# Transformer e Large Language Models 🤖

## La rivoluzione dell'NLP

# Agenda 📋

- Introduzione ai Transformer e LLM
- L'architettura Transformer in dettaglio
- Evoluzione dei Large Language Models
- Capacità e limitazioni
- Applicazioni pratiche
- Implementazione e considerazioni etiche

# Introduzione ai Transformer e LLM 🌟

- **Transformer**: Architettura rivoluzionaria introdotta nel 2017 ("Attention is All You Need")
- **Large Language Models (LLM)**: Modelli con miliardi di parametri addestrati su enormi corpora testuali
- Hanno trasformato radicalmente l'NLP e l'intero panorama dell'AI
- Capacità emergenti che vanno oltre l'addestramento esplicito

# Attention is all you need

- https://arxiv.org/html/1706.03762v7


alt text

# Il meccanismo di self-attention 🔍

- Ogni elemento interagisce direttamente con tutti gli altri
- Cattura dipendenze a lungo termine in modo efficiente
- Permette elaborazione parallela (vs. sequenziale nelle RNN)
- Un esempio da visualizzare: https://github.com/jessevig/bertviz

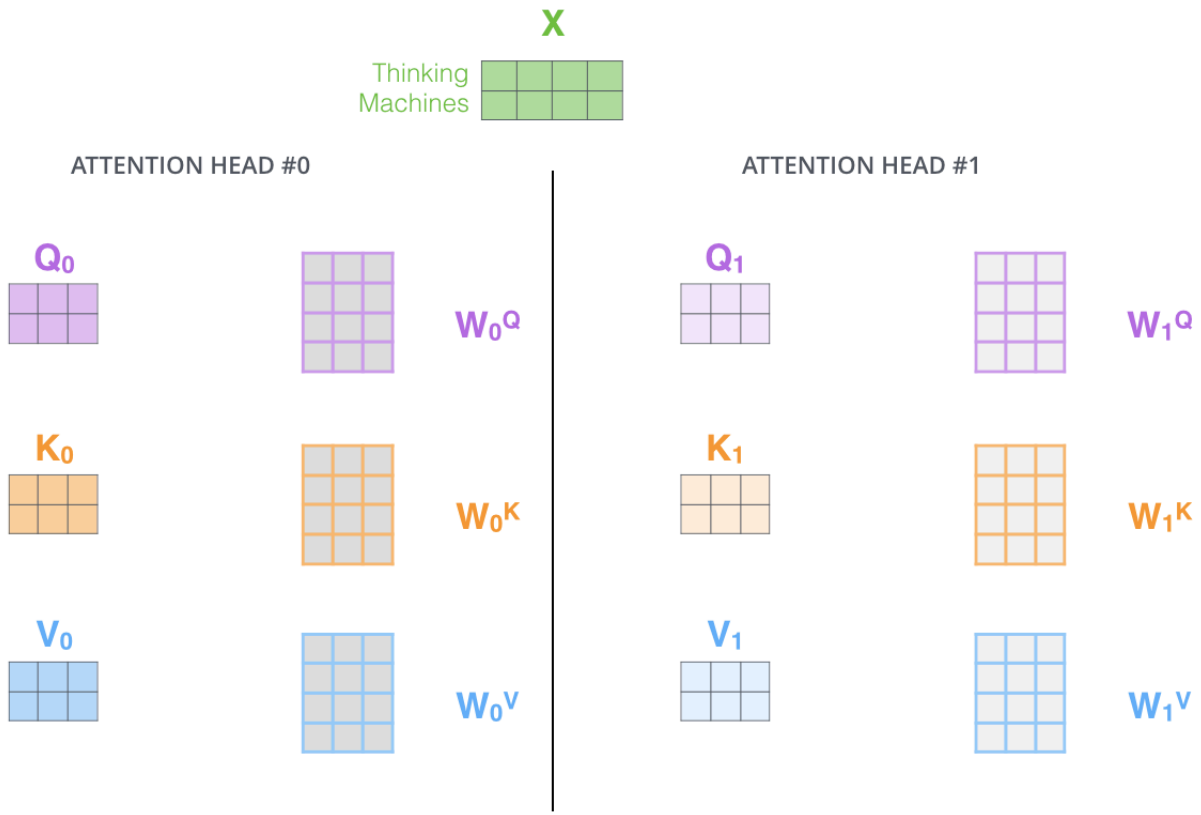
# Query, Key e Value 🗝️

Per ogni elemento della sequenza, vengono calcolati tre vettori:

- **Query (Q)**: Cosa l'elemento "sta cercando"
- **Key (K)**: Cosa l'elemento "offre" agli altri
- **Value (V)**: Il contenuto informativo dell'elemento

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

# Multi-Head Attention 🧠




- Multiple "teste" di attenzione in parallelo
- Ogni testa può specializzarsi in diversi tipi di relazioni

# Positional Encoding 📍

- I Transformer non hanno nozione intrinseca dell'ordine
- Si aggiungono encoding posizionali agli embedding

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

# Architettura completa del Transformer 🏗️

- **Encoder**: Elabora l'input in parallelo
- **Decoder**: Genera l'output autoregressivamente
- Layer Normalization e Residual Connections
- Feed-Forward Networks

# Da BERT a GPT: Paradigmi di pre-addestramento 🔄

- **BERT** (2018): Encoder-only, bidirezionale
  - Masked Language Modeling (MLM)
  - Next Sentence Prediction (NSP)
- **GPT** (2018-2023): Decoder-only, unidirezionale
  - Predizione della parola successiva
  - Scaling massiccio: GPT-1 (117M) → GPT-4 (trilioni?)

# BERT


alt text

https://github.com/google-research/bert

# Scaling laws e emergent abilities 📈

- **Scaling laws**: Relazioni prevedibili tra dimensioni, dati e performance
- **Emergent abilities**: Capacità che emergono improvvisamente oltre certe soglie
  - In-context learning
  - Chain-of-thought reasoning
  - Instruction following
  - Tool use

# Tecniche di addestramento avanzate 🛠️

- **Instruction tuning**: Addestramento a seguire istruzioni
- **RLHF** (Reinforcement Learning from Human Feedback):
  - Modello di ricompensa basato su preferenze umane
  - Ottimizzazione tramite reinforcement learning
- **Mixture of Experts** (MoE):
  - Attivazione selettiva di "esperti" specializzati
  - Efficienza computazionale

# Mixture of Experts

# Mixtral 8x7B


alt text fit

https://huggingface.co/blog/vtabbott/mixtral

# Modelli multimodali 🖼️🔊📝

- **Vision-Language Models**: Comprensione visiva integrata (GPT-4V, Gemini)
- **Generazione multimediale**: Immagini (DALL-E), audio, video
- **Modelli unificati**: Integrazione seamless di diverse modalità

# Capacità fondamentali dei LLM 💪

- **Comprensione contestuale**: Disambiguazione, coreference resolution
- **Generazione fluente e coerente**: Coerenza locale e globale, adattamento stilistico
- **In-context learning**: Few-shot learning, adattamento a nuovi compiti
- **Ragionamento e problem solving**: Chain-of-thought, decomposizione di problemi

# Limitazioni fondamentali 🚫

- **Allucinazioni**: Generazione di contenuti plausibili ma fattuali errati
- **Bias e stereotipi**: Perpetuazione di pregiudizi presenti nei dati
- **Limitazioni di contesto**: Finestra di contesto finita, degradazione dell'attenzione
- **Mancanza di comprensione profonda**: Comprensione simbolica limitata, grounding

# Applicazioni pratiche 🔧

- **Automazione documentale**: Generazione, analisi, estrazione, summarization
- **Assistenza alla scrittura**: Brainstorming, drafting, editing, traduzione
- **Assistenti virtuali**: Customer service avanzato, esperienze conversazionali
- **Supporto decisionale**: Analisi di dati testuali, sintesi di evidenze
- **Creatività aumentata**: Ideazione creativa, content creation scalabile

# Implementazione pratica 🖥️

- **Approcci di deployment**:
  - API commerciali vs modelli open-source
  - On-premise vs cloud vs edge
- **Ottimizzazione e personalizzazione**:
  - Fine-tuning e adattamento al dominio
  - Retrieval-Augmented Generation (RAG)
- **Integrazione nei flussi di lavoro**:
  - Connessione con sistemi esistenti
  - Human-in-the-loop approaches

# Conclusione 🎯

- I Transformer hanno rivoluzionato l'NLP con il meccanismo di self-attention
- I LLM hanno mostrato capacità emergenti sorprendenti con lo scaling
- Nonostante limitazioni come allucinazioni e bias, le applicazioni pratiche sono trasformative
- L'implementazione efficace richiede considerazioni tecniche, strategiche ed etiche
- Il futuro promette ulteriori progressi in allineamento, efficienza e multimodalità

# Grazie per l'attenzione! 🙏

## Domande?