Università degli Studi di Palermo

dipartimento di ingegneria unipa

# Architetture e modelli

Corso di Natural Language Processing (Elaborazione del Linguaggio Naturale)

a.a. 2025/2026

Prof. Roberto Pirrone

CHILAB

# Sommario

- Tipologie di LLM
- Mixture of Experts
- Models
  - GPT
  - Claude
  - Gemini
  - Gemma
  - LLaMA
  - Mistral
  - Phi
  - DeepSeek
  - Qwen
- LLM in italiano
- Interagire con i modelli

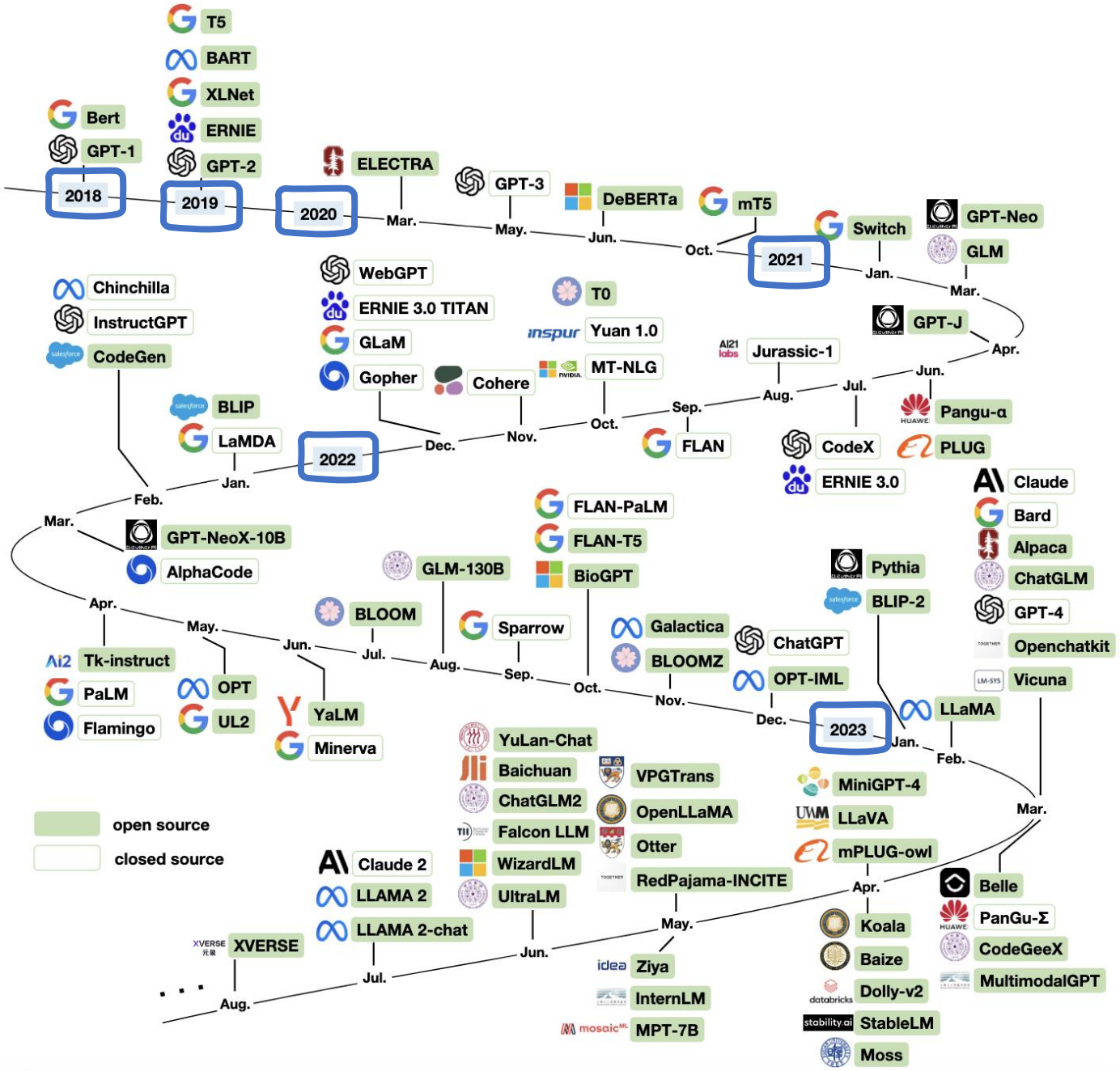
# Tipologie di LLM

- Embedding LLM
  - Modelli di tipo Encoder-only
  - Generano rappresentazioni vettoriali del testo dato in input

- Generator LLM
  - Modelli di tipo Decoder-only
  - Generano testo in linguaggio naturale dato un prompt

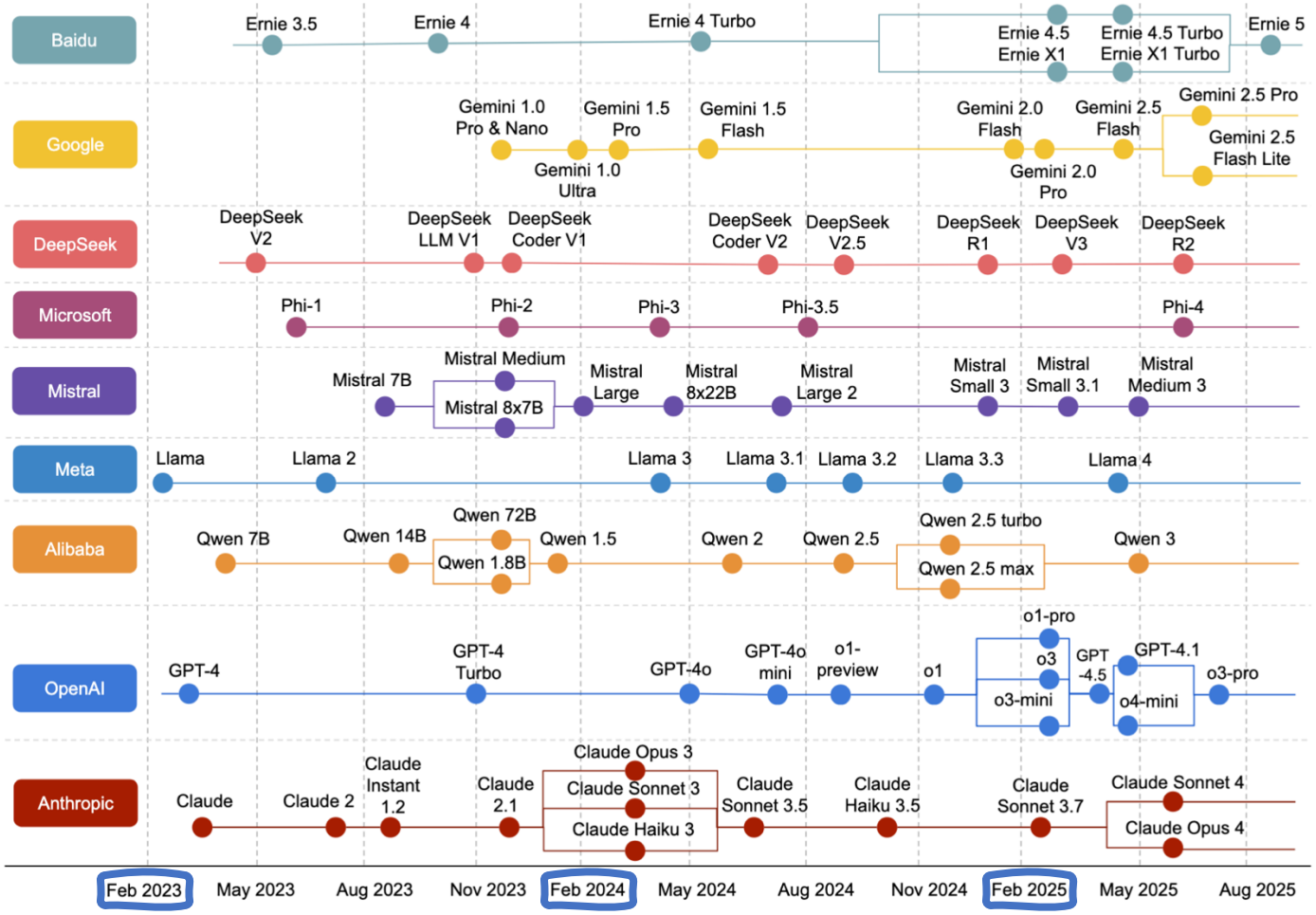
# Tipologie di LLM

- Delvin et al, 2017
  Attention is all you need




https://arxiv.org/abs/2308.14149

# Tipologie di LLM




https://arxiv.org/abs/2507.18479

# Tipologie di LLM

- Foundational models
  - Modelli fondazionali (base) e general-purpose
  - Addestrati su una grande quantità di dati al fine di catturare un ampio spettro di conoscenza linguistica e contestuale
  - Addestramento con tecniche self-supervised

- Instruction-tuned models
  - Modelli addestrati per seguire le istruzione date in input
  - Addestrati su dataset annotati
  - Addestramento con tecniche supervisionate (Supervised Fine Tuning - SFT)

# Tipologie di LLM

- Multilingual models
  - Modelli base o instruct con la *conoscenza* di più lingue
  - Possono essere ottenuti from scratch o in seguito ad una fase di fine-tuning orientato alla language adaptation
- Multimodal models
  - Modelli base o instruct
  - Modelli in grado di ricevere in input anche immagini, video e audio, oltre all'elemento testuale (prompt e/o input)

# Mixture of Experts (MoE)

- La dimensione di un modello è una delle dimensioni su cui è possibile agire per ottenere modelli migliori.
- Fissato il budget di calcolo, è meglio addestrare un modello più grande per un meno passi è meglio che non un modello piccolo per più step.
- Un MoE è una soluzione di tradeoff: è costituito da un insieme di reti FFN (gli «esperti») che si specializzano su alcuni token

https://huggingface.co/blog/moe#what-is-a-mixture-of-experts-moe

# Mixture of Experts (MoE)

- Mixture of Experts (MoE) consente di pre-addestrare i modelli con minor budget di calcolo
  - È possibile aumentare drasticamente le dimensioni del modello o del dataset con lo stesso budget di calcolo di un modello denso.
  - Un modello MoE dovrebbe raggiungere la stessa qualità della sua controparte densa molto più velocemente durante il pre-addestramento.



https://huggingface.co/blog/moe#what-is-a-mixture-of-experts-moe

# Mixture of Experts (MoE)

Un MoE è composto da due elementi principali:

- Layer MoE sparsi che sono utilizzati al posto dei layer densi feed-forward network (FFN).
  - I layer MoE hanno un certo numero di esperti, dove ogni esperto è una rete neurale.
  - Gli esperti sono FFN, ma possono anche essere reti più complesse o anche un MoE stesso, portando a MoE gerarchici



https://huggingface.co/blog/moe#what-is-a-mixture-of-experts-moe

# Mixture of Experts (MoE)

Un MoE è composto da due elementi principali:

- Una gate network o *router* che determina quali token sono inviati a quale esperto.
  - È possibile inviare un token a più di un esperto.
  - Come instradare un token a un esperto è una delle grandi decisioni quando si lavora con MoE
  - Il router è composto da parametri appresi e viene pre-addestrato contemporaneamente al resto della rete.



https://huggingface.co/blog/moe#what-is-a-mixture-of-experts-moe

# Mixture of Experts (MoE)

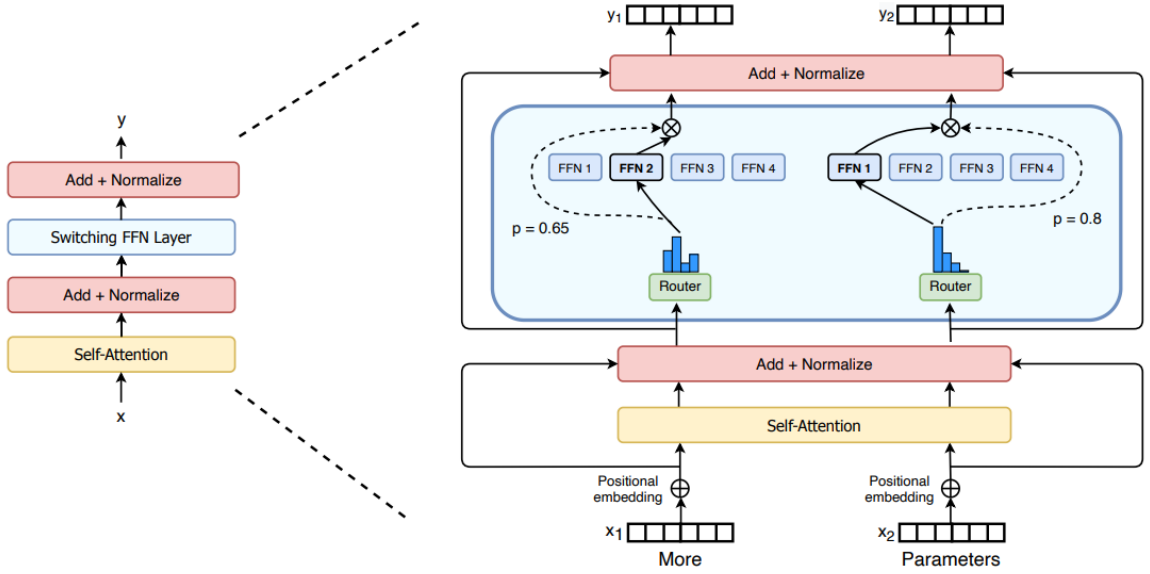



Figure 2: Illustration of a Switch Transformer encoder block. We replace the dense feed forward network (FFN) layer present in the Transformer with a sparse Switch FFN layer (light blue). The layer operates independently on the tokens in the sequence. We diagram two tokens ($x_1$ = "More" and $x_2$ = "Parameters" below) being routed (solid lines) across four FFN experts, where the router independently routes each token. The switch FFN layer returns the output of the selected FFN multiplied by the router gate value (dotted-line).

https://huggingface.co/blog/moe#what-is-a-mixture-of-experts-moe

# Mixture of Experts (MoE)

- Nei MoE i layer FFN dei Transformers sono sostituiti dai layer MoE, composti da una gate network e un certo numero di esperti

- Sebbene i MoE offrano vantaggi come un pre-addestramento più efficiente dal punto di vista del calcolo e un'inferenza più veloce rispetto ai modelli densi, presentano anche delle sfide

https://huggingface.co/blog/moe#what-is-a-mixture-of-experts-moe

# Mixture of Experts (MoE)

- A fronte di un pretraining molto più efficiente, i MoE storicamente hanno faticato a generalizzare durante il fine-tuning, portando all'overfitting.

- Sebbene un MoE possa avere molti parametri, solo alcuni di essi vengono utilizzati durante l'inferenza.
  - Inferenza molto più veloce rispetto a un modello denso con lo stesso numero di parametri.
  - Tutti i parametri devono essere caricati nella RAM, quindi i requisiti di memoria si mantengono elevati.



https://huggingface.co/blog/moe#what-is-a-mixture-of-experts-moe

# Mixture of Experts (MoE)

- Mixtral 8x7B, ha 8 esperti da 7B e per la sua esecuzione richiede una VRAM capace di contenere un modello denso da 47B parametri.

- 47B e non 8×7B=56B perché nei modelli MoE, solo i layer FFN sono trattati come singoli esperti, mentre il resto dei parametri del modello è condiviso.



https://huggingface.co/blog/moe#what-is-a-mixture-of-experts-moe

# Mixture of Experts (MoE)

- Cosa apprende un esperto?
- È stato osservato che gli encoder esperti si specializzano su un gruppo di token o su concetti superficiali.
  - Esperti in punteggiatura, esperto nei nomi
- D'altra parte, i decoder hanno una minor specializzazione.
  - In un addestramento multilingua, sebbene ci si aspetti che ci siano esperti per ciascuna lingua, questo non accade
  - Questo è dovuto al meccanismo di routing e bilanciamento del carico della rete.

https://huggingface.co/blog/moe#what-is-a-mixture-of-experts-moe

# Mixture of Experts (MoE)

| Expert specialization | Expert position | Routed tokens |
|---|---|---|
| **Sentinel tokens** | Layer 1 | been <extra_id_4><extra_id_7>floral to<br><extra_id_10><extra_id_12><extra_id_15><br><extra_id_17><extra_id_18><extra_id_19>... |
| | Layer 4 | <extra_id_0><extra_id_1><extra_id_2><br><extra_id_4><extra_id_6><extra_id_7><br><extra_id_12><extra_id_13><extra_id_14>... |
| | Layer 6 | <extra_id_0><extra_id_4><extra_id_5><br><extra_id_6><extra_id_7><extra_id_14><br><extra_id_16><extra_id_17><extra_id_18>... |
| **Punctuation** | Layer 2 | , , , , , , , , , - , , , , , ). ) |
| | Layer 6 | , , , , , : . : , & , & & ? & - , , ? , , , . <extra_id_27> |
| **Conjunctions and articles** | Layer 3 | The the the the the the the the the The the the<br>the the the The the the the |
| | Layer 6 | a and and and and and and and or and a and .<br>the the if ? a designed does been is not |
| **Verbs** | Layer 1 | died falling identified fell closed left posted lost felt<br>left said read miss place struggling falling signed died<br>falling designed based disagree submitted develop |
| **Visual descriptions**<br>*color, spatial position* | Layer 0 | her over her know dark upper dark outer<br>center upper blue inner yellow raw mama<br>bright bright over open your dark blue |
| **Proper names** | Layer 1 | A Mart Gr Mart Kent Med Cor Tri Ca Mart<br>R Mart Lorraine Colin Ken Sam Ken Gr Angel A<br>Dou Now Ga GT Q Ga C Ko C Ko Ga G |
| **Counting and numbers**<br>*written and numerical forms* | Layer 1 | after 37 19. 6. 27 I I Seven 25 4, 54 I two dead we<br>Some 2012 who we few lower each |

https://huggingface.co/blog/moe#what-is-a-mixture-of-experts-moe

# GPT

- Generative Pre-trained Transformer (GPT)

- Sviluppato da OpenAI (con la partecipazione di Microsoft).
- ChatGPT è l'ultimo membro di questa famiglia.
- Sottoposto a pretraining con Language Modeling e successivamente a fine-tuning con supervisione.

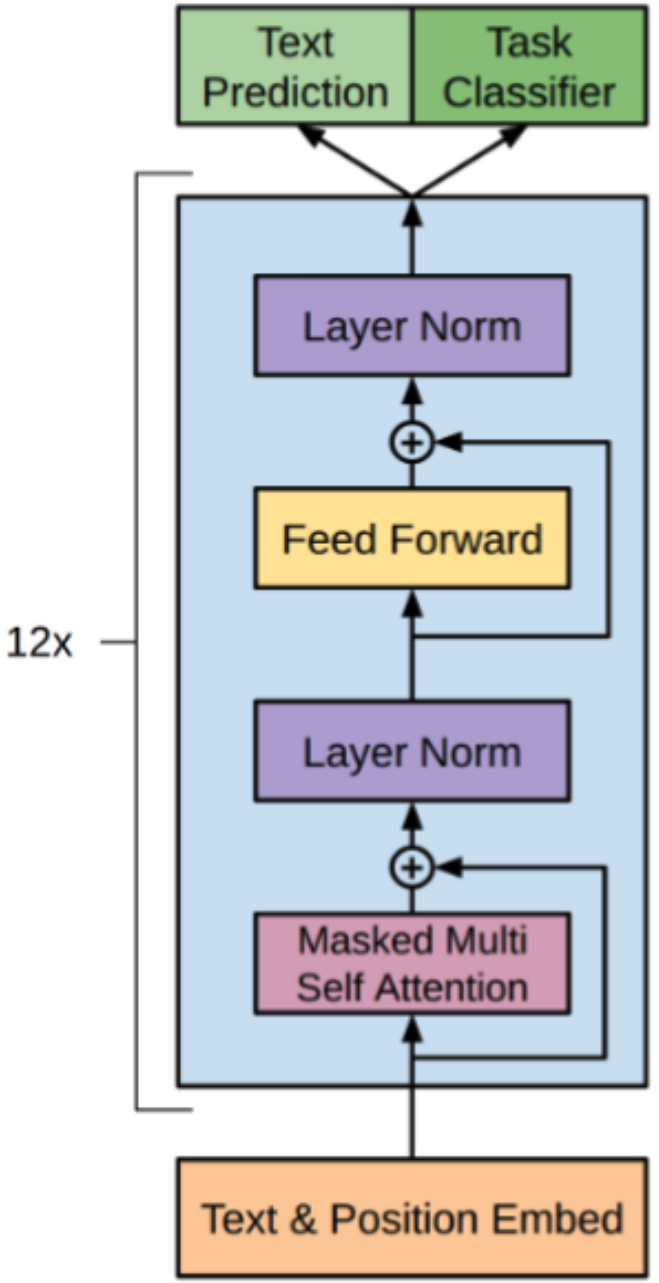


https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

# GPT

- Il modello GPT originale è uno *stack* di 12 blocchi *transformer decoder*.

- Come in BERT, la FFN (*Feed-Forward Network*) alla fine di ogni blocco utilizza l'attivazione GeLU (*Gaussian Error Linear Unit*).




https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

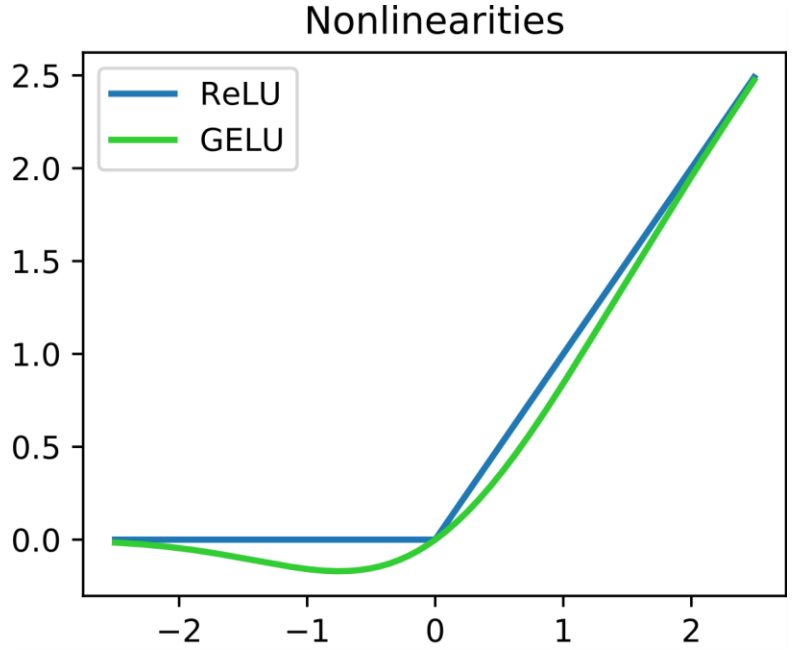
# Funzione di attivazione GeLU

- La Gaussian Error Linear Unit (GeLU) è una variante di ReLU in cui la distribuzione cumulativa gaussiana standard viene utilizzata per modulare l'attivazione lineare.

$$\mathrm{GELU}(x) = xP(X \leq x)$$

$$X \sim \mathcal{N}(0, 1)$$

$$\mathrm{GELU}(x) \sim x\sigma(1.702x)$$



# GPT pretraining non supervisionato

- Dato un set di token $\mathcal{U} = \{u_1, \ldots, u_n\}$

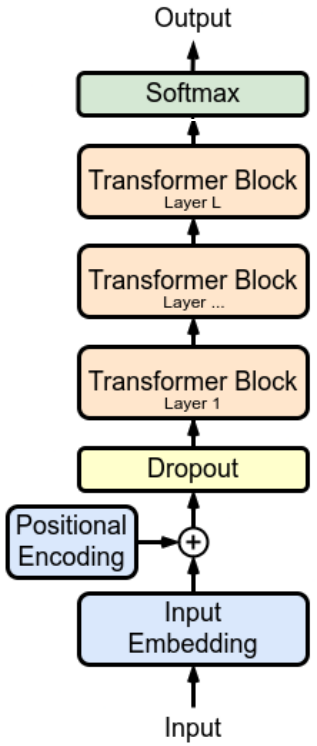
*Token embedding*

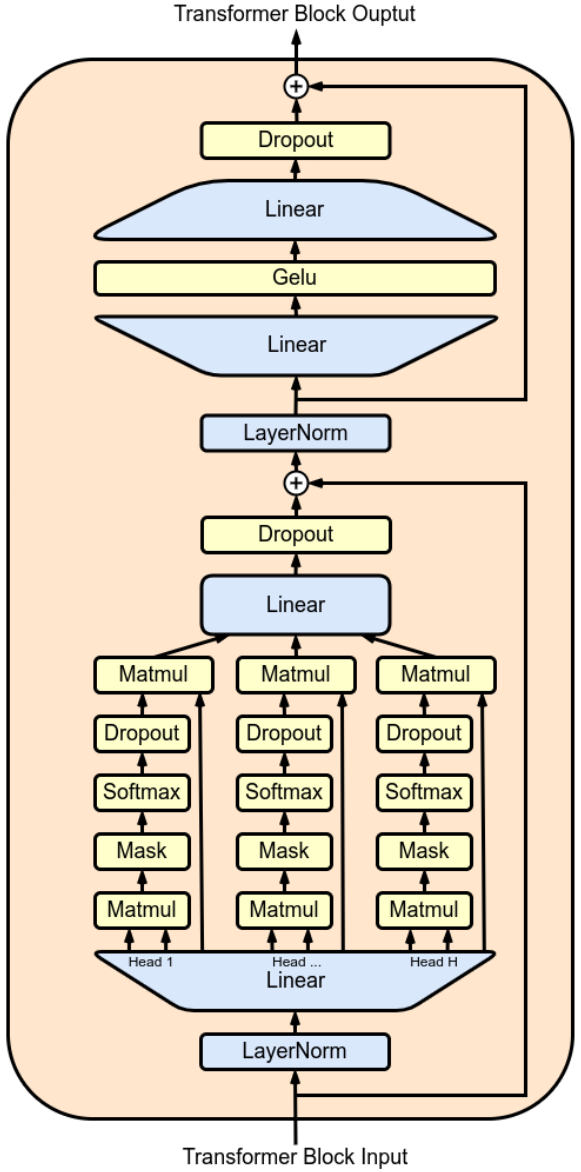
*Position embedding*

$$h_0 = UW_e + W_p$$

$$h_l = \texttt{transformer\_block}(h_{l-1}) \forall i \in [1, n]$$

$$P(u) = \texttt{softmax}(h_n W_e^T)$$




dove $u=(u_{-k}, \ldots, u_{-1})$ è il contesto

https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

# GPT pretraining non supervisionato

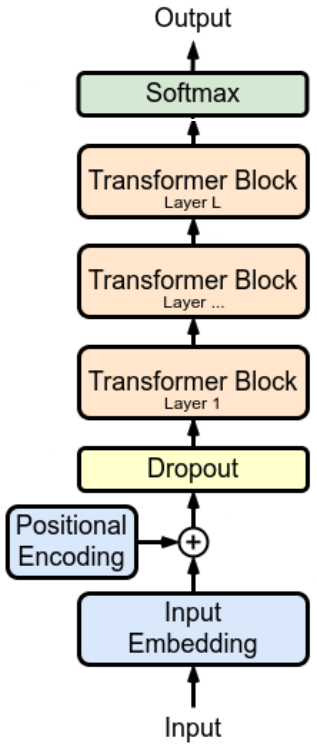
$$h_0 = UW_e + W_p$$

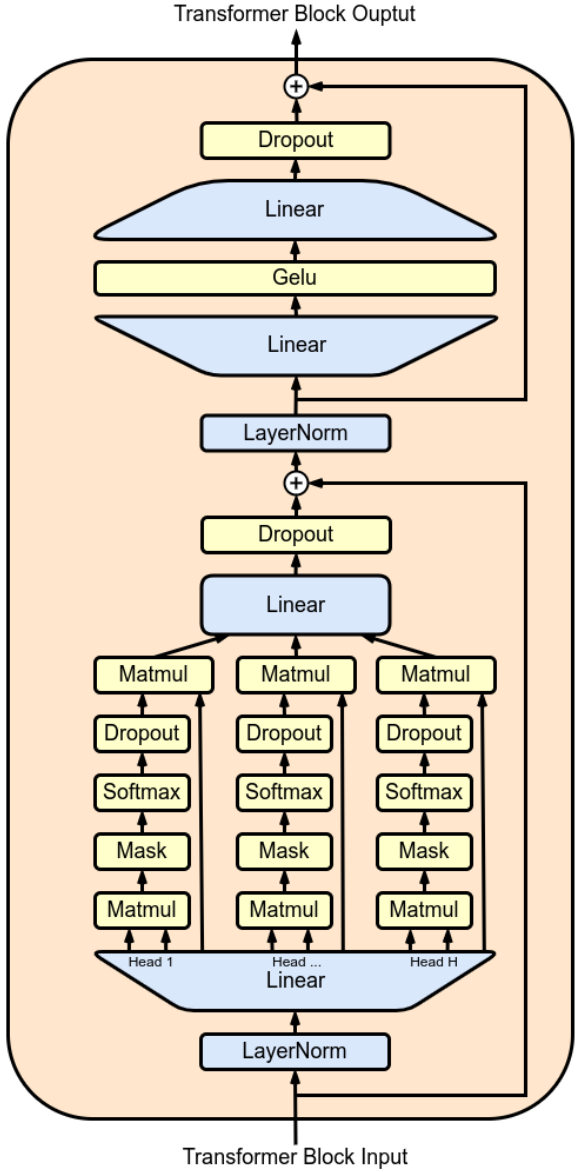
$$h_l = \texttt{transformer\_block}(h_{l-1}) \forall i \in [1, n]$$

$$P(u) = \texttt{softmax}(h_n W_e^T)$$

- Funzione di loss

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$




https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

# GPT fine-tuning supervisionato

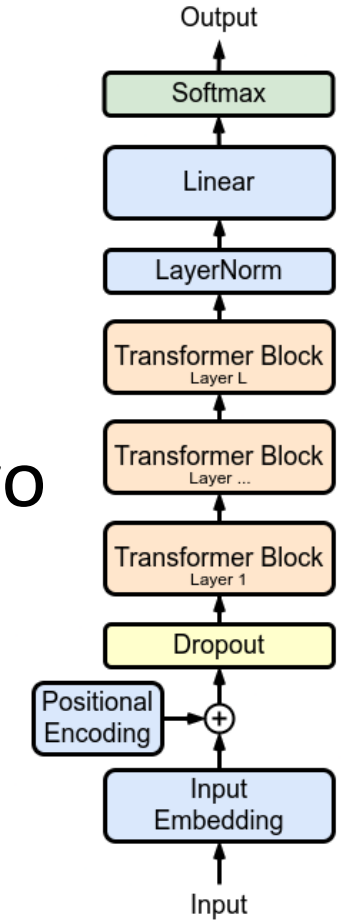
- Viene aggiunto un layer lineare in cima per predire un'etichetta *y* da un set di token $\{x_1, \ldots, x_n\}$

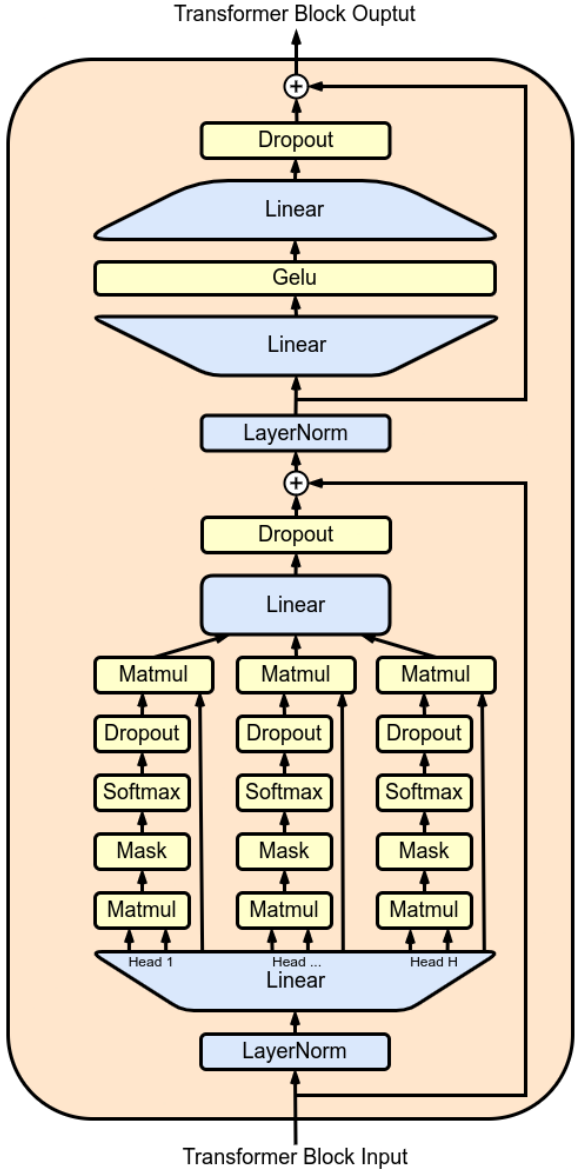
$$P(y|x^1, \ldots, x^m) = \texttt{softmax}(h_l^m W_y).$$

- Viene utilizzata una funzione obiettivo composita:

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m)$$

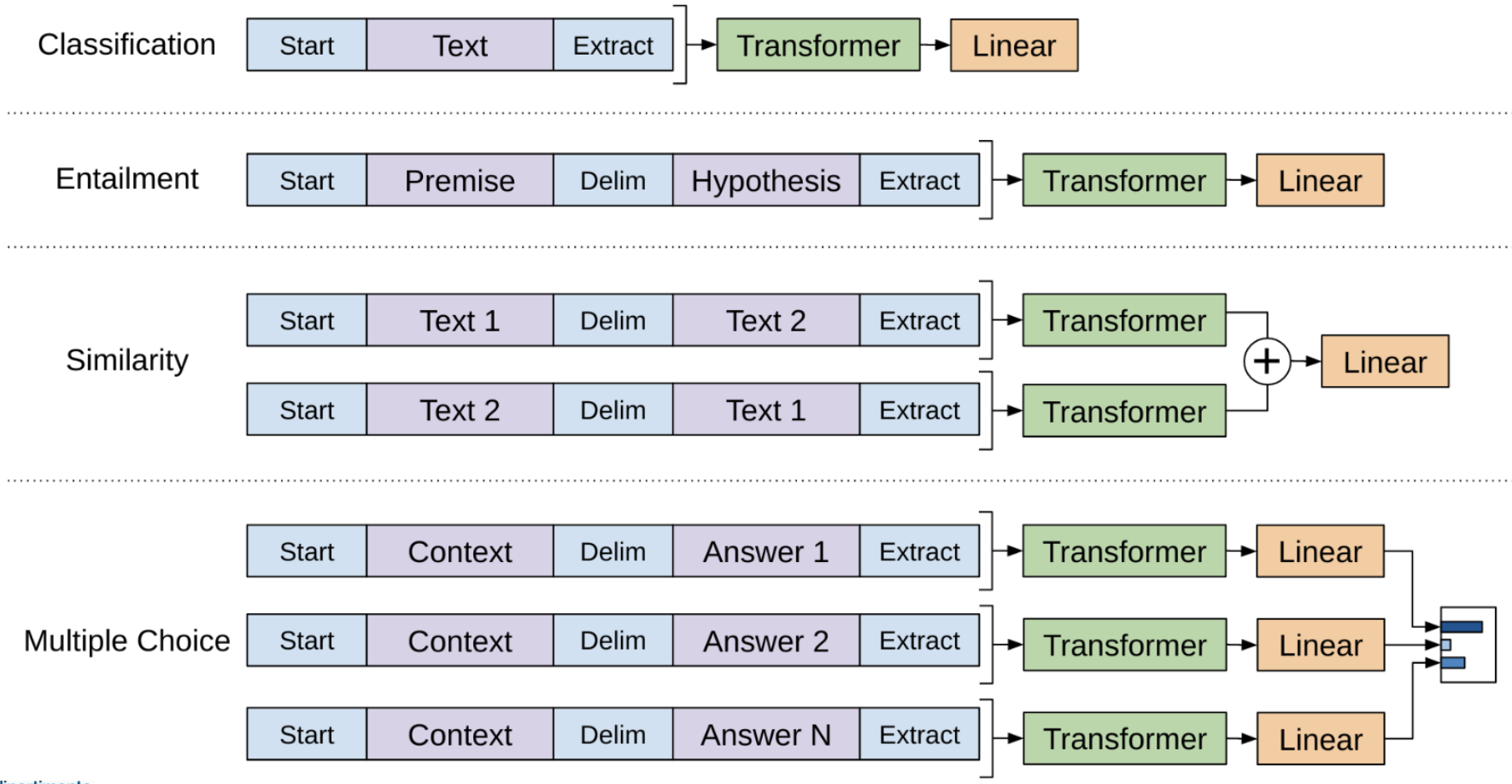
$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$




https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

# GPT task di fine tuning




https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

# Famiglie GPT

| Model | Release | # Params | # Context | Training data | Architettura |
|---|---|---|---|---|---|
| **GPT -1** | 2018 | 117M | | BooksCorpus ~4.5 GB | Decoder-only w/ 12 Attention head x 12 layer + linear softmax |
| **GPT-2** | 2019 | 1.5B | | WebText ~40 GB di testo da ~ 8 M di documenti | GPT-1, normalizzazione modificata |
| **GPT-3** | 2020 | 175B | 2,048 | ~499 B tokens da CommonCrawl (570 GB), WebText, Wikipedia, Books | GPT-2, esteso a scala maggiore |
| **GPT-3.5** | 2022 | ~175B | ~16k max | Human feedback | RLHF |
| **GPT-4** | 2023 | ~1.7T | 8,192 | Dati multimodali | Multimodale |
| **GPT-5** | 2025 | | 400k | | Thinking mode |



https://openai.com/it-IT/

# ChatGPT

- Basato su GPT-3.5 e GPT-4
- Sottoposto a fine tuning per mirare all'uso conversazionale.
- Addestrato con Reinforcement Learning from Human Feedback (RLHF).

https://openai.com/it-IT/

# GPT-omni

- GPT-o è una versione avanzata dei modelli linguistici della serie GPT sviluppata da OpenAI.

- È progettato per offrire risposte più contestuali, coerenti e integrate con informazioni aggiornate.

- Un modello omni è *multilingua* e *multimodale* capace di processare e generare testo, immagini e audio

https://openai.com/it-IT/index/hello-gpt-4o/

# GPT-omni

- GPT è stato addestrato su un data set statico (cutoff date)
- GPT-omni integra la capacità di ragionamento (thinking) ottimizzata e aggiornamenti dinamici tramite cui può accedere a nuove informazioni.
- La funzionalità Deep Research sfrutta il ragionamento per cercare, interpretare e analizzare grandi quantità di testi, immagini e PDF online, adattando il suo approccio in base alle informazioni che trova.



https://openai.com/it-IT/index/hello-gpt-4o/

# Embedding GPT

- Basandosi sui modelli della famiglia GPT, sono stati sviluppati i modelli per la generazione di embedding universali per testo, codice e contenuti multimodali
- Altamente ottimizzati per clustering, ricerca semantica e Retrieval-Augmented Generation (RAG)

| Modello | Max Input | MTEB benchmark |
|---|---|---|
| **text-embedding-3-small** | 8192 | 62.3% |
| **text-embedding-3-large** | 8192 | 64.6% |
| **text-embedding-ada-002** | 8192 | 61.0% |

https://openai.com/it-IT/

# GPT-oss

- OpenAI ha rilasciato i pesi dei modelli GPT-oss
  - gpt-oss-120b
  - gpt-oss-20b

- Addestrati con tecniche di RL e ispirati ai modelli closed-source di OpenAI
- Thinking mode, CoT, few-shot
- Ottimizzazione per l'esecuzione su hardware di consumo.

https://openai.com/it-IT/index/introducing-gpt-oss/?utm_source=chatgpt.com

# Claude

- Claude è una famiglia di LLM sviluppati da Anthropic. Il primo modello è stato rilasciato nel Marzo 2023.
- I modelli Claude sono generative pre-trained transformers che sono stati anche sottoposti a fine-tuning, in particolare utilizzando *Constitutional AI* e RLHF.
- Il nome prende ispirazione da Claude Shannon, il padre della teoria dell'informazione

https://arxiv.org/abs/2212.08073

# Claude

- L’obiettivo è quello di addestrare dei modelli che si allineino all’etica e ai valori umani e quindi agire con onestà, senza recare danno e per essere utili e limitando l’uso di un'estesa human feedback.

- Il metodo prevede due fasi: supervised learning e reinforcement learning.

- I modelli sono stati addestrati su dati disponibili in rete e dati privati

https://arxiv.org/abs/2212.08073

# Claude

- Nella fase di addestramento supervisionato, il modello genera le risposte ai prompt, auto-critica le risposte generate in base ad un insieme di principi guida, una vera e prpria «costituzione» (Constitutional AI) e revisiona le risposte.

- Il modello viene quindi sottoposto a fine tuning su queste risposte riviste.

# Claude

- Nella fase di *Reinforcement Learning from AI Feedback* (RLAIF) le risposte generate e una IA confronta la loro conformità con la costituzione.
- Questo dataset di AI feedback viene utilizzato per addestrare un preference model che valuta le risposte in base a quanto soddisfano (si *allineano* alla) la costituzione.
- La costituzione di Claude include 75 punti, in cui sono incluse sezioni della Dichiarazione Universale dei Diritti Umani

# Claude

Le idee chiave alla base della Constitutional AI sono:

- Allineare il comportamento di un AI con una constitution definita da principi umani, come evitare danni, rispettare le preferenze e fornire informazioni veritiere.
- La constitution modella il modo in cui l'AI agisce.
- Utilizzare tecniche come la self-supervision e l'adversarial training affinché l'AI impari a comportarsi in conformità con la sua constitution senza etichettatura esplicita.

# Claude

Le idee chiave alla base della Constitutional AI sono:

- Sviluppare tecniche di *ottimizzazione vincolata* in modo che l'AI persegua l'utilità all'interno dei suoi vincoli constitutional, anziché classiche funzioni obiettivo.

- Progettare il data set di addestramento e le architetture per codificare comportamenti benefici ed evitare comportamenti non sicuri o ingannevoli.

# Claude

| Name | Release | # Context token | Dettagli |
|---|---|---|---|
| **Claude** | 2023 | 9k | Decoder-only, RLHF, Constitutional AI |
| **Claude 2** | 2023 | 100k | PDF in input |
| **Claude 3 Haiku** | 2024 | 200k | Immagini in input, multilingua, reasoning |
| **Claude 3.5 Sonnet** | 2024 | 200k | Immagini in input, multilingua, reasoning |
| **Claude 3.7 Opus** | 2024 | 200k | Immagini in input, multilingua, reasoning |
| **Claude Haiku 4.1** | 2025 | 200k | Coding |
| **Claude Sonnet 4.5** | 2025 | 200k | Coding |
| **Claude Opus 4.5** | 2025 | 200k | Coding |

https://www.anthropic.com/research

# Gemini

- Gemini è una famiglia di LLM MoE multimodali sviluppati da Google DeepMind, successori di LaMDA e PaLM 2.

- Sono modelli transformer decoder-only, con modifiche per consentire addestramenti e inferenza efficienti sulle TPU.

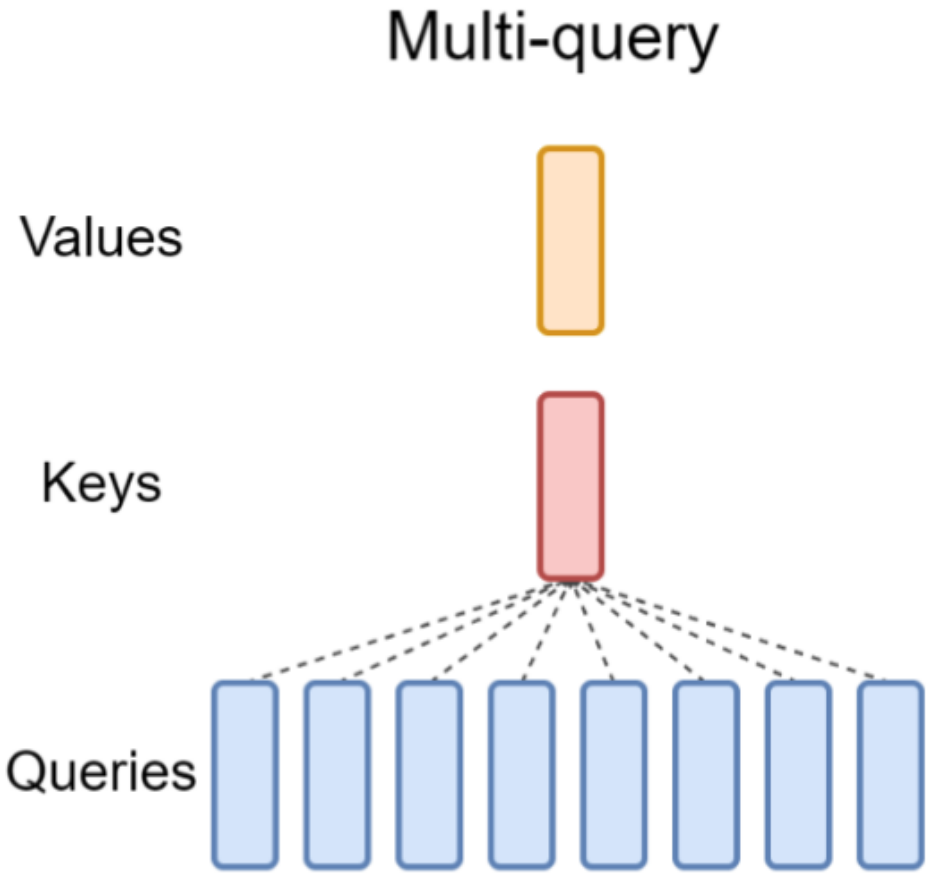
- Hanno una lunghezza di contesto di 32,768 token, con *multi-query attention*.



https://arxiv.org/abs/2312.11805

# Multi-Query Attention (MQA)

- La Multi-Query Attention (MQA) è una variante della multi-head attention
- Mentre il numero di head per **Q** rimane invariato rispetto alla multi-head attention, una sola head viene mantenuta per **K** e **V**.
- Consente inferenze più veloci
  - KV cache più piccola
  - Meno operazioni matriciali




https://pub.towardsai.net/multi-query-attention-explained-844dfc4935bf

# Gemini 1.0 – 2023

Sono tutti modelli multimodali che accettano diverse tipologie di input

- Gemini 1.0 Ultra
- Gemini 1.0 Pro
- Gemini 1.0 Nano, progettato per *compiti su dispositivo*.

- Gemini 1.5 Pro
- Gemini 1.5 Flash

https://arxiv.org/abs/2312.11805

# Gemini 2.0 – 2024

- Gemini 2.0 Flash, Sviluppato da Google con un focus su multimodalità generativa, capacità agentiche e velocità.

- Gemini 2.0 Flash-Lite, primo modello progettato per l'efficienza dei costi e la velocità.

# Gemini 2.5 – 2025

- Supportano l’uso di tool
  - Canva, DeepSearch, Thinking

- Gemini 2.5 Pro
- Gemini 2.5 Flash
- Gemini 2.5 Flash-Lite
- Gemini 2.5 Flash Image (Nano Banana AI)

https://storage.googleapis.com/deepmind-media/gemini/gemini_v2_5_report.pdf

# Gemma

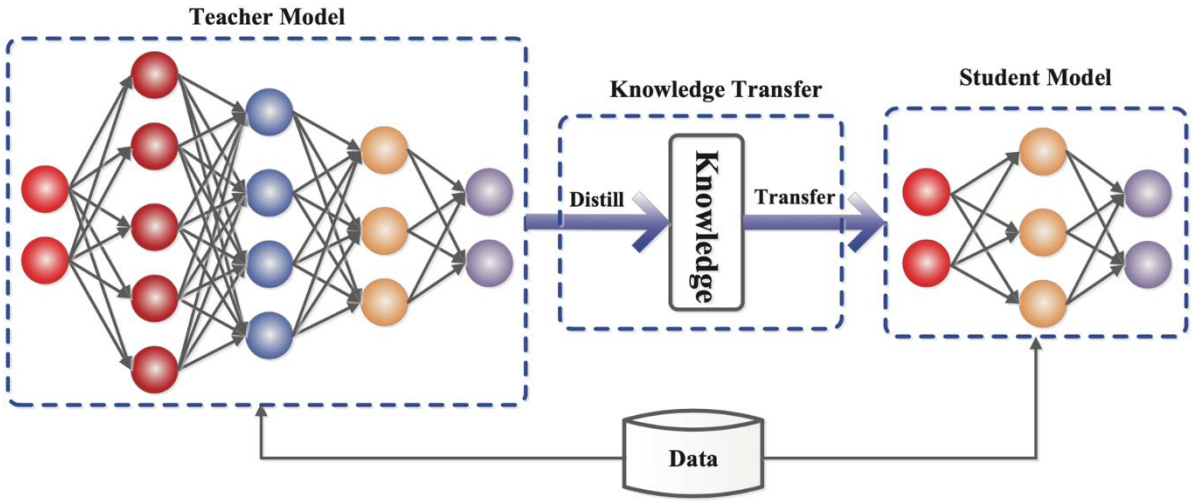
- Gemma è una famiglia di modelli open-source, leggeri e all'avanguardia
- Sono stati sviluppati da Google DeepMind usando la stessa ricerca e tecnologia utilizzate per creare i modelli Gemini (distilled models).
- I modelli sono disponibili in due versioni:
  - Preaddestrato (PT): versione fondazionale di Gemma.
  - Ottimizzato per le istruzioni (IT): versione di Gemma Instruction-tuned, capaci di rispondere a input conversazionali (chatbot).



https://ai.google.dev/gemma/docs?hl=it

# Distilled Model

- Un modello "distilled" (distillato) si riferisce a un modello più piccolo e leggero (o *student model*) che è stato addestrato per replicare il comportamento e le prestazioni di un modello più grande e complesso, noto come "modello insegnante" (o *teacher model*).

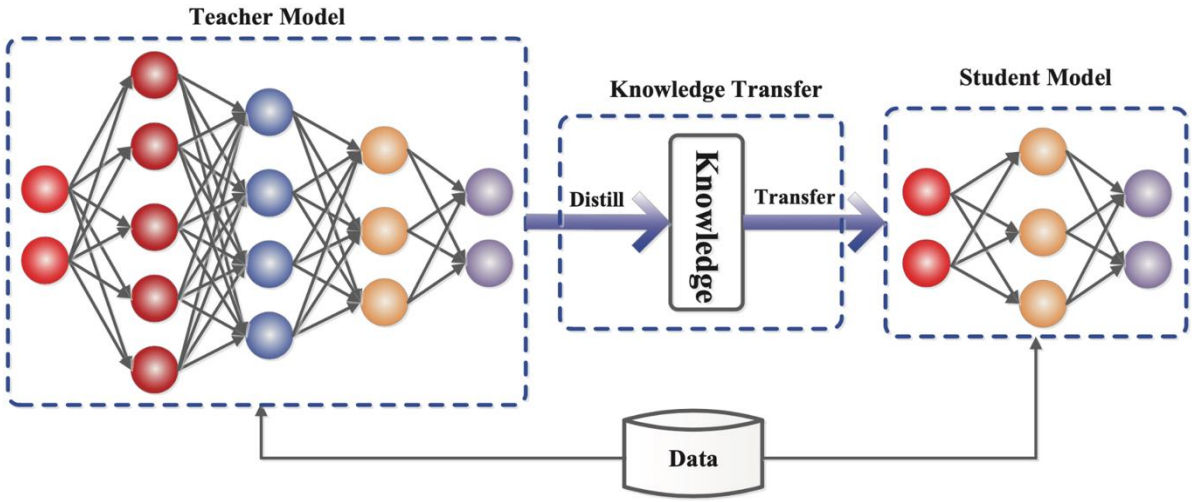



https://humanloop.com/blog/model-distillation

# Distilled Model

- Il modello distillato apprende anche i «soft target» del modello insegnante *ovvero le sue softmax*

- Una apposita *distillation loss* penalizza le softmax del modello studente se differiscono da quelle del modello insegnante




https://humanloop.com/blog/model-distillation

# Gemma 1, Gemma 2

- Modello text-to-text in Inglese, in grado di svolgere compiti quali generazione, QA, summarization, reasoning.
- Gemma 1 è stato rilasciato sia come modello PT che IT nelle due versioni 2B e 7B.
- Gemma 2 è stato rilasciato sia come modello PT che IT nelle tre versioni 2B e 9B e 27B.

| | Gemma 1 2B | Gemma 1 7B | Gemma 2 2B | Gemma 2 9B | Gemma 2 27B |
|---|---|---|---|---|---|
| # tokens | 6 Trilioni | 6 Trilioni | 2 Trilioni | 8 Trilioni | 13 Trilioni |



https://ai.google.dev/gemma/docs/core/model_card

# Gemma 3

- I modelli della famiglia Gemma 3 sono nativamente multi-lingua e multi-modali
  - Supporto per più di 140 lingue
  - Contesto di 128k token
  - Consentono lo svolgimento di task multimodali: Image captioning, Visual Question Answering
- Il training set include anche dati visuali e una maggiore quantità di testo in altre lingue

| | Gemma 3 4B | Gemma 3 12B | Gemma 3 27B |
|---|---|---|---|
| # tokens | 4 Trilioni | 12 Trilioni | 14 Trilioni |

https://ai.google.dev/gemma/docs/core/model_card

# LLaMA

- Large Language Model Meta AI (LLaMA) è stato rilasciato da Meta nel Febbraio 2023.
- Versioni con 7, 16, 33 e 65 miliardi di parametri.
- La versione da 13 miliardi di parametri ha riportato performance migliori di GPT-3.
- LLaMA 2 è stato rilasciato nel Luglio 2023 con versioni da 7, 13 e 70 miliardi di parametri.

# LLaMA

| Name | Release | # Params | # Context token | Training data | Architettura |
|---|---|---|---|---|---|
| **LLaMA** | 2023 | 7B, 13B, 33B, 65B | 2048 | 1.4T | Decoder-only |
| **LLaMA 2** | 2023 | 7B, 13 B, 69B | 4096 | 2T | SFT, RLHF, RoPE |
| **LLaMA 3** | 2024 | 8B, 71B | 8192 | 15T | Grouped-Query Attention (GQA) |
| **LLaMA 3.1** | 2024 | 8B, 71B 405B | 128K | 15T | Multilingual |
| **LLaMA 3.2** | 2024 | 1B, 3B, 11B, 90B | 128K | 15T | Rejection Sampling (RS), Direct Preference Optimization (DPO) |
| **LLaMA 4** | 2025 | 17B x 16E, 17B x 128E | 1M, 10M | 22T, 40T | MoE, multimodal |

https://www.llama.com/docs/model-cards-and-prompt-formats/

# LLaMA

- LLaMA è stato addestrato con 1.4 trilioni di tokens
  - Pagine web scraped da CommonCrawl.
  - Repository open-source di codice sorgente da GitHub.
  - Wikipedia in 20 lingue diverse.
  - Libri di dominio pubblico da Project Gutenberg.
  - Il codice sorgente LaTeX per articoli scientifici caricati su ArXiv.
  - Domande e risposte dai siti web di Stack Exchange.



https://www.llama.com/docs/model-cards-and-prompt-formats/

# LLaMA 2

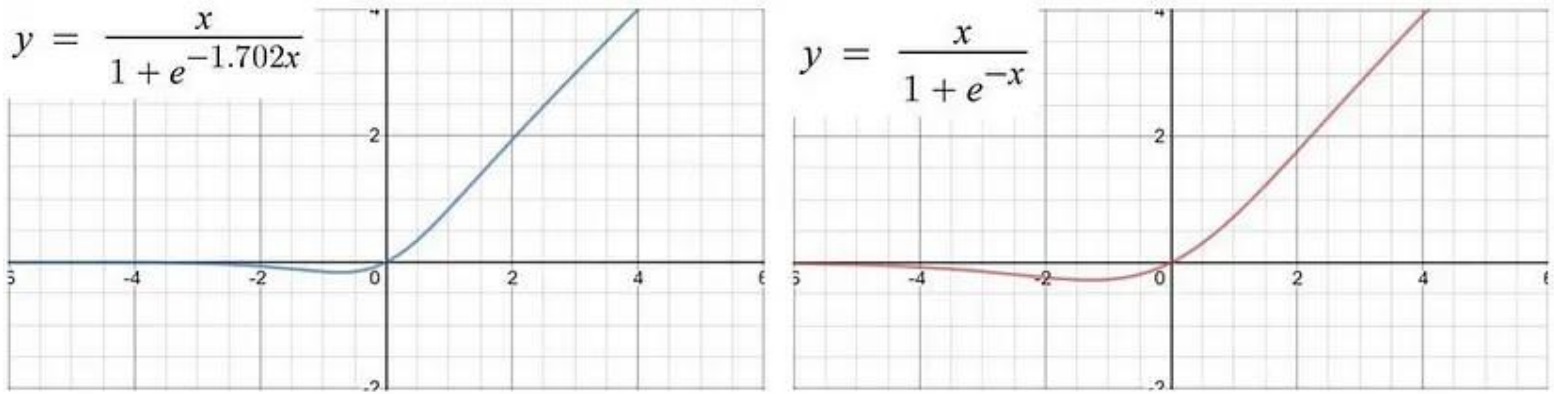
- Basato su stack di transformer decoder.
- Differenze architettoniche minori rispetto a GPT-3:
  - Funzione di attivazione *SwiGLU* invece di ReLU
  - *Rotary Positional Embeddings* (ROPE)
  - *Root Mean Squared Normalization* (RMSNorm)
- Aumenta la lunghezza del contesto a 4K token.

https://www.llama.com/docs/model-cards-and-prompt-formats/

# SwiGLU activation function

- Swish Gated Linear Unit (SwiGLU) ha una funzione di attivazione che è la combinazione di *Swish* e *Gated* LUs.
- Le Swish LU sono una generalizzazione dell'approssimazione GeLU.

$$\text{Swish}(x) = x \cdot \sigma(\beta x)$$



# SwiGLU activation function

- Le Gated LUs incorporano un'attivazione lineare all'interno della funzione sigmoid.

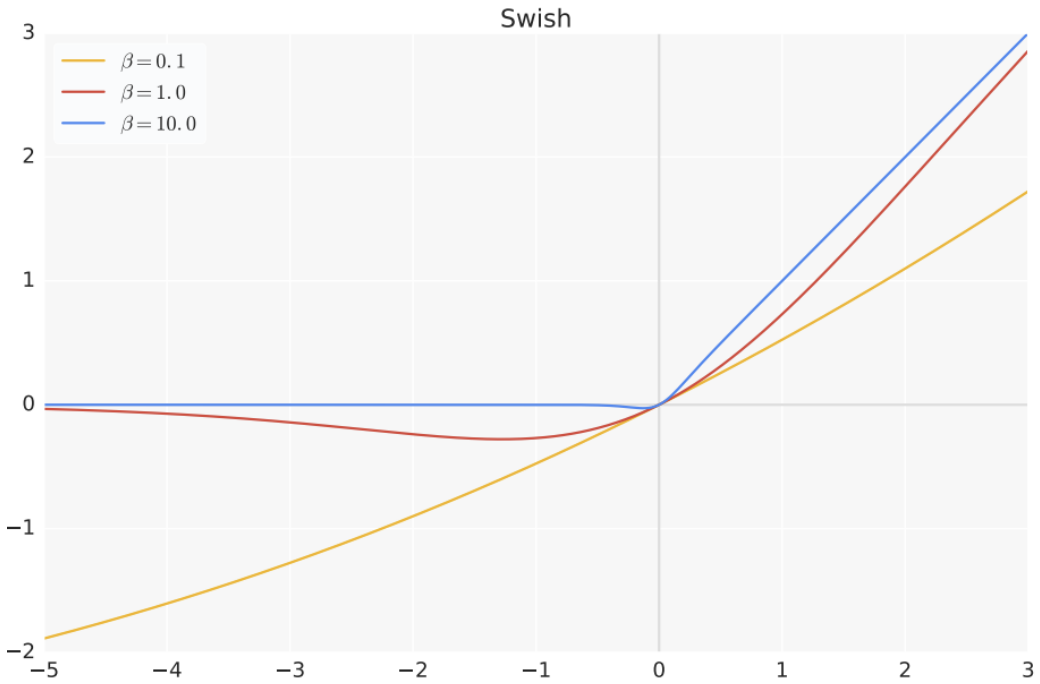
$$\mathrm{GLU}(x) = \sigma(Wx + b)$$

- Meccanismo di gating: il neurone viene attivato in base all'input che riceve.

# SwiGLU activation function

- La SwiGLU incorpora le attivazioni precedenti.

$$\mathrm{SwiGLU}(x) = x \cdot \sigma(\beta x) + (1 - \sigma(\beta x)) \cdot \sigma(W \cdot x + b)$$



# Rotary Positional Embedding (RoPE)

- Rotary Positional Embedding (RoPE) codificano la *posizione relativa* tra due vettori della self-attention

- I vettori di self-attention **q**, **k** e **v** possono essere rappresentati in termini dei loro embedding

$$\boldsymbol{q}_m = f_q(\boldsymbol{x}_m, m)$$
$$\boldsymbol{k}_n = f_k(\boldsymbol{x}_n, n)$$
$$\boldsymbol{v}_n = f_v(\boldsymbol{x}_n, n),$$

- L'elemento (*m*, *n*) della attention matrix, cioè il *prodotto interno* $\mathbf{q}_m\mathbf{k}_n$ è formulato in termini di posizione relativa $m - n$ come funzione $g(\boldsymbol{x}_m, \boldsymbol{x}_n, m - n)$

Università degli Studi di Palermo

dipartimento di ingegneria unipa

https://krasserm.github.io/2022/12/13/rotary-position-embedding/

CHILAB

# Rotary Positional Embedding (RoPE)

- RoPE esprime $f_q(\boldsymbol{x}_m, m)$, $f_k(\boldsymbol{x}_n, n)$ e $g$ in termini di matrici di rotazione:

$$
\begin{aligned}
f_q(\mathbf{x}_m, m) &= \mathbf{R}_{\theta,m}\mathbf{W}_q\mathbf{x}_m = \mathbf{q}_m \\
f_k(\mathbf{x}_n, n) &= \mathbf{R}_{\theta,n}\mathbf{W}_k\mathbf{x}_n = \mathbf{k}_n \\
g(\mathbf{x}_m, \mathbf{x}_n, n-m) &= \mathbf{x}_m^\top\mathbf{W}_q^\top\mathbf{R}_{\theta,n-m}\mathbf{W}_k\mathbf{x}_n = \mathbf{q}_m^\top\mathbf{k}_n
\end{aligned}
$$

dove $\mathbf{R}_{\theta,n-m} = \mathbf{R}_{\theta,m}^\top\mathbf{R}_{\theta,n}$

# Rotary Positional Embeddings (RoPE)

- In *d* dimensioni il vettore Θ è una sequenza di step angolari costanti

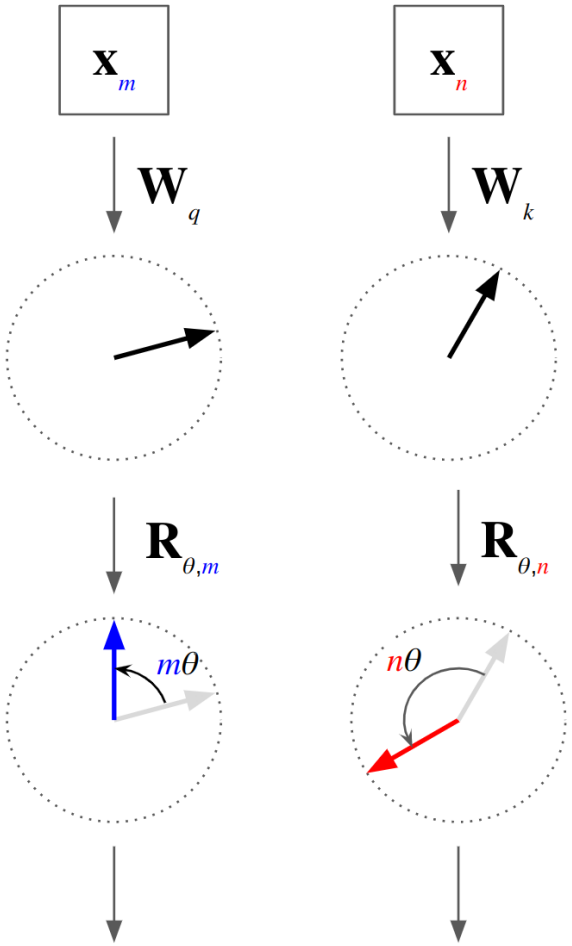
$$\Theta = \{\theta_i = 10000^{-2(i-1)/d}, i \in [1, 2, \ldots, d/2]\}$$

- La matrice di rotazione per un generico indice *t* è

$$\mathbf{R}^d_{\Theta,t} = \begin{pmatrix} \cos t\theta_1 & -\sin t\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ \sin t\theta_1 & \cos t\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos t\theta_2 & -\sin t\theta_2 & \cdots & 0 & 0 \\ 0 & 0 & \sin t\theta_2 & \cos t\theta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos t\theta_{d/2} & -\sin t\theta_{d/2} \\ 0 & 0 & 0 & 0 & \cdots & \sin t\theta_{d/2} & \cos t\theta_{d/2} \end{pmatrix}$$

https://krasserm.github.io/2022/12/13/rotary-position-embedding/

# Rotary Positional Embeddings (RoPE) caso 2D




Token representations at positions $m$ and $n$

Non-rotated query and key (no position information)

$$\mathbf{R}_{\theta,t} = \begin{pmatrix} \cos t\theta & -\sin t\theta \\ \sin t\theta & \cos t\theta \end{pmatrix}$$

Rotated query and key (*absolute* position information)

$$\mathbf{q}^{\mathsf{T}}_m \mathbf{k}_n = g(\mathbf{x}_m, \mathbf{x}_n, n-m)$$

Inner product of query and key (*relative* position information)

https://krasserm.github.io/2022/12/13/rotary-position-embedding/

# Root Mean Squared Normalization (RMSNorm)

- Sostituisce la LayerNorm nei transformer

$$\hat{x} = \frac{x - \mu}{\sigma + \epsilon} \longrightarrow \hat{x} = \frac{x}{RMS(x) + \epsilon}$$

$$RMS(x) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$$

- Minore onere computazionale
- Stabilità nel training

https://medium.com/@hugmanskj/mastering-llama-rmsnorm-ae5c5d504e9a

# LLaMA 2

- LLaMA 2 è stato addestrato con 2 trilioni di tokens ed è stato sottoposto a finetuning su 27.540 coppie prompt-risposta.

- È stato utilizzato RLHF con rejection sampling e Proximal policy optimization (PPO), una tecnica di RL dove gli aggiornamenti ad ogni epoca sono effettuati in maniera conservativa per non perdere stabilità.



https://www.llama.com/docs/model-cards-and-prompt-formats/

# LLaMA 2 – special tokens

- **<s></s>**
  - I BOS (Begin Of Sentence) e EOS (End Of Sentence) tokens.
  - Sono usati per separare le domande e risposte in una conversazione multi-turn.

- **[INST][/INST]**
  - Token che incapsulano i messaggi dell'utente in conversazioni multi-turn.

- **<<SYS>><</SYS>>**
  - Token che incapsulano i messaggi per il modello (system prompt).

https://www.llama.com/docs/model-cards-and-prompt-formats/

# LLaMA 2 – prompt

```
<s>[INST]
<<SYS>>{{ system_prompt }}<</SYS>>

{{ user_message }} [/INST]
    ----------------------------------------------------------------------------
<s>[INST] <<SYS>>
{{ system_prompt }}
<</SYS>>

{{ user_message_1 }} [/INST] {{ model_answer_1 }} </s>
<s>[INST] {{ user_message_2 }} [/INST]
```

https://www.llama.com/docs/model-cards-and-prompt-formats/

# LLaMA 3

- LLaMA 3 è stato addestrato utilizzando principalmente dati in inglese, con oltre il 5% in più di 30 altre lingue.

- Il suo dataset è stato filtrato da un classificatore di qualità del testo, e il classificatore è stato addestrato con testo sintetizzato da LLaMA 2.

https://www.llama.com/docs/model-cards-and-prompt-formats/

# LLaMA 3 – special tokens

| Special token | Description |
|---|---|
| <\|begin_of_text\|> | Specifica l'inizio del prompt. |
| <\|end_of_text\|> | Ultimo token generato, indica la fine del testo generato. |
| <\|finetune_right_pad_id\|> | Token utilizzato per il padding di sequenze ai fini di finetuning per batch. |
| <\|start_header_id\|><br><\|end_header_id\|> | Token che racchiudono l'indicazione del ruolo per un dato messaggio. I possibili ruoli sono: [system, user, assistant, and ipython] |
| <\|eom_id\|> | Fine del messaggio, per interazioni multi-step o tool-call. |
| <\|eot_id\|> | Fine del turno, utilizzata al termine di un messaggio e il turno passa ad un altro interlocutore. |
| <\|python_tag\|> | Tag per indicare una chiamata ad un tool. |

https://www.llama.com/docs/model-cards-and-prompt-formats/

# LLaMA 3 – roles

LLaMA 3 supporta 4 diversi ruoli:

- **System**
  - Imposta in contesto in cui interagire con il modello IA. Tipicamente include regole, linee-guida o informazioni necessarie che aiutano il modello a rispondere efficacemente.
- **User**
  - Rappresenta l'umano che interagisce con il modello. Include gli input, i comandi e le domande da porre al modello.
- **Ipython**
  - Nuovo ruolo che si riferisce ai tool. Questo ruolo viene usato per segnare gli output provenienti dall'uso di un tool esterno.
- **Assistant**
  - Rappresenta la risposta generata dal modello IA in risposta al contesto fornito nel prompt.

https://www.llama.com/docs/model-cards-and-prompt-formats/

# LLaMA 3

```
<|begin_of_text|>
<|start_header_id|>system<|end_header_id|>
You are a helpful AI assistant for travel tips and
recommendations<|eot_id|>
<|start_header_id|>user<|end_header_id|>
What can you help me with?<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>
```

https://www.llama.com/docs/model-cards-and-prompt-formats/

# LLaMA 3

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>

Cutting Knowledge Date: December 2023
Today Date: 23 July 2024

You are a helpful assistant<|eot_id|>
<|start_header_id|>user<|end_header_id|>
What is the capital of France?<|eot_id|>

<|start_header_id|>assistant<|end_header_id|>
```

https://www.llama.com/docs/model-cards-and-prompt-formats/

# LLaMA 4

- LLaMA 4 è una collezione di modelli MoE pretrained instruction-tuned
- Sono state rilasciate due versioni di grandezza diversa, Llama 4 Scout e Llama 4 Maverick.
- Sono modelli ottimizzati per il multimodal understanding, task multilingua, coding, chiamata di tool e sistemi ad agenti.
- I modelli hanno una knowledge cutoff ad Agosto 2024.

https://www.llama.com/docs/model-cards-and-prompt-formats/

# LLaMA 4

- Multimodalità

- Tutti i modelli LLaMA 4 sono nativamente multimodali, sfruttando una early fusion che permette di eseguire il pretrain con grandi quantità di *token* testuali e visivi non etichettati – un cambio di prospettiva rispetto ai pesi multimodali separati e frozen.

https://www.llama.com/docs/model-cards-and-prompt-formats/

# LLaMA 4

- Contesto molto ampio

- LLaMA 4 Scout supporta fino a 10M token di contesto (uno dei più lunghi disponibili) permettendo l'implementazione di nuovi casi d'uso, personalizzazioni e applicazioni multimodali.

https://www.llama.com/docs/model-cards-and-prompt-formats/

# LLaMA 4

- Image Grounding

- LLaMA 4 risulta essere un ottimo modello nei task visuali e image grounding, capace di allineare i prompt dell'utente con concetti visuali e offrire risposte relative ad aree specifiche dell'immagine.

https://www.llama.com/docs/model-cards-and-prompt-formats/

# LLaMA 4

| Feature | LLaMA 4 Scout | LLaMA 4 Maverick |
|---|---|---|
| **Multimodal** | *Input:* Text + up to 5 images<br>*Output*: Text-only | |
| **Multilingual** | Arabic, English, French, German, Hindi, Indonesian, Italian, Portuguese, Spanish, Tagalog, Thai, and Vietnamese.<br>Image understanding is English-only. | |
| **Active parameter** | 17B | |
| **Total parameters** | 109B | 400B |
| **# of experts** | 16 | 128 |
| **# context tokens** | 10M | 1M |

https://www.llama.com/docs/model-cards-and-prompt-formats/

# Mistral

- Mistral AI è stata co-fondata nell'Aprile 2023 da persone provenienti da Google DeepMind e Meta Platforms.

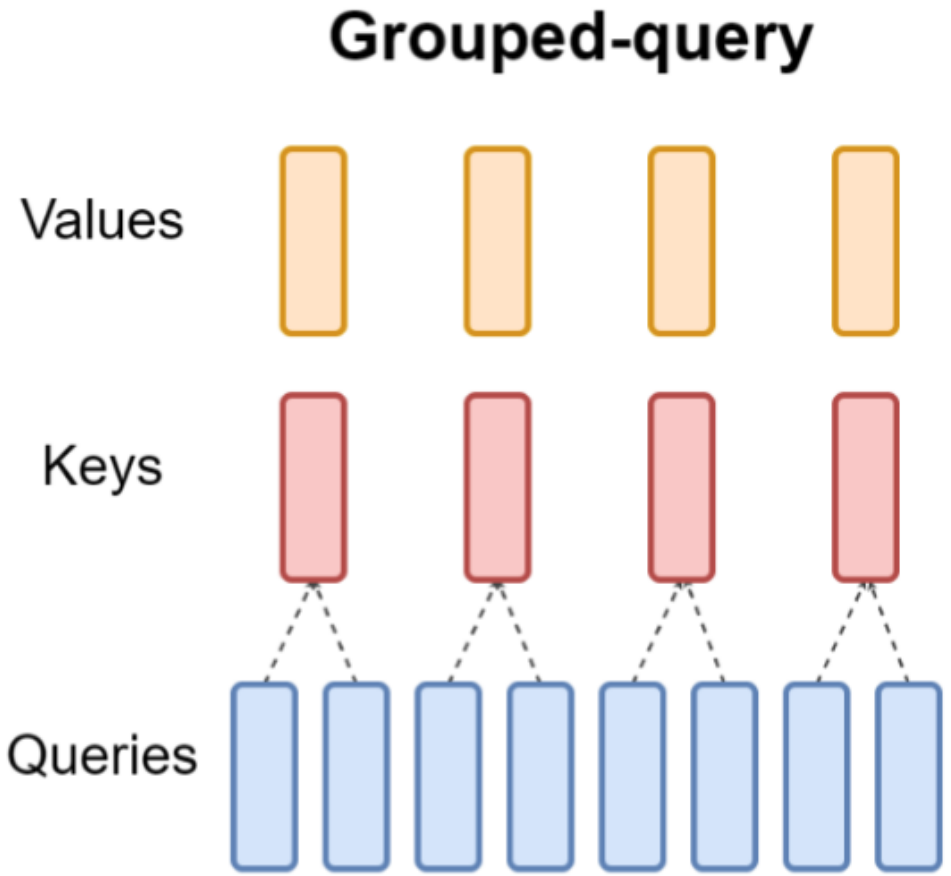
- Rilascia modelli open-weights.

- I modelli mistral utilizzano la *Grouped-Query Attention* (GQA) e la *Sliding Window Attention* (SWA)

https://mistral.ai/news/announcing-mistral-7b

# Grouped-Query Attention (GQA)

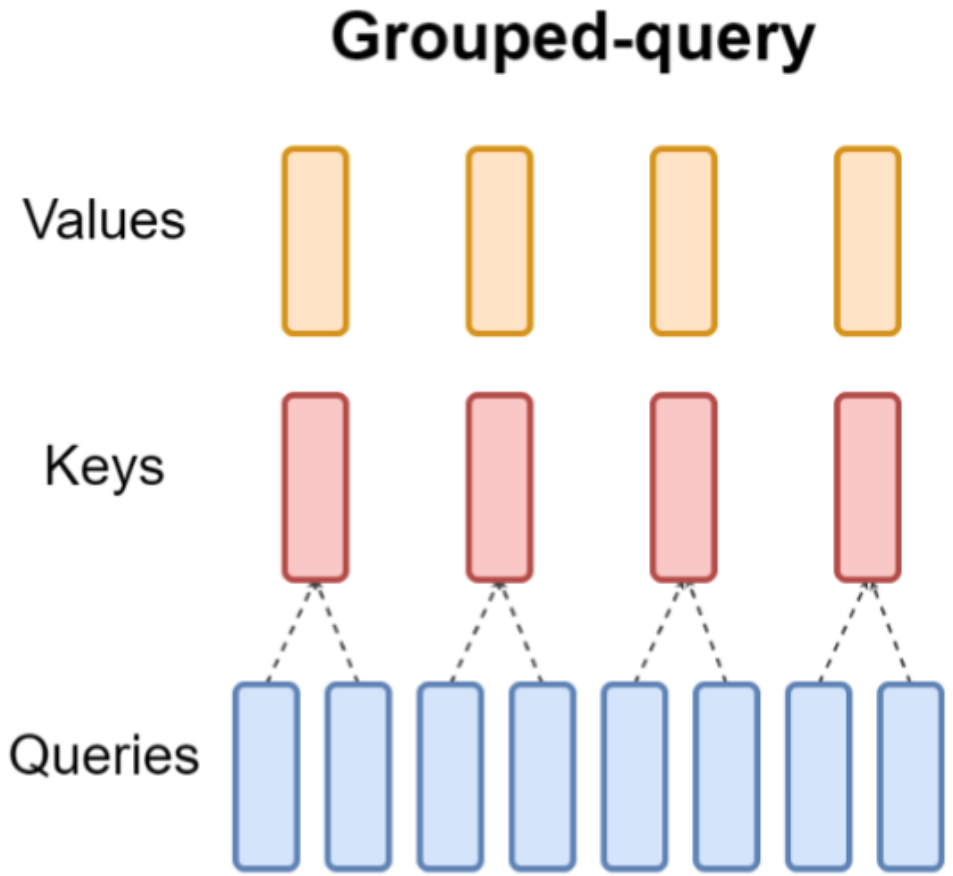
- Nella Grouped-Query Attention (GQA), le head di query sono divise in *G* gruppi, ciascuno dei quali condivide una singola head di value e key.




https://arxiv.org/abs/2305.13245

# Grouped-Query Attention (GQA)

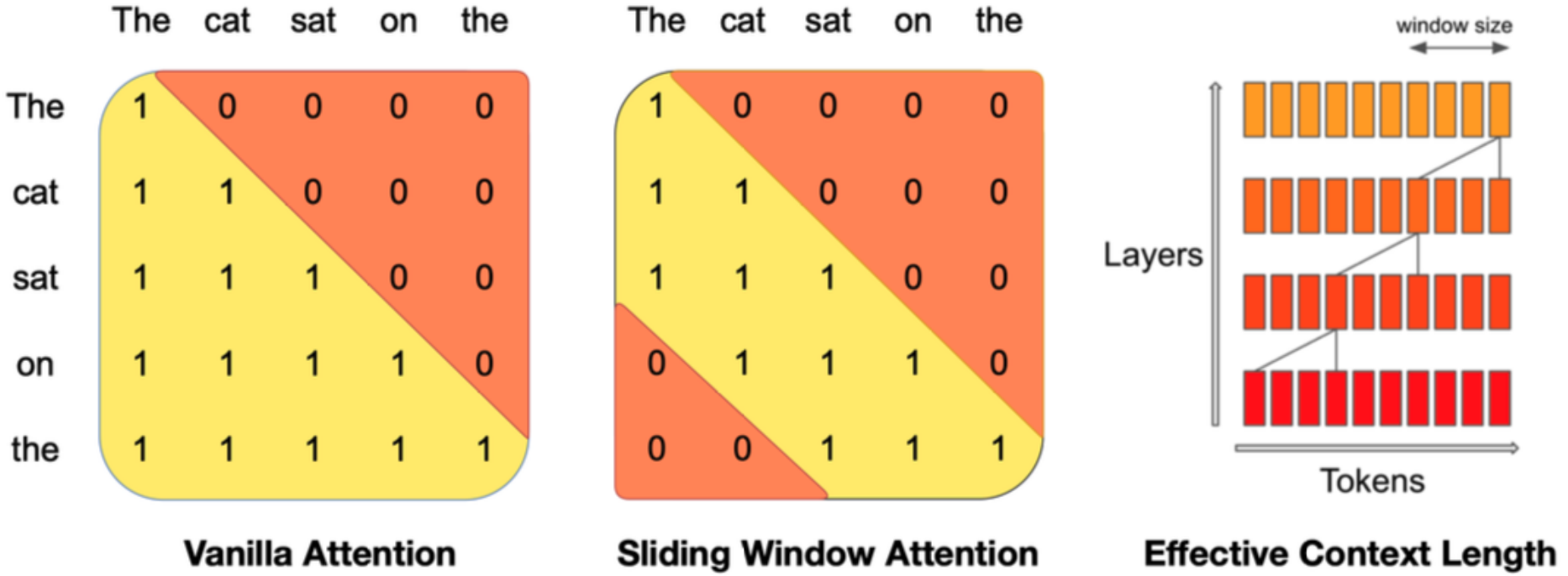
- GQA accelera significativamente la velocità di inferenza
- Riduce la memoria richiesta in fase di decoding,
  - Batch più grandi
  - Maggiore throughput
- Cruciale per applicazioni real-time.




https://arxiv.org/abs/2305.13245

# Sliding Window Attention (SWA)

- La Sliding Window Attention (SWA) consente la gestione di sequenze più lunghe in modo più efficace e con ridotto costo computazionale




https://mistral.ai/news/announcing-mistral-7b

# Mistral

- Mistral 7B è un LLM con 7.3B di parametri che utilizza un architettura di tipo GQA e SWA.
- Mixtral 8x7B: architettura sparse MoE con 8 esperti, che conferiscono al modello un totale di 46.7 miliardi di parametri utilizzabili.
- Mixtral 8x22B: utilizza un'architettura simile a quella di Mistral 8x7B, ma ogni esperto ha 22 miliardi di parametri invece di 7.

https://mistral.ai/models

# Mistral

- Mistral Large 2 123B, è un modello multilingua, addestrato anche per generare codice, supporta la thinking mode.
- Mistral 3.1 Small 24B, è un modello multilingua che consente un input di 128k tokens, è stato rilasciato in versione base e instruct
- Ministral 3B e Ministral 8B sono le versioni più piccole dei modelli di Mistral che possono essere utilizzati in contesti con risorse limitate
- Mistral Embeddings



https://mistral.ai/models

# Phi

- La serie Microsoft Phi è stata introdotta come una linea di Small Language Models (SLMs) altamente efficienti e performanti nonostante le dimensioni ridotte
- Obiettivo: performance di alto livello possono essere raggiunte con un ingombro computazionale significativamente ridotto.
- Ideali per scenari con vincoli di memoria o calcolo (compute-constrained environments) e per applicazioni in cui la latenza è un fattore critico.

https://azure.microsoft.com/en-us/products/phi

# Phi

- Nello sviluppo dei loro modelli, grande attenzione è stata data alla definizione del data set su cui addestrare i modelli
  - Data set disponibili
  - Dati sintetici
  - Data curation, filtering e decontamination

- Sono modelli decoder-only che implementano la classica Multi-Head Attention

# Phi

| Nome | Relase | # Params | Dettagli |
|---|---|---|---|
| Phi 1 | 2023 | 1.3B | Python coding |
| Phi 1.5 | 2023 | 1.3B | Foundational model |
| Phi 2 | 2023 | 2.7B | QA, chat, coding, base model |
| Phi 3 | 2024 | 3.8B, 7B, 14B | Instruct model, SFT, DPO, anche in versione MoE e multimodale, supporto multilingua |
| Phi 4 | 2025 | 3,8B, 14B | Instruct model, pre-training in tre fasi, SFT, DPO, CoT e reasoning |

https://azure.microsoft.com/en-us/products/phi

# DeepSeek

## *How Chinese A.I. Start-Up DeepSeek Is Competing With Silicon Valley Giants*

Jan. 23, 2025, 5:01 a.m. ET

The day after Christmas, a small Chinese start-up called DeepSeek unveiled a new A.I. system that could match the capabilities of cutting-edge chatbots from companies like OpenAI and Google.

That alone would have been a milestone. But the team behind the system, called DeepSeek-V3, described an even bigger step. In a research paper explaining how they built the technology, DeepSeek's engineers said they used only a fraction of the highly specialized computer chips that leading A.I. companies relied on to train their systems.

https://web.archive.org/web/20250123102900/https://www.nytimes.com/2025/01/23/technology/deepseek-china-ai-chips.html

## L'AI Deepseek affossa Nvidia: 589 miliardi bruciati, è la perdita più grande della storia

28 Gennaio 2025 294

Proprio il primato di Nvidia nel settore dell'AI, e cioè il suo punto di forza, si è tramutato nella sua maggiore debolezza con l'emergere di DeepSeek. Il modello AI sviluppato da DeepSeek ha dimostrato che **è possibile raggiungere risultati paragonabili a quelli dei leader del settore, come ChatGPT, con investimenti molto più contenuti**. Tradotto: non serve comprare tutti quei chip Nvidia di ultima generazione che si credevano necessari per proseguire nello sviluppo di un'AI avanzata. E dunque il valore di Nvidia agli occhi degli investitori, fatalmente, cala, nonostante la sua leadership nel settore resti indiscussa.

DeepSeek stessa, ovviamente, fa utilizzo di chip Nvidia (A100, nello specifico) ottenuti prima che scattasse il divieto di esportazione degli Stati Uniti, e li combina con chip meno potenti. La mancanza di risorse ha spinto l'azienda a cercare metodi per continuare nello sviluppo, spingendola quindi a realizzare modelli altamente efficienti, e arrivando così sul mercato con una proposta che rivaleggia con quelle concorrenti ma con costi nettamente inferiori.

https://www.hdblog.it/mercato/articoli/n606821/ai-deepseek-nvidia-589-miliardi-bruciati-perdita/

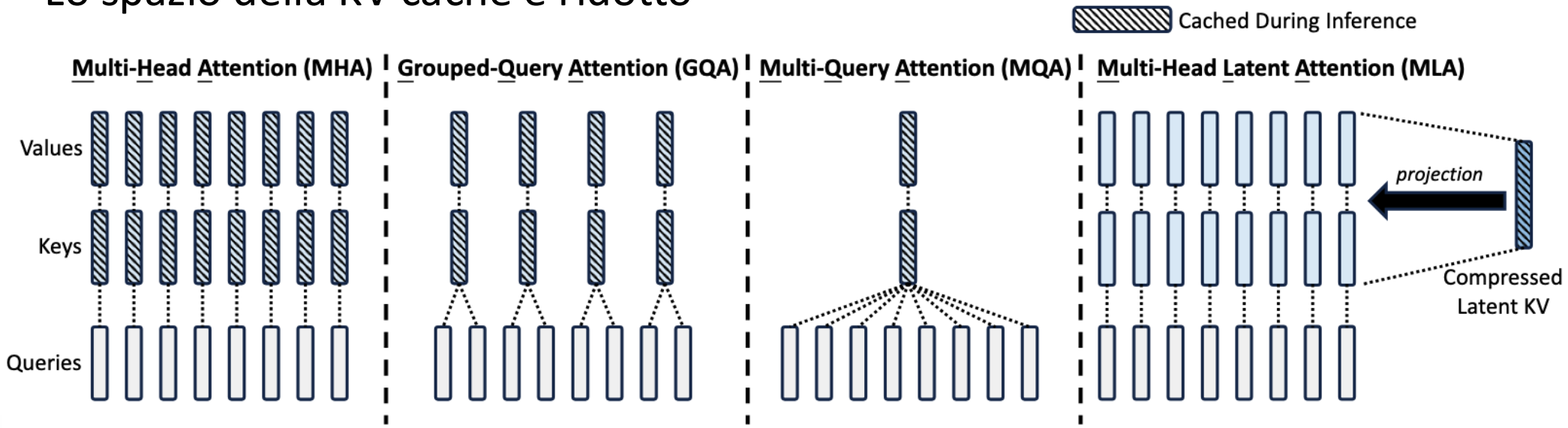
# DeepSeek

- DeepSeek è una compagnia cinese che ha rilasciato i suoi modelli LLM open-source a inizio 2025
- Modelli generativi decoder-only addestrati su GPU con risorse limitate e utilizzando diverse tecniche di ottimizzazione
- Oltre ai modelli LLM, hanno rilasciato anche modelli specializzati per task di coding e matematici, nonché MoE multimodali, che utilizzano la *Multi-head Latent Attention* (MLA)

https://huggingface.co/deepseek-ai

# Multi-head Latent Attention (MLA)

- La Multi-head Latent Attention (MLA) utilizza un’approssimazione a basso rango per proiettare le matrici KV ottenute con una strategia MHA
  - Lo spazio della KV cache è ridotto



# DeepSeek

| | # Params | Dettagli |
|---|---|---|
| DeepSeek LLM | 7B | Pretrained from scratch su dati cinesi e inglesi (2 trilioni di token). Architettura LLaMA-based con MHA |
| DeepSeek LLM | 67B | Pretrained from scratch su dati cinesi e inglesi (2 trilioni di token). Architettura LLaMA-based con GQA |
| DeepSeek LLM Chat | 7B, 67B | SFT, DPO |
| DeepSeek-R1-Zero | 671B (37B attivi) | MoE, RL, 128k contesto |
| DeepSeek-R1 | 671B (37B attivi) | MoE, SFT, RLHF, 128k contesto |
| DeepSeek V2 - lite | 16B (2.4B attivi) | MoE, Multi-head Latent Attention (MLA), 32k contesto |
| DeepSeek V2 | 236B (21B attivi) | MoE, Multi-head Latent Attention (MLA), 128k contesto |
| DeepSeek V3 | 671B (37B attivi) | MoE, Hybrid thinking mode, SFT, RL |

https://huggingface.co/deepseek-ai

# Qwen

- I modelli della famiglia Qwen sono stati sviluppati dalla compagnia cinese Alibaba Cloud.
- Modelli open-source decoder-only
- Comprende sia modelli text-to-text, Visual Language Models (VL), specializzati per la generazione di codice e calcolo matematico, omni e per la generazione di Embeddings

https://huggingface.co/Qwen/collections#collections

# Qwen

| | Relase | # Params | Dettagli |
|---|---|---|---|
| Qwen 1 | 2023 | 1B, 7B, 14B, 72B | Rilasciato in versione base e chat (RLHF) |
| Qwen 1.5 | 2024 | 0.5B, 1.8B, 4B, 7B, 14B, 32B, 72B, 110B | Rilasciato in versione base e chat (RLHF), versione MoE |
| Qwen 2 | 2024 | 0.5B, 1.5B, 7B, 72B | Rilasciato in versione base e instruct, GQA |
| Qwen 2.5 | 2024 | 0.5B, 1.5B, 3B, 7B, 14B, 32B, 72B | Rilasciato in versione base e instruct, versioni specializzate |
| Qwen 3 | 2025 | 0.6B, 1.7B, 4B, 8B, 14B, 32B, | Supporto multilingua, thinking mode |

https://huggingface.co/Qwen/collections#collections

# LLM in italiano

- LLaMantino-2 finetuned da LLaMA-2
- LLaMantino-2 chat finetuned da LLaMA-2 chat
- Fauno finetuned da LLaMA-2 chat
- Camoscio instruction-finetuned da LLaMA-2

- Anita finetuned da LLaMA-3 Instruct

- Minerva Mistral-based addestrato da zero

# Interagire con i modelli

- API a pagamento
  - API per interrogare il modello
  - Uso di servizi di host esterni

- Open-source
  - HuggingFace Transformers
  - Inferenze con vLLM e Ollama
  - Inferenze e applicazioni avanzate: LangChain e LlamaIndex
  - Addestramento: Unsloth

# Modelli open-source @ HuggingFace

- Gemma
  - https://huggingface.co/google/collections
- Llama
  - https://huggingface.co/meta-llama/collections
- Mistral
  - https://huggingface.co/mistralai/models
- Phi
  - https://huggingface.co/microsoft/collections
- DeepSeek
  - https://huggingface.co/deepseek-ai/collections
- Qwen
  - https://huggingface.co/Qwen/collections