>>AI CONF 2026

GenAI fundamentals

Alessandro Vitale

CEO

Conversate

improove

Kudos++

Executive

beSharp. we make IT run.

LEXROOM

REPLY STORM

Gold

seeweb

FASTWEB + vodafone

GenAI dalle fondamenta

Token, training, costi e agenti — l'AI generativa spiegata dalle basi

Alessandro Vitale

Lavoro

- Fondato due startup di AI

- In precedenza Head of Strategic Planning @ Siemens

Impegno Civile

- Task Force per l'AI nella Pubblica Amministrazione

Agenzia per l'Italia Digitale
Presidenza del Consiglio dei Ministri

- G7 Multistakeholder Conference sull'AI

Divulgazione

- Lezioni & Conferenze

- Autore di un libro e di un podcast

conversate

La mappa del talk

1. **Come nasce una risposta**
   il modello che genera
2. **Com'è fatto e come si addestra**
   dal testo grezzo all'assistente
3. **Farlo girare**
   efficienza, hardware e costi
4. **Fargli fare cose utili**
   dal chatbot al sistema che agisce
5. **Quanto è bravo**
   misurare i modelli — e la tua app

conversate

PARTE 1

Come nasce una risposta

il modello che genera

Cos'è un token

- Tutto — prompt e risposta — viene prima spezzato in token.
- Il modello fa una cosa sola: dato l'elenco di token finora, indovina il prossimo.
- ≈ i mattoncini Lego del linguaggio. L'italiano è meno efficiente dell'inglese → più token.

IN UNA FRASE

L'unità minima che il modello legge e scrive. Non parole, non lettere: pezzi di parola.

https://huggingface.co/spaces/Xenova/the-tokenizer-playground

https://openrouter.ai/blog/insights/opus-47-tokenizer-analysis/

Il contesto (context window)

- Esempio: 200k token di finestra.
- Superato il limite, perde l'inizio della conversazione.
- Non è memoria persistente: a ogni richiesta rilegge tutto da capo.

IN UNA FRASE

Quanti token il modello tiene 'in testa' insieme: prompt + conversazione + risposta.

https://chatgpt.com/pricing/

Come sceglie il token: temperature

- Bassa = sempre il più probabile (deterministico, ripetitivo).
- Alta = più varietà, e più rischio di errori.
- Spiega perché la stessa domanda dà risposte diverse.

IN UNA FRASE

Non sceglie IL prossimo token: produce una probabilità su tutti. La temperature decide quanto è 'creativo'.

https://blog.lukesalamone.com/posts/what-is-temperature/

Embedding

- Frasi simili → vettori vicini nello spazio.
- Base della ricerca semantica e del RAG.
- L'esempio classico: 're – uomo + donna ≈ regina'.

IN UNA FRASE

Trasformare testo in vettori di numeri che catturano il significato.

conversate

PARTE 2

Com'è fatto e come si addestra

dal testo grezzo all'assistente

Pretraining

- Nessuna etichetta umana: il testo stesso è la risposta (self-supervised).
- Qui impara lingua, fatti, ragionamento di base.
- Esce un modello grezzo che completa testo, ma non è ancora un assistente.

IN UNA FRASE

La fase enorme e costosa: legge una porzione gigantesca di testo imparando a predire il token successivo.

SFT — Supervised Fine-Tuning

- Da 'completatore di testo' a 'assistente che risponde'.
- Dataset più piccolo ma di alta qualità.

IN UNA FRASE

Gli mostri esempi curati da umani: domanda → risposta ideale.

https://openai.com/index/instruction-following/

RLHF

- Si addestra un modello di preferenza dai confronti a coppie.
- Il modello viene ottimizzato per produrre risposte preferite.
- È ciò che lo rende utile, educato, allineato.

IN UNA FRASE

Reinforcement Learning from Human Feedback: gli umani confrontano risposte e il modello impara cosa è 'preferito'.

https://openai.com/index/instruction-following/

RLVR — la 'RL da codice'

- I test passano? La risposta matematica è giusta?
- Perfetto per codice e matematica.
- È il motore dietro i salti recenti nel reasoning.

IN UNA FRASE

Reinforcement Learning from Verifiable Rewards: il premio non lo dà un umano ma un verificatore automatico.

Reasoning / test-time compute

- Spesso il ragionamento non lo vedi.
- Più calcolo a runtime → risposte migliori sui problemi difficili.
- Un secondo asse di scaling, oltre alla dimensione del modello.

IN UNA FRASE

Lasciare che il modello 'pensi' prima di rispondere: produce ragionamento intermedio (chain of thought).

https://x.com/polynoamial/status/2064210146558136827

Modelli multimodali / omni

- 'Omni' = un solo modello che gestisce più modalità in input e output.
- Non pezzi separati incollati insieme.

IN UNA FRASE

Non solo testo: anche immagini, audio, a volte video — e a volte li producono.

Diffusion models

- Diverso dagli autoregressive (un token alla volta).
- Per le immagini è lo standard.
- Per il testo è frontiera sperimentale.

IN UNA FRASE

Il paradigma dietro la generazione di immagini: si parte dal rumore e lo si 'ripulisce' passo dopo passo.

https://blog.google/innovation-and-ai/technology/developers-tools/diffusion-gemma-faster-text-generation/

conversate

PARTE 3

Farlo girare

efficienza, hardware e costi

KV cache

- È ciò che rende veloce la generazione.
- È il motivo per cui il 'prompt caching' abbatte costi e latenza...
- ...quando riusi lo stesso contesto.

IN UNA FRASE

Generando token per token, il modello rifarebbe ogni volta gli stessi calcoli sui token già visti. La KV cache li salva.

Memory bound vs compute bound

- Generare token è memory bound: il limite è spostare dati, non i calcoli.
- Elaborare il prompt iniziale è compute bound.
- Spiega perché l'output costa più dell'input.

IN UNA FRASE

Due colli di bottiglia diversi nel far girare un modello.

https://x.com/TheAhmadOsman/status/2062312164455862286

MoE — Mixture of Experts

- Modelli enormi come capacità...
- ...ma molto più economici da far girare.
- Architettura ormai comune nei modelli di frontiera.

IN UNA FRASE

Invece di un'unica rete gigante sempre attiva, tante sotto-reti 'esperte': per ogni token se ne accendono solo alcune.

Quantizzazione

- Modello più leggero e veloce.
- Perdita di qualità minima.
- Permette di far girare modelli grossi su hardware modesto.

IN UNA FRASE

Comprimere i pesi del modello usando meno bit per numero (es. da 16 a 4 bit).

Distillazione

- Lo student diventa molto più capace di quanto sarebbe da solo.
- A una frazione del costo.
- Molti modelli 'small' potenti nascono così.

IN UNA FRASE

Un modello grande e bravo ('teacher') insegna a uno piccolo ('student') a imitarlo.

Il costo dei token

- Input: di solito economico.
- Output: più caro (spesso diverse volte l'input).
- Cached input: scontatissimo, riusa la KV cache.

IN UNA FRASE

Si paga a token, con tre tariffe diverse.

https://platform.claude.com/docs/en/about-claude/pricing
https://x.com/SemiAnalysis_/status/2064815044085318040
https://www.microsoft.com/en-us/microsoft-365/blog/2026/06/16/copilot-cowork-is-now-generally-available/

conversate

PARTE 4

Fargli fare cose utili

dal chatbot al sistema che agisce

System prompt

- Chi è, tono, regole, cosa può e non può fare.
- L'utente non lo vede, ma plasma tutto il comportamento.

IN UNA FRASE

Le istruzioni 'di regia' date al modello prima della conversazione.

https://platform.claude.com/docs/en/release-notes/system-prompts

Tool use (function calling)

- Il modello non esegue: decide quando e con quali argomenti chiamare.
- Tu esegui e gli ridai il risultato.
- Il salto da 'chatbot' a 'sistema che fa cose'.

IN UNA FRASE

Dare al modello la capacità di chiamare strumenti esterni: web, codice, query a un DB.

https://developers.openai.com/api/docs/guides/function-calling

MCP — Model Context Protocol

- Invece di integrare ogni tool a mano, parli un protocollo comune.
- ≈ la USB-C degli strumenti AI.
- Un server MCP espone capacità, il modello le usa.

IN UNA FRASE

Uno standard aperto per collegare i modelli a strumenti e dati esterni.

https://x.com/AymericRoucher/status/1908151635802489271

Skill

- Il modello le 'scopre' e le usa al volo.
- Invece di avere tutto sempre nel prompt.

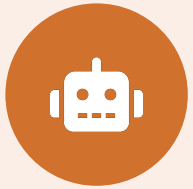
IN UNA FRASE

Capacità modulari e riutilizzabili: istruzioni + risorse impacchettate, che si attivano quando servono.

https://github.com/anthropics/skills/blob/main/skills/pptx/SKILL.md

Agenti, harness, Claude Code, Cowork

- Claude Code: per sviluppatori, dal terminale.
- Cowork: per lavoro non tecnico.
- Il salto da 'ti rispondo' a 'lo faccio io'.



IN UNA FRASE

Agente = modello + tool + un loop ('harness') che pianifica, agisce, osserva il risultato e ripete finché il task è fatto.

https://lilianweng.github.io/posts/2023-06-23-agent/

conversate

PARTE 5

Quanto è bravo

misurare i modelli — e la tua app

Benchmark e LM Arena

- Benchmark = test standard (MMLU, GPQA, SWE-bench…).
- LM Arena = umani votano a coppie in cieco → classifica Elo.
- Attenzione: i benchmark si 'saturano' e a volte finiscono nel training.

IN UNA FRASE

Come si misurano i modelli.

https://www.anthropic.com/news/claude-fable-5-mythos-5
https://arena.ai/leaderboard/text/pareto
https://x.com/ArtificialAnlys/status/2069148772446425563/photo/1

Evals

- Costruisci un set di casi rappresentativi.
- Misuri se una modifica migliora o peggiora.
- Senza eval navighi a vista. È il 'test automatico' dell'era LLM.

IN UNA FRASE

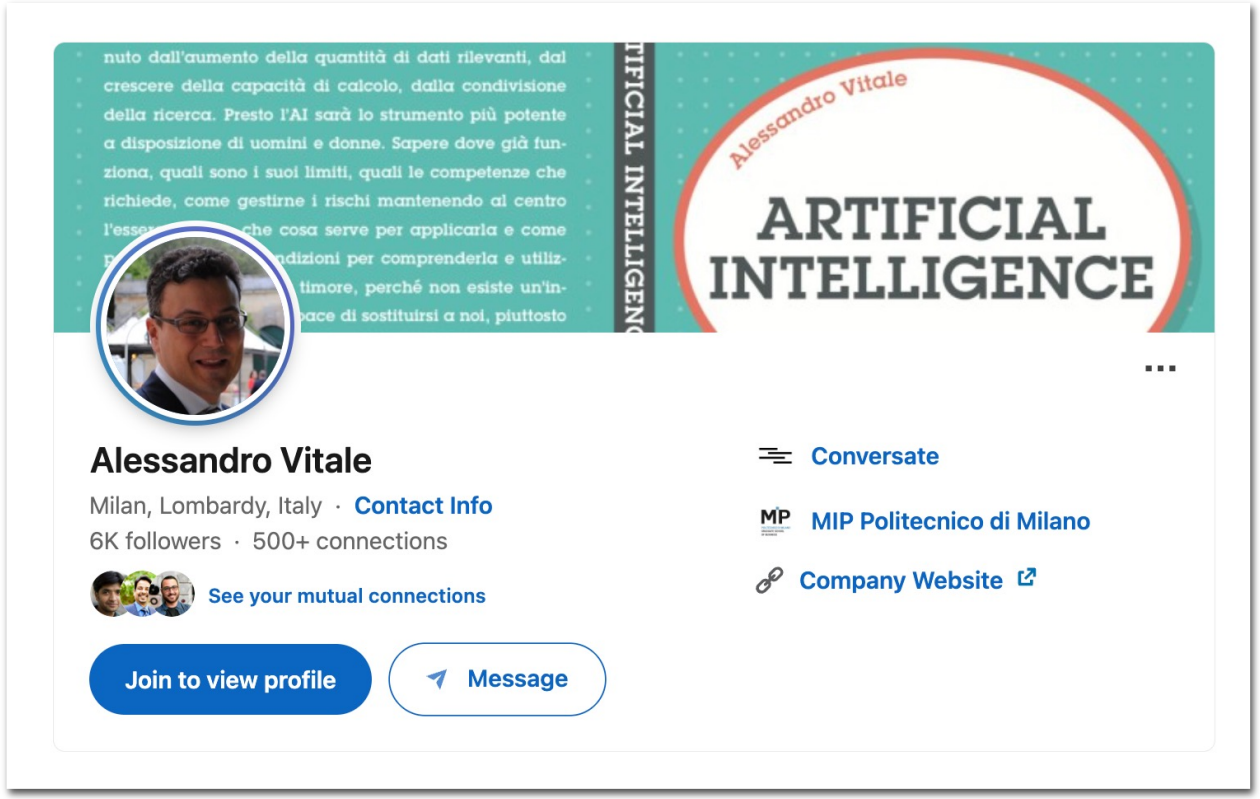
Diverso dai benchmark pubblici: come testi la TUA applicazione.

conversate

Le 5 idee da portare a casa

1. Il modello fa una cosa: indovina il token successivo. Tutto il resto è ingegneria attorno a questo.
2. Si costruisce a strati: pretraining → SFT → RLHF/RLVR. Il 'carattere' arriva dal post-training.
3. Costi e velocità dipendono da token, KV cache e hardware: ottimizzare il contesto fa risparmiare.
4. Tool use + MCP + agenti trasformano il chatbot in un sistema che fa cose.
5. Benchmark per orientarsi, ma sono le tue eval a dirti se la TUA app funziona.

Contatti



https://linkedin.com/in/alessandrovitale

Podcast: Intelligenza Mensile

https://www.youtube.com/@AlessandroVitale-ri2he

https://open.spotify.com/show/1UYwQnoFb2yN2VCBDPhU49

Thank you!

👉 slides & videos: https://www.improove.tech/videos

>>AI CONF 2026

improove