione moderna di immagini e video (Stable Diffusion, Midjourney, DALL-E, Sora).

Il principio è opposto e complementare a quello delle GAN:

1. **Forward process**: si parte da un'immagine reale e le si aggiunge gradualmente rumore, fino a ridurla a puro rumore casuale.
2. **Reverse process**: la rete neurale impara a invertire il processo, "ripulendo" il rumore un passo alla volta fino a ricostruire un'immagine coerente.

Una volta addestrata, la rete genera immagini nuove partendo da rumore casuale, guidata da un prompt testuale.

DIFFUSION MODEL - PERCHÉ HANNO SOPPIANTATO LE GAN

- **Stabilità di training**: niente "gioco a somma zero" tra due reti, l'addestramento è più prevedibile.
- **Qualità e controllo**: risultati più dettagliati e più facili da guidare tramite testo (prompt).
- **Contro**: generazione più lenta (richiede più passaggi di denoising) rispetto a una singola inferenza di una GAN.

RETI NEURALI RICORRENTI

Le Reti Neurali Ricorrenti (RNN - Recurrent Neural Networks) sono un tipo di rete neurale artificiale progettata specificamente per elaborare sequenze di dati, dove l'ordine temporale delle informazioni è importante.

Caratteristiche principali:

1. **Memoria interna**: A differenza delle reti neurali tradizionali, le RNN mantengono uno "stato interno" che funziona come una forma di memoria, permettendo di considerare le informazioni precedenti per elaborare l'input corrente.
2. **Collegamenti ricorrenti**: I neuroni hanno connessioni che formano cicli, permettendo all'informazione di persistere da un passo temporale all'altro.

RETI NEURALI RICORRENTI - VARIANTI

Principali varianti:

1. **LSTM** (Long Short-Term Memory):
   Una LSTM (Long Short-Term Memory) è una variante avanzata delle reti neurali ricorrenti che utilizza un sistema di "gate" (cancelli) per controllare il flusso delle informazioni, permettendo alla rete di memorizzare selettivamente informazioni importanti per lunghi periodi e risolvere il problema del vanishing gradient.
2. **GRU** (Gated Recurrent Unit):
   La GRU (Gated Recurrent Unit) è una versione semplificata della LSTM che combina i gate di dimenticanza e di input in un unico "gate di aggiornamento", mantenendo prestazioni simili, ma con minor complessità computazionale.

RETI NEURALI RICORRENTI - APPLICAZIONI

Applicazioni comuni:

- Elaborazione del linguaggio naturale
- Traduzione automatica
- Riconoscimento vocale
- Previsione di serie temporali
- Generazione di testo
- Analisi di sequenze biologiche (DNA, proteine)

RETI NEURALI RICORRENTI - PRO E CONTRO

Vantaggi:

- Capacità di gestire input di lunghezza variabile
- Mantenimento del contesto temporale
- Flessibilità nell'elaborazione sequenziale

Limitazioni:

- Difficoltà nell'apprendimento di dipendenze a lungo termine (parzialmente risolto con LSTM)
- Costo computazionale elevato per sequenze lunghe
- Possibili problemi di instabilità durante l'addestramento

RETI NEURALI CONVOLUZIONALI

Le Reti Neurali Convoluzionali (CNN o ConvNet) sono un tipo di rete neurale artificiale specificamente progettata per processare dati strutturati a griglia, come le immagini, che utilizza operazioni di convoluzione per estrarre automaticamente caratteristiche rilevanti attraverso filtri che analizzano porzioni locali dell'input.

La convoluzione è un'operazione matematica che applica un filtro (o kernel) a una porzione di dati di input, facendolo scorrere sistematicamente su di essa per produrre una nuova rappresentazione dove ogni valore di output è una somma pesata dei valori di input vicini. Questa operazione permette di rilevare pattern locali come bordi, texture o forme nelle immagini.

RETI NEURALI CONVOLUZIONALI - COMPONENTI

1. **Strati di convoluzione**:
   - Applicano filtri (kernel) all'input
   - Estraggono caratteristiche come bordi, texture, forme
   - Condivisione dei parametri per ridurre la complessità
2. **Strati di pooling**:
   - Riducono le dimensioni (downsampling)
   - Mantengono le caratteristiche più importanti
   - Rendono la rete più robusta a piccole variazioni
3. **Strati completamente connessi**:
   - Combinano le caratteristiche estratte
   - Effettuano la classificazione finale

RETI NEURALI CONVOLUZIONALI - APPLICAZIONI

Applicazioni principali:

- Riconoscimento di immagini
- Visione artificiale
- Elaborazione video
- Analisi medica (raggi X, risonanze)
- Sistemi di guida autonoma

IL TRANSFORMER: SUPERARE IL COLLO DI BOTTIGLIA DELLE RNN

Le RNN, appena viste, elaborano il testo **parola per parola in sequenza**: lento e con difficoltà a mantenere il contesto su testi lunghi.

Nel 2017 il paper **"Attention Is All You Need"** introduce il **Transformer**: un'architettura che elabora **l'intera sequenza in parallelo**, usando il meccanismo di **self-attention** per capire quali parole del testo sono rilevanti tra loro, indipendentemente dalla distanza.

È l'architettura su cui si basano tutti i moderni Large Language Models (GPT, Claude, Gemini, Llama).

LLM: ARCHITETTI DELLA PROBABILITÀ

I Large Language Models non sono oracoli che attingono a una conoscenza cosciente, ma **sofisticati predittori statistici**. Il loro unico compito è la **predizione del token successivo**: data una sequenza di parole, calcolano qual è la parola più verosimile.

Caso studio "Bennato": Alla domanda *"Chi canta La terra dei cachi?"*, un modello potrebbe generare candidati ("Elio", "Francesco", "Edoardo"). Se sceglie inizialmente "Edoardo" (per una fluttuazione statistica), si auto-intrappola: la parola successiva più probabile dopo "Edoardo" non sarà "e le Storie Tese", ma **"Bennato"**. Il risultato "Edoardo Bennato" è linguisticamente perfetto, ma fattualmente errato.

Le Allucinazioni non sono un bug: il modello non "mente", segue semplicemente la distribuzione di probabilità più alta, indipendentemente dalla verità.

LARGE LANGUAGE MODELS (LLM)

I Large Language Models (LLM) sono un tipo di AI generativa che utilizza algoritmi di apprendimento automatico per creare nuovi contenuti, come testi, immagini e suoni.

Si tratta di

- Modelli di AI basati su reti neurali transformer addestrati sul testo.
- Task principale: **Next Token Prediction** (Predizione della prossima parola).
- Capacità emergenti: Generazione di codice, risposte conversazionali, riscrittura di testi.

IN-CONTEXT LEARNING: IMPARARE AL VOLO

Una delle capacità più sorprendenti degli LLM: possono imparare a svolgere un nuovo compito **solo leggendo alcuni esempi nel prompt**, senza alcun riaddestramento (i pesi restano congelati).

- **Zero-shot**: nessun esempio, solo l'istruzione ("Traduci questa frase in francese").
- **Few-shot**: alcuni esempi nel prompt mostrano il pattern desiderato prima di chiedere il compito vero.

Questa capacità emerge dalla scala del modello e dei dati di addestramento, ed è il motivo per cui il "prompt engineering" è diventato un'abilità pratica rilevante.

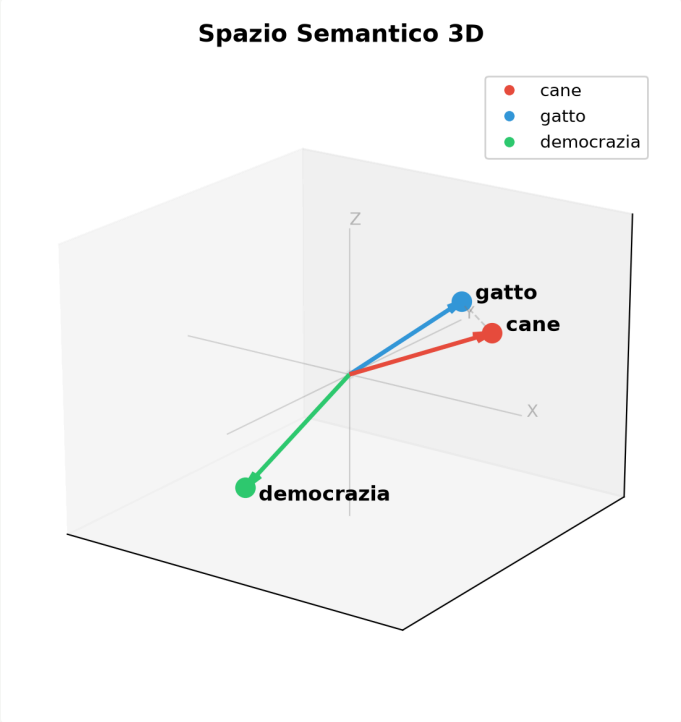
MECCANISMO DELL'ATTENZIONE

- Assegna pesi differenziati alle parole in input per migliorare la comprensione del contesto.
- **Esempio**: "The animal didn't cross the street because it..." → "it" è legato a "the animal".

LA GEOMETRIA DEL LINGUAGGIO: TOKEN E EMBEDDING

Per elaborare i concetti, l'IA trasforma il testo in numeri:

1. **Frammentazione**: Il testo è diviso in **token** (parole o sotto-parole)
2. **Embedding**: Ogni token viene convertito in un **vettore** (sequenza di numeri) e posizionato in uno spazio semantico multidimensionale
3. Concetti simili (es. "cane" e "gatto") finiscono vicini nello spazio; "democrazia" finisce lontano



LA GEOMETRIA DEL LINGUAGGIO: IPOSTESI DISTRIBUZIONALE

"Dimmi con chi vai e ti dirò chi sei"

Il modello definisce il significato di una parola in base a quelle che la circondano abitualmente (Distributional Hypothesis).

Parole che appaiono in contesti simili hanno vettori simili.

LA GEOMETRIA DEL LINGUAGGIO: POLISEMIA DINAMICA

La rete risolve la **polisemia** (parole con più significati) durante il calcolo.

Esempio: la parola "pesca"

- In un contesto di **mare** → i calcoli interni "trascinano" il vettore verso la zona dei pesci
- In un contesto di **orto/frutta** → verso la zona dei frutti

Lo stesso token cambia significato in base al contesto circostante.

IL CICLO DI VITA DI UN MODELLO: 4 FASI

1. Pre-training
Il modello ingesta una quantità colossale di dati (Internet, libri, codice) per imparare sintassi e semantica. I parametri vengono stabiliti per la prima volta e poi **congelati**: il modello non impara più dalle tue correzioni quotidiane.

2. Supervised Fine-tuning
Il modello viene addestrato al dialogo tramite esempi di conversazioni "Utente/Assistente".

3. Allineamento (RLHF)
Si ottimizza il modello secondo le preferenze umane. Previene risposte pericolose, ma introduce la **Sicofania** (tendenza dell'IA a dare ragione all'utente per compiacerlo) e potenziali bias della casa produttrice.

4. Verifica Automatica
Nel coding si usano dati sintetici e problemi verificabili matematicamente: il codice funziona o no (feedback binario). Ecco perché l'IA è progredita molto più velocemente nel coding che in ambiti soggettivi.

OLTRE IL TRAINING: RAG

Il pre-training "congela" la conoscenza del modello a una data precisa. Come si fa a rispondere su eventi recenti o su documenti aziendali privati, senza riaddestrare il modello?

La tecnica standard è la **RAG (Retrieval-Augmented Generation)**:

1. La domanda dell'utente viene usata per **cercare** i documenti più pertinenti in una base di conoscenza esterna (spesso un database vettoriale, basato sugli embedding già visti).
2. I documenti trovati vengono **inseriti nel prompt** insieme alla domanda.
3. Il modello genera la risposta basandosi sul contesto fornito, riducendo (ma non azzerando) il rischio di allucinazioni.

REASONING MODELS: PENSARE PRIMA DI RISPONDERE

Dal 2024 i principali laboratori (OpenAI, Anthropic, Google, DeepSeek) hanno introdotto modelli che, prima di rispondere, generano una **catena di ragionamento interna** (chain-of-thought), esplorando più strade e correggendosi.

- Il modello "pensa di più" su problemi difficili (matematica, codice, logica), impiegando più tempo/calcolo a runtime.
- Questo calcolo aggiuntivo speso **a inferenza** (test-time compute) è oggi un asse di miglioramento importante quanto la scala del training.
- Contropartita: risposte più lente e più costose rispetto a un modello "diretto", da riservare ai compiti che ne beneficiano davvero.

COMPRESSIONE E EFFICIENZA

- **Distillation**: modello più piccolo (student) impara da un modello più grande (teacher).
- **Quantization**: riduzione della precisione dei numeri per rendere i modelli più leggeri.

MIXTURE OF EXPERTS (MOE)

Una tecnica per migliorare l'efficienza e le capacità dei modelli di machine learning, specialmente i Large Language Models (LLM).

L'idea centrale è di dividere il problema di apprendimento tra diversi "esperti" (sotto-modelli), ognuno specializzato in un diverso aspetto dei dati.

MIXTURE OF EXPERTS - COME FUNZIONA

1. **Esperti (Experts)**:
    - Sono tipicamente reti neurali più piccole (ad esempio, feed-forward networks).
    - Ogni esperto impara a gestire specifici tipi di input o task.
2. **Rete Gating (Gating Network)**:
    - È un componente cruciale che determina quale esperto (o combinazione di esperti) attivare per un dato input.
    - Impara a instradare dinamicamente l'input all'esperto più appropriato.
    - Solitamente, per ogni input vengono selezionati solo pochi esperti (es. i top-2), mantenendo basso il costo computazionale.

MIXTURE OF EXPERTS - VANTAGGI

Efficienza Computazionale:

- Attivando solo una piccola frazione degli esperti per ogni input, i modelli MoE possono avere un numero enorme di parametri totali pur mantenendo bassi i costi di inferenza e training per singolo token.
- Questo permette di scalare i modelli a dimensioni molto più grandi rispetto ai modelli densi tradizionali con lo stesso budget computazionale.

MIXTURE OF EXPERTS - VANTAGGI PERFORMANCE

Migliori Performance:

- La specializzazione degli esperti permette al modello di apprendere rappresentazioni più ricche e complesse.
- Può portare a una maggiore qualità e accuratezza, specialmente su dataset vasti e diversificati.

Scalabilità:

- È più facile aumentare la capacità del modello aggiungendo più esperti.

MIXTURE OF EXPERTS - SFIDE

Complessità di Training:

- L'allenamento può essere più complesso e richiedere un tuning attento degli iperparametri.
- Bilanciare il carico tra gli esperti (load balancing) è cruciale per evitare che alcuni esperti vengano sovrautilizzati e altri sottoutilizzati.

Overhead di Comunicazione:

- La necessità di instradare gli input e aggregare gli output degli esperti può introdurre latenza, specialmente in sistemi distribuiti.

Requisiti di Memoria:

- Anche se solo pochi esperti sono attivi per ogni token, tutti i parametri del modello devono essere caricati in memoria durante l'inferenza.

AI CHE UTILIZZANO MIXTURE OF EXPERTS

Molti dei più recenti e performanti Large Language Models utilizzano architetture MoE:

- **Mixtral 8x7B (Mistral AI)**: Il modello open-source che ha reso popolare l'approccio, 8 esperti con 2 selezionati per token. È rimasto un riferimento storico per capire il concetto.
- **Llama 4** (Meta): Scout e Maverick (MoE multimodale) sono stati effettivamente rilasciati; la variante più grande annunciata, Behemoth, non è mai stata pubblicata (progetto accantonato dopo ritardi ripetuti). Da qui Meta ha spostato parte della ricerca frontier su modelli proprietari a pesi chiusi.
- **DeepSeek**: la serie **R1** (671 miliardi di parametri totali, 37 miliardi attivi) ha rappresentato una svolta nel 2025 per rapporto costo/prestazioni nel coding; l'azienda ha poi continuato a evolvere la propria linea MoE con versioni successive più recenti.
- **Grok** di xAI: le generazioni più recenti restano MoE su scala di trilioni di parametri totali.
- **Qwen** di Alibaba: la famiglia open-weight più recente offre sia varianti MoE (con notazione "A" per i parametri attivi, es. 30B-A3B) sia varianti dense.
- **Gemini** di Google: MoE fin dalla serie 1.5, confermato anche nelle generazioni successive.

MOE NEI MODELLI PROPRIETARI

Nel 2026 l'uso di MoE è ormai lo standard de facto per (quasi) tutti i frontier model, sia open weight che proprietari:

- **GPT (OpenAI)**: i dettagli architetturali non sono pubblici, ma è opinione diffusa nella comunità tecnica che le versioni più recenti (dalla famiglia GPT-4 in poi) utilizzino MoE per bilanciare scala e costi di inferenza.
- **Claude (Anthropic)**: l'architettura esatta non è dichiarata pubblicamente; come per gli altri laboratori proprietari, i dettagli su parametri e architettura restano riservati.

La tendenza generale è chiara: più un modello cresce, più conviene attivare solo una parte della rete per ogni richiesta, risparmiando calcolo ed energia.

PROMPTING E SISTEMI

- **Modello**: Architettura, dati e parametri.
- **Interfaccia**: Wrapper tra l'utente e il modello.
- **Server**: Dove il modello è ospitato (privacy e localizzazione).

COSA SONO GLI AGENTI AI?

Un chatbot risponde a una domanda con del testo. Un **agente AI** va oltre: riceve un obiettivo, pianifica dei passi ed **esegue azioni concrete** nel mondo digitale, valutando i risultati e correggendo la rotta.

Il ciclo tipico di un agente:

1. **Ragiona** su come raggiungere l'obiettivo
2. **Agisce** usando uno strumento (Tool Use)
3. **Osserva** il risultato
4. Ripete il ciclo finché l'obiettivo non è raggiunto

TOOL USE E FUNCTION CALLING

Gli LLM di per sé non possono navigare sul web, eseguire codice o leggere un database: sono predittori di testo. Il **Tool Use** (o Function Calling) permette al modello di:

- Richiedere l'esecuzione di una funzione esterna (es. "cerca sul web", "esegui questo codice Python", "leggi questo file")
- Ricevere il risultato e usarlo per proseguire il ragionamento o formulare la risposta finale

È il meccanismo alla base degli agenti AI e di standard recenti come il **MCP (Model Context Protocol)**, che standardizza il modo in cui i modelli si collegano a strumenti e fonti dati esterne.

CLASSIFICAZIONE DEI LLM

Gli LLM (Large Language Models) possono essere classificati in diversi modi, in base a vari criteri.

Ecco alcune delle principali categorie e criteri di classificazione:

CLASSIFICAZIONE DEI LLM - ARCHITETTURA

Architettura del modello:

- Transformer-based (GPT (OpenAI), BERT (Google), Claude (Anthropic), LLaMA (Meta), PaLM (Google))
  Usa self-attention per processare input in parallelo ed è efficace su sequenze lunghe
- LSTM-based (Long Short-Term Memory)
  Usa reti neurali ricorrenti, processa input sequenzialmente, buono per sequenze moderate
- Hybrid architectures
  Combina elementi di diverse architetture (Transformer-XL)

CLASSIFICAZIONE DEI LLM - PARADIGMA

Paradigma di training:

- Autoregressive (es. GPT)
  Predice token successivo basandosi sui precedenti, unidirezionale, adatto per generazione
- Masked Language Models (es. BERT)
  Predice token mascherati in una sequenza, bidirezionale, per comprensione del linguaggio
- Encoder-Decoder (es. T5)
  Combina encoding dell'input e decoding dell'output, per trasformazioni di testo (es. traduzione)

CLASSIFICAZIONE DEI LLM - DIMENSIONI

Gli LLM possono essere catalogati in base al numero di parametri (che riflette la loro dimensione e capacità). Ecco una classificazione generale:

Small LLM (< 1B parametri):

- BERT base (110M)
- DistilBERT (66M)
- GPT-2 small (117M)

Medium LLM (1B - 10B parametri):

- T5 (3B)
- GPT-J (6B)
- BLOOM-7B (7B)

CLASSIFICAZIONE DEI LLM - DIMENSIONI (CONT.)

Large LLM (10B - 100B parametri):

- LLaMA-13B
- Mixtral 8x7B (Mistral AI, 46.7B totali, architettura MoE)
- LongWriter-Zero-32B (THU-KEG, 33B, specializzato nella generazione di testo lungo e coerente)

Extra Large LLM (> 100B parametri):

- GPT-3 (175B) e BLOOM (176B): storicamente i primi a raggiungere questa scala
- PaLM (540B) / PaLM-2 (340B)
- Modelli frontier attuali (2026): i produttori proprietari (GPT, Claude, Gemini, Grok) non dichiarano più la dimensione esatta; tra i modelli open weight più recenti si superano ormai i **trilione di parametri totali** grazie ad architetture MoE (es. famiglie DeepSeek, Qwen, Llama), pur attivandone solo una piccola frazione per ogni richiesta

CLASSIFICAZIONE DEI LLM - DIMENSIONI - NOTA

Note importanti:

- La dimensione non garantisce sempre prestazioni migliori
- L'efficienza dell'architettura e la qualità dei dati di training sono cruciali
- Alcuni produttori non rivelano le dimensioni esatte dei loro modelli

CLASSIFICAZIONE DEI LLM - DOMINIO E LINGUA

Dominio di specializzazione:

- General-purpose
- Domain-specific (es. modelli per il settore medico o legale)

Lingue supportate:

- Monolingue
- Multilingue

CLASSIFICAZIONE DEI LLM - CAPACITÀ

Capacità di task:

- Single-task
  Modelli addestrati per svolgere un singolo compito specifico
- Multi-task
  Modelli capaci di eseguire molteplici compiti linguistici diversi

CLASSIFICAZIONE DEI LLM - TRAINING

Approccio di training:

- Supervised
- Unsupervised
- Semi-supervised
- Self-supervised

Tipo di dati di training:

- Testuali
- Multimodali (testo + immagini, audio, video, ecc.)

ECOSISTEMA LLM: MODELLI, ACCESSIBILITÀ ED EFFICIENZA

Proprietary (Closed Source):

- Definizione: "Black box". Accessibile solo via API o interfaccia web. Codice e pesi non accessibili.
- Vantaggi: Massime prestazioni (Frontier Models), facilità d'uso, manutenzione gestita.
- Esempi: GPT-5 (OpenAI), Gemini 3 (Google), Claude Opus / Sonnet (Anthropic), Grok (xAI).

Open Weights / Open Models:

- Definizione: I "pesi" del modello sono pubblici e scaricabili. Possono essere eseguiti in locale (on-premise) o su cloud privato.
- Vantaggi: Privacy dei dati totale, nessuna dipendenza dal vendor, personalizzazione (Fine-tuning).
- Esempi: Llama 4 (Meta), Mistral (Mistral AI), Qwen (Alibaba), DeepSeek.

AI OPENSOURCE - DEFINIZIONE

La **Open Source AI Definition** stabilisce i criteri per considerare un sistema di intelligenza artificiale (IA) come open source, garantendo le seguenti libertà:

1. **Utilizzo**: Libertà di usare il sistema per qualsiasi scopo senza necessità di permessi.
2. **Studio**: Possibilità di analizzare il funzionamento del sistema e ispezionarne i componenti.
3. **Modifica**: Facoltà di modificare il sistema per qualsiasi scopo, inclusa la modifica dei suoi output.
4. **Condivisione**: Capacità di distribuire il sistema, con o senza modifiche, per qualsiasi scopo.

AI OPENSOURCE - APPLICAZIONE

Queste libertà si applicano sia al sistema completo che ai suoi singoli componenti.
Per esercitarle, è necessario avere accesso alla forma preferita per apportare modifiche, che include:

- **Informazioni sui Dati**: Descrizione dettagliata dei dati utilizzati per l'addestramento, comprendente provenienza, caratteristiche, metodi di ottenimento e selezione, procedure di etichettatura e metodologie di elaborazione.
- **Codice**: Codice sorgente completo utilizzato per addestrare ed eseguire il sistema, inclusi processi di elaborazione dei dati, configurazioni di addestramento e architettura del modello.
- **Parametri**: Parametri del modello, come pesi o altre impostazioni, necessari per replicare o modificare il sistema.

AI OPENSOURCE - ELEMENTI

Questi elementi devono essere resi disponibili sotto termini approvati dall'Open Source Initiative (OSI), assicurando trasparenza e possibilità di modifica.

La definizione mira a promuovere l'autonomia, la trasparenza e la collaborazione nel campo dell'IA, allineandosi ai principi del software open source.

https://opensource.org/ai/open-source-ai-definition

OPEN WEIGHTS VS OPEN MODELS

Vista la libertà totale del modello Open Models, le aziende hanno ben presto mirato verso il modello Open Weights.

La differenza tra Open Weights e Open Models (spesso chiamati "True Open Source") è sottile, ma cruciale: riguarda il livello di trasparenza e libertà che ti viene concesso "sotto il cofano".

Spesso nel marketing (come fa Meta con Llama) si usa il termine "Open Source" in modo improprio. Ecco cosa devi sapere per distinguere le due categorie.

OPEN WEIGHTS (PESI APERTI)

È la categoria più comune oggi. L'azienda ti fornisce il "motore" finito, pronto per essere acceso, ma non ti dà i progetti per costruirlo né ti dice esattamente quali materiali ha usato.

- Cosa ottieni: I parametri del modello (i pesi neurali) scaricabili. Puoi eseguirlo sul tuo PC o server e spesso puoi farne il fine-tuning (addestrarlo ulteriormente sui tuoi dati).
- Cosa manca: Non hai accesso al dataset originale di addestramento (quali libri/siti ha letto?) né al codice completo della pipeline di training.
- Licenze: Spesso hanno restrizioni (es. "non uso militare", "non per aziende con >700M utenti").
- Esempi: Llama 4 (Meta), Mistral (Mistral AI), Gemma (Google), Qwen (Alibaba), DeepSeek.

OPEN MODELS / OPEN SOURCE AI

Questa è la vera filosofia open source applicata all'AI. L'obiettivo è la riproducibilità scientifica totale.

- Cosa ottieni: Tutto ciò che serve per ricostruire il modello da zero: i pesi, il dataset completo (o la lista dettagliata delle fonti), il codice di addestramento, i log di training e le note di sviluppo.
- Vantaggio chiave: Puoi verificare se ci sono bias nei dati alla fonte e hai la certezza matematica di come è stato costruito il "cervello" dell'AI.
- Esempi: OLMo (Allen Institute for AI), Pythia (EleutherAI), Bloom.

TABELLA DI CONFRONTO RAPIDO

Caratteristica	Open Weights	Open Models
Pesi scaricabili	✅ Sì	✅ Sì
Eseguibile in locale	✅ Sì	✅ Sì
Dataset di Training	❌ Segreto / Non divulgato	✅ Pubblico / Documentato
Codice di Training	❌ Spesso assente	✅ Completamente disponibile
Licenza d'uso	Spesso restrittiva (Custom License)	Permissiva (es. Apache 2.0, MIT)
Riproducibilità	Bassa (Black box parziale)	Alta (Trasparenza scientifica)

QUALE SCEGLIERE?

Se sei un'azienda, gli Open Weights vanno benissimo: ottieni un modello potente (come Llama) gratis da usare sui tuoi server.

Se lavori in ambiti regolamentati (sanità, finanza) o accademici, potresti preferire Open Models per avere la garanzia assoluta di sapere su quali dati è stato addestrato il modello (evitando problemi di copyright o bias nascosti).

L'AI ACT EUROPEO

L'**AI Act** (Regolamento UE 2024/1689) è la prima legge organica al mondo sull'intelligenza artificiale: regola lo sviluppo e l'uso dei sistemi AI in base al **rischio** che comportano.

Quattro livelli di rischio:

- **Inaccettabile**: vietato (es. social scoring, manipolazione subliminale)
- **Alto rischio**: obblighi stringenti (es. selezione del personale, credito, sanità, giustizia)
- **Limitato**: obblighi di trasparenza (es. dichiarare che si sta parlando con un chatbot, o che un contenuto è generato da AI)
- **Minimo**: nessun obbligo specifico (es. filtri antispam, videogiochi)

AI ACT - SCADENZE E SANZIONI

Il regolamento si applica in modo progressivo: alcuni obblighi erano già attivi dal 2025, mentre il **2 agosto 2026** segna la piena applicazione, inclusi gli obblighi di trasparenza sui contenuti generati da AI (l'obbligo di watermarking è stato invece rinviato al 2 dicembre 2026 dal successivo "Digital Omnibus" sull'AI).

Le sanzioni per le violazioni più gravi possono arrivare fino a **35 milioni di euro o il 7% del fatturato globale annuo** dell'azienda: un impatto potenzialmente enorme per chi sviluppa o integra sistemi AI ad alto rischio.

EFFICIENZA E ARCHITETTURA

Il concetto di efficienza si è evoluto verso modelli più piccoli ma estremamente capaci (SLM) e architetture intelligenti.

- Frontier Models (Dense & MoE):

  Modelli massivi (>100B parametri) progettati per ragionamento complesso, coding e creatività. Spesso utilizzano architetture Mixture of Experts (MoE) per attivare solo una parte del cervello del modello per ogni richiesta, risparmiando energia.

- Small Language Models (SLM) & Edge AI:

  Modelli compatti (<10B parametri) ottimizzati per girare su laptop o smartphone.

  Non sono solo "modelli grandi tagliati" (pruned), ma addestrati specificamente su dati di altissima qualità per massimizzare l'efficienza.

  Esempi: Microsoft Phi (ultime versioni), Google Gemma 3, Llama (versioni edge più recenti)

ESEGUIRE UN LLM IN LOCALE

Avere a disposizione i pesi di un modello (Open Weights) non basta: serve anche uno strumento che lo faccia girare, e hardware sufficiente.

La tecnica chiave è la **quantizzazione**: riduce la precisione numerica dei pesi (ad esempio da 16 a 4 bit), tagliando drasticamente i requisiti di RAM/VRAM a fronte di una perdita di qualità contenuta. È ciò che rende possibile eseguire un modello da alcuni miliardi di parametri anche su un normale laptop, senza GPU dedicate di fascia alta.

I due strumenti più diffusi per farlo senza scrivere codice:

- **Ollama**: scarica, quantizza e serve i modelli da riga di comando (o tramite una piccola API locale), con un comando semplice come `ollama run llama3`.
- **LM Studio**: stessa filosofia, ma con un'interfaccia grafica pensata per chi vuole scaricare e provare modelli locali senza usare il terminale.

FIN DOVE SI PUÒ SPINGERE IL LOCALE? IL CASO COLIBRÌ

Un esempio estremo di quanto lontano si possa spingere questa logica: **Colibrì**, un motore di inferenza sperimentale scritto da un singolo sviluppatore, riesce a far girare **GLM-5.2** (un modello open weights da **753 miliardi di parametri**) su un normale portatile consumer.

Come? Trattando VRAM, RAM e disco come un'unica gerarchia di memoria fluida: il grosso del modello resta su un SSD NVMe (370 GB in int4), i parametri condivisi in RAM, e gli "esperti" della sua architettura MoE vengono caricati dinamicamente solo quando servono (cache LRU).

Il prezzo da pagare è la velocità: da 0,05-0,1 token/secondo su hardware modesto fino a 1-2 token/secondo su un Mac di fascia alta, utilizzabile ma lontano dai tempi di risposta di un servizio cloud.

https://aitalk.it/it/colibri

DISCLAIMER

Per la realizzazione di questo materiale sono stati utilizzati strumenti di intelligenza artificiale (tra cui **ChatGPT**, **Gemini** e **Claude**) per generare immagini, trovare spunti e idee, correggere le bozze, verificare i link e impaginare graficamente i contenuti.

Ogni slide è stata comunque **rivista integralmente dall'autore**, che si assume la responsabilità di questa pubblicazione.

Nonostante la revisione, in un materiale di queste dimensioni possono sfuggire errori o imprecisioni: se ne trovi, **segnalalo pure**, è un aiuto prezioso per migliorare il corso.

GRAZIE

Domande?

https://github.com/matteobaccan/CorsoAI