Università di Perugia
Dipartimento di Matematica e Informatica

Corso di laurea in Informatica
Prova Finale

Large Language Models - Modelli Generativi Pre-Addestrati: Caratteristiche, Opportunità, Applicazioni

Professore
Prof. Alfredo Milani

Laureando
Federico Cervelli

Anno Accademico 2022-2023

Abstract

L'era digitale ha portato a un'esplosione di dati testuali provenienti da una vasta gamma di fonti, come social media, articoli online, documenti aziendali e molto altro. Questo enorme flusso di testo ha presentato sfide e opportunità nella comprensione e nell'elaborazione del linguaggio naturale. È in questo contesto che emergono i Large Language Models (LLM), modelli di apprendimento automatico capaci di analizzare e generare testi con una comprensione sempre più avanzata.

L'importanza dei LLM risiede nella loro capacità di affrontare complessità linguistiche e comunicative che vanno ben oltre la semplice elaborazione delle parole. Questi modelli, allenati su enormi quantità di testi, riescono a cogliere sfumature contestuali, riconoscere relazioni tra concetti e produrre testi coerenti e significativi. Questo rappresenta una pietra miliare nell'evoluzione dell'elaborazione del linguaggio naturale, aprendo la strada a una comunicazione più ricca e adatta alle esigenze umane e aziendali.

L'importanza dei LLM si riflette in vari settori. Nel campo della medicina, possono essere utilizzati per analizzare enormi dataset di record medici al fine di individuare modelli diagnostici. Nel settore legale, possono automatizzare la ricerca giuridica e la generazione di documenti. Nell'industria dell'istruzione, possono personalizzare l'apprendimento e l'interazione con gli studenti.

Tuttavia, con questa importanza emergono anche sfide etiche e tecniche. La scalabilità dei LLM richiede risorse computazionali significative, mentre la loro potenza può portare a risultati indesiderati, come l'incorporamento di bias nei dati o la diffusione di disinformazione. Pertanto, esplorare in modo completo e responsabile le potenzialità e le problematiche dei LLM diventa un passo cruciale per plasmare un futuro in cui l'elaborazione del linguaggio naturale possa essere sfruttata in modo etico ed efficace.

L'obiettivo di questa tesi è esaminare in dettaglio i Large Language Models, esplorandone la storia, le caratteristiche, le opportunità e le applicazioni, nonché le implicazioni etiche e sociali associate al loro utilizzo sempre più diffuso. Il presente lavoro mira a fornire una panoramica completa e critica dei LLM, con l'intento di contribuire a una comprensione più approfondita di come questi modelli stiano trasformando il campo dell'elaborazione del linguaggio naturale.

Indice

1 Storia dei Large Language Models (LLM) 4
1.1 Origini e precursori . . . 4
1.2 Inizio dell'era Transformer . . . 4
1.3 BERT e i primi Modelli Pre-Addestrati . . . 5
1.4 GPT-1 . . . 6
1.5 GPT-2 . . . 6
1.6 GPT-3 . . . 7
1.7 GPT-4 . . . 8

2 Fondamenti Teorici 9
2.1 Concetto di rete neurale . . . 9
2.2 Concetto di LLM . . . 10
2.3 Principi di addestramento . . . 11
2.4 Architetture comuni . . . 12
2.4.1 Reti Neurali Trasformer . . . 12
2.4.2 Reti Neurali Ricorrenti (RNN) . . . 12
2.4.3 Vantaggi e svantaggi . . . 12
2.5 Metriche di valutazione . . . 13

3 Caratteristiche dei LLM 16
3.1 Dimensione dei dati e potenza di calcolo richiesta . . . 16
3.2 Struttura dei livelli e dei parametri . . . 19
3.3 Capacità di generalizzazione ed overfitting . . . 23
3.4 Limiti e sfide nell'addestramento e nell'uso . . . 25

4 Applicazioni dei LLM 35
4.1 Analisi del sentiment dell'opinione pubblica . . . 35
4.2 Generazione di contenuti testuali . . . 35
4.3 Supporto negli ambienti di sviluppo . . . 36

5 Opportunità offerte dai LLM 37
5.1 Educazione Personalizzata . . . 37
5.2 Assistenza Sanitaria . . . 38
5.3 Integrazione con dispositivi elettronici . . . 39
5.4 Automazione nei contesti legali . . . 39

6 Conclusioni e Riferimenti 41
6.1 Conclusione . . . 41

1 Storia dei Large Language Models (LLM)

1.1 Origini e precursori

L'interesse per l'elaborazione automatica del linguaggio naturale (NLP) ha radici profonde nella storia dell'intelligenza artificiale e dell'informatica. Fin dagli anni '50 e '60, i ricercatori hanno esplorato come le macchine potessero comprendere e generare linguaggio umano. Inizialmente, tali tentativi erano basati su approcci simbolici e regole linguistiche, con risultati limitati.

Negli anni '80 e '90, le prime reti neurali hanno iniziato a essere utilizzate per compiti di NLP[1], segnando una svolta nell'approccio alla comprensione del linguaggio. Tuttavia, a causa delle limitazioni computazionali e della complessità dei modelli, questi sforzi non hanno raggiunto la potenza e la scalabilità dei moderni LLM.

Uno dei primi lavori rilevanti nella direzione dei LLM è stato il concetto di "word embeddings"[2] (rappresentazioni vettoriali di parole), che ha aperto la strada all'idea che il significato delle parole potesse essere catturato da vettori numerici, permettendo alle reti neurali di manipolarle più efficacemente.

Inoltre, nel 2003, Bengio et al.[3] hanno introdotto i modelli di linguaggio neurali probabilistici, dimostrando come le reti neurali potessero essere addestrate per generare testo in modo coerente e fluente. Questo lavoro è stato uno dei primi passi verso l'utilizzo delle reti neurali per creare testo, anche se le dimensioni e le capacità di questi modelli erano ancora molto limitate rispetto ai LLM attuali.

Parallelamente, nel campo del trattamento dell'informazione e del recupero dei documenti, gli sforzi per comprendere il significato semantico delle parole e dei testi hanno portato all'uso di algoritmi di "topic modeling" e altre tecniche per estrarre informazioni significative da grandi collezioni di testi.

Questi lavori pionieristici hanno gettato le basi concettuali e tecniche per l'evoluzione verso i Large Language Models. Tuttavia, è stato solo con l'avvento delle architetture Transformer e l'accumulo di enormi quantità di dati testuali che i LLM hanno raggiunto l'efficacia e la versatilità che li caratterizzano oggi.

1.2 Inizio dell'era Transformer

L'architettura Transformer, presentata nel 2017 da Vaswani et al. [4], ha segnato una rottura radicale con gli approcci precedenti all'elaborazione del linguaggio. Al cuore del Transformer c'è il concetto di "self-attention" (auto-attenzione), che ha permesso ai modelli di catturare relazioni a lungo raggio tra le parole all'interno di una frase o di un testo. Questo approccio contrastava con le reti neurali ricorrenti (RNN), che avevano difficoltà nel catturare dipendenze a lunga distanza.

L'uso della self-attention ha reso gli LLM basati su Transformer altamente paralleli e quindi facilmente scalabili, consentendo l'addestramento su enormi quantità di dati. Inoltre, il meccanismo di self-attention ha reso possibile l'elaborazione di sequenze di lunghezza variabile senza dover ridurre la dimensione della finestra di contesto, come avrebbe richiesto una RNN. Questo ha permesso l'elaborazione efficiente di documenti lunghi e discorsi complessi.

L'architettura Transformer ha anche introdotto l'idea di addestrare modelli prevedendo le parole in una sequenza da un contesto di parole circostante, sia a sinistra che a destra. Questo approccio bidirezionale ha permesso ai modelli di avere una migliore comprensione del contesto in cui si inseriscono le parole, migliorando la qualità delle rappresentazioni linguistiche apprese.

1.3 BERT e i primi Modelli Pre-Addestrati

Il rilascio di BERT (Bidirectional Encoder Representations from Transformers) da parte di Google nel 2018[5] è stato un punto di svolta nell'evoluzione dei LLM. BERT ha introdotto un nuovo paradigma nell'elaborazione del linguaggio naturale, aprendo la strada all'utilizzo di modelli pre-addestrati su grandi quantità di testo per migliorare le prestazioni su una vasta gamma di compiti di linguaggio.

Una delle innovazioni chiave introdotte da BERT è il concetto di pre-addestramento seguito da fine-tuning. Questo approccio si basa sull'idea di addestrare inizialmente il modello su un'enorme quantità di testo non etichettato, in modo che apprenda rappresentazioni linguistiche generali e contestuali. Questa fase di pre-addestramento consente al modello di sviluppare una comprensione profonda delle relazioni semantiche e sintattiche all'interno del testo.

Successivamente, il modello viene sottoposto a fine-tuning su un compito specifico, ad esempio il riconoscimento delle entità, la classificazione di sentiment, la traduzione automatica o qualsiasi altro compito di NLP. Durante il fine-tuning, il modello viene adattato a compiti specifici mediante l'uso di dati etichettati. Questo passaggio consente al modello di acquisire una conoscenza specializzata nel compito mirato, pur mantenendo le informazioni linguistiche generali apprese durante il pre-addestramento.

Il fine-tuning su compiti specifici ha permesso ai modelli di adattarsi in modo più mirato e preciso a compiti specifici, riducendo il bisogno di addestrare da zero un modello per ogni compito. Questo ha portato a risparmi di tempo e risorse significativi nello sviluppo di applicazioni di NLP.

L'introduzione di BERT ha ampliato l'impiego dell'elaborazione del linguaggio naturale in numerosi settori, dalla ricerca all'industria. I modelli pre-addestrati come BERT sono diventati la base di molte soluzioni NLP e hanno fornito risultati sor-

prendenti anche in compiti altamente complessi, come la comprensione del linguaggio naturale, la generazione di testo, la traduzione automatica e altro ancora.

1.4 GPT-1

Sebbene GPT-1 sia stato rilasciato prima di BERT, è giusto parlarne ora per evidenziare quanto sia stato importante per lo sviluppo dei modelli successivi GPT-2, GPT-3 e più recentemente, GPT-4.

Sviluppato nel giugno 2018 da OpenAI, un'*organizzazione senza fini di lucro di ricerca sull'intelligenza artificiale con lo scopo di promuovere e sviluppare un'intelligenza artificiale amichevole*[1] fondata nel dicembre 2015 a San Francisco da Elon Musk, Sam Altman e altri, Nonostante le sue dimensioni relativamente più piccole rispetto a modelli come BERT, GPT-1 ha dimostrato un'enorme promessa nella generazione di testo coerente e comprensibile. Con i suoi 117 milioni di parametri, GPT-1 ha sfidato le convenzioni e ha mostrato che la capacità di generare testo credibile e fluente può essere ottenuta con approcci generativi. Questo ha sollecitato gli scienziati a esplorare ulteriormente questa tecnologia, spingendo verso modelli con ancora più parametri.

1.5 GPT-2

GPT-2, la seconda iterazione dei modelli GPT sviluppati da OpenAI, ha segnato un notevole passo avanti rispetto a GPT-1. Rilasciato nel 2019, GPT-2 ha affrontato molte delle limitazioni del suo predecessore, dimostrando come l'aumento dei parametri abbia un ruolo cruciale nel miglioramento della coerenza e della precisione del modello.

GPT-2 è noto soprattutto per le sue dimensioni impressionanti: il modello era composto da circa 1,5 miliardi di parametri, un aumento significativo rispetto ai 117 milioni di GPT-1.

Quando è stato rilasciato, GPT-2 ha scatenato una notevole ondata di interesse mediatico. Questo perché il testo generato dal modello spesso risultava così convincente da essere difficile da distinguere da quello scritto da esseri umani. In un articolo di Alex Hern pubblicato su The Guardian[6], l'output di GPT-2 viene descritto come "plausibile prosa da giornale". Inoltre, si menziona come gli stessi sviluppatori di questa tecnologia abbiano espresso preoccupazione sul fatto che GPT-2 potesse essere troppo rischioso da rilasciare pubblicamente.

[1] `https://it.wikipedia.org/wiki/OpenAI`

1.6 GPT-3

GPT-3 (Generative Pre-trained Transformer 3) rappresenta una tappa epocale nell'evoluzione dei modelli GPT. Rilasciato nel 2020 da OpenAI, GPT-3 ha spinto oltre i confini stabiliti da GPT-2, dimostrando l'incredibile potenziale dei modelli di linguaggio basati su architetture Transformer.

Ciò che distingue GPT-3 in modo significativo è la sua enormità. Con ben 175 miliardi di parametri, GPT-3 è diventato uno dei modelli più grandi mai creati.

GPT-3 ha dimostrato di essere altamente versatile. Può generare testo in molteplici lingue e stili, affrontare una varietà di compiti di NLP, dal completamento automatico alla traduzione automatica, dalla generazione di codice alla risposta a domande complesse. La sua abilità di adattarsi a diversi contesti lo rende un modello estremamente flessibile e potente.

Tuttavia, con la sua grande potenza generativa, GPT-3 ha anche sollevato importanti questioni etiche. La possibilità di creare testo altamente persuasivo o fuorviante ha innescato discussioni sulla disinformazione e sull'uso responsabile della tecnologia. Queste preoccupazioni hanno sottolineato la necessità di una regolamentazione appropriata e di una comprensione consapevole dei limiti della generazione automatica di testo.

Inoltre, l'arrivo di GPT-3 ha suscitato preoccupazioni legate al futuro dell'occupazione e al possibile impatto sull'occupazione umana, specialmente nei settori legati all'elaborazione del linguaggio e alla generazione di contenuti.

Alcuni dei punti chiave che hanno alimentato le preoccupazioni includono:

- **Automazione di Compiti Manuali**: GPT-3 è in grado di generare testo in modo rapido ed efficiente, compiendo compiti come la scrittura di articoli, la composizione di e-mail, la produzione di contenuti per siti web e molto altro. Questa capacità ha sollevato la preoccupazione che potrebbe comportare la sostituzione di lavori manuali che coinvolgono la scrittura e la generazione di contenuti.

- **Impatti nell'Industria Editoriale**: Nel settore dell'editoria e del giornalismo, l'efficienza con cui GPT-3 può produrre articoli e notizie ha fatto sorgere timori sulla possibile riduzione del personale coinvolto nella scrittura e nella produzione di contenuti. Questo potrebbe avere conseguenze per la qualità e l'autenticità delle notizie generate.

- **Servizi di Scrittura Professionale**: Anche professionisti come scrittori freelance, copywriter e autori potrebbero vedere una maggiore concorrenza da parte di soluzioni automatizzate come GPT-3. La capacità di GPT-3 di produrre testi di alta qualità potrebbe portare ad una diminuzione delle opportunità di lavoro o alla riduzione dei prezzi.

1.7 GPT-4

GPT-4 è stato ufficialmente rilasciato il 14 marzo 2023 ed è ora accessibile attraverso l'utilizzo di chiamate API e per gli utenti di ChatGPT Plus. Seguendo il modello di altri "transformers", GPT-4 è stato pre-addestrato per prevedere il prossimo token utilizzando sia dati pubblici che "dati concessi in licenza da fornitori di terze parti". Successivamente, è stato ottimizzato attraverso l'apprendimento di rinforzo, grazie al feedback umano.

A differenza dei suoi predecessori, questo modello può comprendere istruzioni più sfumate e imprecise, ed è molto più creativo, inoltre accetta sia testo che immagini come input.

Nonostante il suo rilascio, ci sono ancora alcuni dettagli mancanti riguardo a GPT-4. Ad esempio, non sono state fornite informazioni ufficiali da OpenAI riguardo al numero esatto di parametri utilizzati nel modello. Anche dettagli sull'infrastruttura di calcolo e l'architettura sottostante impiegata per eseguire GPT-4 non sono stati ancora divulgati.

2 Fondamenti Teorici

2.1 Concetto di rete neurale

Le reti neurali artificiali sono un'importante classe di algoritmi utilizzati nell'intelligenza artificiale e nell'apprendimento automatico. Ispirate dal funzionamento del cervello umano, le reti neurali artificiali sono progettate per modellare e apprendere da dati complessi, consentendo alle macchine di svolgere compiti che richiedono comprensione, riconoscimento di pattern e decisioni.

le reti neurali sono strutture complesse di dati statistici organizzate per la modellazione di relazioni non lineari. Sono strumenti potenti che trovano applicazione nell'apprendimento automatico e nell'intelligenza artificiale. Esse sono in grado di simulare e comprendere relazioni intricate tra dati di input e output che spesso non possono essere adeguatamente rappresentate attraverso altre funzioni analitiche.

Un'analisi più dettagliata rivela che una rete neurale artificiale riceve segnali esterni attraverso uno strato di nodi, che sono le unità di elaborazione iniziali. Questi nodi di ingresso sono collegati a numerosi nodi interni distribuiti in diversi livelli. Ciascun nodo interno elabora i segnali in ingresso utilizzando pesi e bias associati ai collegamenti tra i nodi. Questi calcoli trasformano i segnali in ingresso in un'uscita che viene poi trasmessa ai nodi successivi attraverso ulteriori connessioni.

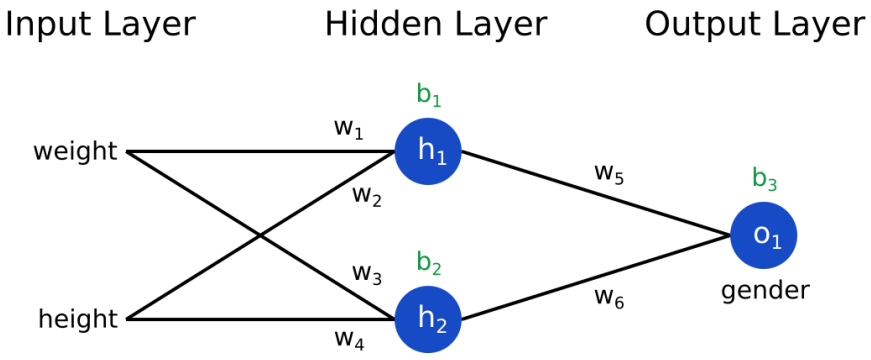


Figura 1: Esempio di Rete Neurale che "indovina" il sesso della persona a partire da altezza e peso.
Fonte: `https://victorzhou.com/blog/intro-to-neural-networks/`

Affronteremo in modo più dettagliato la struttura a livelli delle reti neurali quando parleremo delle sottoclassi specifiche di reti neurali utilizzate per creare i LLM. (3.2)

2.2 Concetto di LLM

I LLM rappresentano una classe avanzata di modelli di apprendimento automatico progettati per comprendere, generare e manipolare il linguaggio naturale umano. A differenza degli approcci tradizionali che si basavano su regole di linguaggio specifiche o su caratteristiche manualmente definite, i LLM sfruttano *reti neurali profonde*[2], una variante delle reti neurali caratterizzata da molti stadi intermedi, per apprendere direttamente dai dati testuali. Questi modelli sono noti per la loro capacità di affrontare la complessità e l'ambiguità del linguaggio, rendendoli uno dei più grandi progressi nell'elaborazione del linguaggio naturale.

Alla base dei LLM vi è l'idea di addestrare modelli su vaste quantità di testo in modo da acquisire una comprensione più ampia delle strutture linguistiche, delle relazioni semantiche e delle sfumature contestuali. Questi modelli apprendono in maniera supervisionata, dove vengono forniti grandi dataset di testo con etichette o obiettivi specifici, ad esempio predire la parola successiva in una frase o generare un testo coerente a partire da un input.

L'architettura tipica dei LLM si basa sulle reti neurali *trasformer*, che utilizzano meccanismi di attenzione e autoattenzione per catturare le dipendenze a lungo raggio all'interno del testo. Questi modelli suddividono il testo in *token*[3] (ad esempio, singole parole o sottoparti di parole) e li rappresentano attraverso *vettori di embedding*[4]. Gli strati di attenzione consentono al modello di focalizzare su parti specifiche del testo durante l'elaborazione, catturando le relazioni semantiche tra i token.

[2]https://en.wikipedia.org/wiki/Deep_learning#Deep_neural_networks
[3]https://en.wikipedia.org/wiki/Lexical_analysis#Tokenization
[4]https://en.wikipedia.org/wiki/Word_embedding

2.3 Principi di addestramento

Il processo di addestramento dei LLM è uno degli aspetti chiave che li rende così potenti nell'elaborazione del linguaggio naturale. Questo processo sfrutta il potere delle reti neurali profonde e richiede enormi quantità di dati testuali per acquisire una comprensione profonda del linguaggio umano. Qui, esamineremo in dettaglio i principi fondamentali che guidano l'addestramento dei LLM, in ordine temporale.

1. **Dati di addestramento:** L'addestramento dei LLM richiede enormi dataset di testo provenienti da una vasta gamma di fonti. Questi dati possono includere libri, articoli di notizie, pagine web, testi legali, conversazioni sociali e altro ancora. La diversità dei dati è cruciale per consentire al modello di apprendere la complessità e la varietà del linguaggio umano.

2. **Preprocessing e Tokenizzazione:** Prima dell'addestramento, i testi vengono sottoposti a un processo di preprocessing, che include la rimozione di caratteri speciali, la trasformazione in minuscolo e altre operazioni per rendere i dati uniformi. Successivamente, il testo viene suddiviso in token, come singole parole o parti di parole, in modo che il modello possa elaborarli in modo sequenziale.

3. **Addestramento supervisionato:** L'addestramento dei LLM è un processo supervisionato, dove il modello apprende fornendo esempi di input e gli obiettivi associati. Ad esempio, il modello può essere addestrato per prevedere la parola successiva in una frase. Questo richiede grandi quantità di dati etichettati, dove il testo è suddiviso in sequenze di token e ogni token è etichettato con il token successivo.

4. **Backpropagation e ottimizzazione:** Durante l'addestramento, il modello compie previsioni sui token successivi e calcola l'errore rispetto ai token veri. Questo errore viene propagato all'indietro attraverso la rete neurale utilizzando l'algoritmo di retropropagazione (backpropagation)[7]. Gli strati della rete vengono aggiornati iterativamente attraverso algoritmi di ottimizzazione come la discesa del gradiente, cercando di minimizzare l'errore di previsione.

5. **Fine-tuning e adattamento:** Dopo l'addestramento iniziale, i LLM possono essere ulteriormente adattati a compiti specifici attraverso il fine-tuning. In questa fase, il modello viene adattato a un dominio o a un compito specifico utilizzando dati etichettati relativi a quel compito. Questo consente ai LLM di adattarsi alle esigenze specifiche e di ottenere prestazioni migliori su compiti particolari.

2.4 Architetture comuni

Due delle architetture più comuni che hanno dimostrato di essere altamente efficaci sono le reti neurali trasformer e le reti neurali ricorrenti (RNN).

2.4.1 Reti Neurali Trasformer

Le reti neurali trasformer sono divenute l'architettura dominante per i LLM grazie alla loro capacità di catturare le relazioni a lungo raggio all'interno del testo. Questa architettura è basata su un meccanismo di attenzione chiamato "multi-head self-attention"[5]. In breve, il modello calcola l'attenzione tra ogni coppia di token nel testo, consentendo di catturare le dipendenze semantiche in modo contestuale.

L'architettura trasformer è costituita da vari strati chiamati "trasformatori", ognuno dei quali comprende sottocomponenti come l'attività di attenzione multi-head, la normalizzazione del batch e reti neurali feedforward. Gli strati trasformer operano in parallelo, consentendo al modello di affrontare l'ambiguità e la complessità del linguaggio attraverso diversi percorsi di analisi.

2.4.2 Reti Neurali Ricorrenti (RNN)

Le reti neurali ricorrenti sono un'altra architettura comune utilizzata nei LLM. Sono noti per la loro capacità di elaborare sequenze di dati, come testi. Le RNN operano sequenzialmente, elaborando un token alla volta e mantenendo una "memoria" interna che cattura le informazioni dai token precedenti.

Tuttavia, le RNN presentano limitazioni nella cattura delle dipendenze a lungo raggio e nell'elaborazione di sequenze lunghe. Per mitigare questo problema, sono state introdotte varianti delle RNN, come le Long Short-Term Memory (LSTM) e le Gated Recurrent Units (GRU), che permettono una migliore gestione delle informazioni a lungo termine.

2.4.3 Vantaggi e svantaggi

Le reti neurali trasformer offrono vantaggi significativi in termini di comprensione delle dipendenze a lungo raggio e gestione del contesto. Tuttavia, richiedono risorse computazionali significative a causa delle loro complessità, e il loro addestramento può richiedere molto tempo e dati. Le reti neurali ricorrenti, invece, sono meno complesse ma possono soffrire di problemi di vanishing gradient[8] e difficoltà nell'elaborazione di sequenze lunghe.

[5] `https://shorturl.at/dnCT4`

2.5 Metriche di valutazione

La valutazione dei LLM è essenziale per misurare le loro prestazioni e comprendere come si comportano in diverse attività di elaborazione del linguaggio naturale. Esistono diverse metriche di valutazione che consentono di quantificare l'efficacia dei LLM in vari compiti linguistici.
Qui esploreremo alcune delle metriche di valutazione più comuni e importanti.

- **Perplessità (Perplexity)**: Misura quanto un modello sia "confuso" o "perplesso" di fronte a nuovi dati o sequenze di parole.

  Essa viene calcolata utilizzando una misura basata sulla probabilità. Più specificamente, dato un testo di riferimento (o un insieme di dati di test), il modello calcola la probabilità di generare effettivamente quella sequenza di parole.

  Una perplessità bassa indica che il modello è in grado di generare testo che si avvicina molto al testo di riferimento, con una distribuzione di probabilità ben calibrata sulle possibili sequenze di parole. Ciò suggerisce che il modello ha una buona comprensione della struttura del linguaggio e delle relazioni tra le parole.

  D'altra parte, una perplessità alta indica che il modello ha difficoltà a predire le sequenze di parole corrette e può produrre testo meno coerente o meno preciso.

- **Valutazione Umanistica**: La valutazione umanistica rappresenta un approccio cruciale per valutare la qualità e la natura del testo generato dai LLM. Coinvolge esseri umani, spesso annotatori, che assegnano punteggi e giudizi ai testi prodotti dal modello. L'obiettivo è valutare una serie di fattori che includono coerenza, pertinenza, grammatica, comprensibilità e altro ancora.

  Tuttavia, ci sono alcune sfide da affrontare quando si utilizza questa metrica. La percezione della qualità del testo può variare da individuo a individuo, introducendo un elemento di soggettività nell'analisi. Per garantire risultati affidabili, è essenziale che gli annotatori siano ben addestrati e che vi sia un accordo tra di loro su come valutare i testi. La progettazione accurata degli esperimenti e la definizione chiara dei criteri di valutazione sono fondamentali per ridurre la variabilità tra gli annotatori.

  Gli studiosi e gli sviluppatori spesso seguono protocolli rigorosi per la valutazione umanistica. Questi possono includere l'addestramento e la calibrazione degli annotatori, l'uso di campioni di testi di riferimento e l'analisi statistica per misurare l'affidabilità delle valutazioni umane.

  La valutazione umanistica, se gestita con attenzione, può fornire una visione più completa delle prestazioni di un modello nei contesti reali e guidare ulteriori miglioramenti.

- **F1-score**: Questa metrica riveste un ruolo rilevante nella valutazione dei LLM quando si affrontano compiti di classificazione, come l'analisi del sentiment o l'identificazione delle entità.

  L'F1-score è una metrica che considera sia la *precisione* (il rapporto tra i veri positivi e la somma tra veri positivi e falsi positivi) che il *richiamo* (il rapporto tra i veri positivi e la somma tra veri positivi e falsi negativi). Essenzialmente, l'F1-score rappresenta l'equilibrio tra la capacità del modello di identificare correttamente le istanze positive (richiamo) e la capacità di evitare falsi positivi (precisione). Una sua valutazione elevata indica che il modello è in grado di bilanciare bene tra queste due metriche, mentre un valore basso potrebbe indicare un disequilibrio tra precisione e richiamo.

- **BLEU (Bilingual Evaluation Understudy)**: Il BLEU è una metrica fondamentale quando si tratta di valutare la qualità delle traduzioni prodotte dai LLM. Questa metrica è progettata per misurare quanto bene le frasi generate dal modello si avvicinano alle frasi di riferimento tradotte da esseri umani.

  La natura del BLEU si basa su un confronto statistico tra le frasi generate e le frasi di riferimento. Vengono conteggiate le parole o i n-grammi presenti in entrambe le frasi, e l'indice BLEU viene calcolato come una media geometrica delle precisioni di n-grammi. Questo punteggio fornisce una stima di quanto il testo generato sia simile alle frasi umane di riferimento.

  Tuttavia, il BLEU potrebbe presentare alcune limitazioni nel catturare la vera qualità delle traduzioni. Per esempio, potrebbe non considerare la fluidità e la coerenza del testo generato, focalizzandosi principalmente sulla sovrapposizione delle parole con le frasi di riferimento. Le traduzioni fluide e naturali potrebbero non ricevere un punteggio alto se non corrispondono esattamente alle frasi di riferimento, limitando così la capacità del BLEU di riconoscere la creatività del modello nel generare testo.

  Inoltre, il BLEU potrebbe avere difficoltà nel gestire la complessità semantica e sintattica delle traduzioni umane, poiché si basa su un confronto di parole o n-grammi. Le traduzioni che presentano variazioni creative o strutturali potrebbero non ottenere un punteggio adeguato, nonostante la loro validità come traduzioni a tutti gli effetti.

- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)**: Il ROUGE rappresenta un insieme di metriche ampiamente impiegate per valutare la qualità della generazione di testi, in particolare per compiti come la creazione di riassunti o sintesi. La sua funzione principale è misurare la sovrapposizione

dei n-grammi, ovvero le sequenze di n parole consecutive, tra i testi generati e i testi di riferimento umani.

Questo insieme di metriche è orientato verso il richiamo (recall), concentrandosi sulla capacità del modello di generare sequenze di parole che si sovrappongono alle sequenze di riferimento. Misurando la sovrapposizione degli n-grammi, il ROUGE cattura l'efficacia del modello nell'incorporare informazioni rilevanti presenti nei testi di riferimento.

Nel contesto delle applicazioni di generazione di testi, come riassunti o sintesi, il ROUGE fornisce una valutazione utile della coerenza tra il testo generato e il contenuto dei testi di riferimento. Tuttavia, è importante riconoscere che il ROUGE, come il BLEU, potrebbe non cogliere appieno la qualità semantica, la creatività e l'originalità del testo generato. Sequenze di parole simili non sempre riflettono completamente l'essenza e la struttura del testo di riferimento.

3 Caratteristiche dei LLM

3.1 Dimensione dei dati e potenza di calcolo richiesta

La dimensione dei dati e la potenza di calcolo richiesta emergono come fattori fondamentali che influenzano l'addestramento, le prestazioni e l'efficacia dei LLM. La portata e la complessità delle operazioni che li coinvolgono hanno un impatto significativo su come questi modelli apprendono, generalizzano e rispondono alle richieste dei compiti di linguaggio naturale.

L'enorme dimensione dei dati utilizzata per addestrare i LLM è una delle caratteristiche distintive di questi modelli. I modelli di lingua di grandi dimensioni vengono addestrati su collezioni di testi vastissime, spesso comprendenti miliardi di parole o addirittura più dati testuali. Questo approccio di addestramento su grandi quantità di testo permette ai modelli di apprendere le sfumature del linguaggio umano, la sintassi, il contesto e le relazioni semantiche.

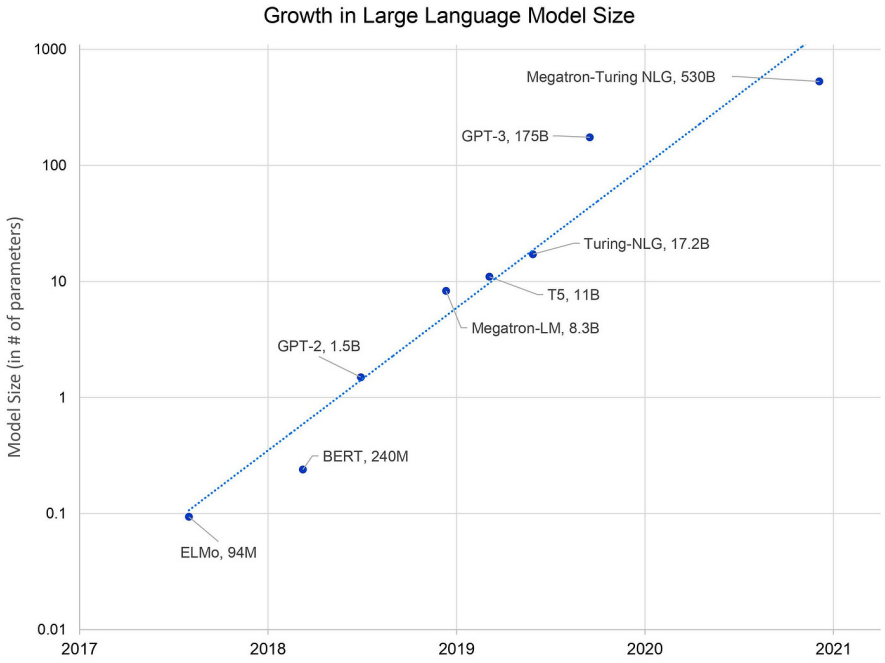


Figura 2: Crescita del numero di parametri nel tempo nei LLM
Fonte: `https://golden.com/wiki/Large_language_model-BYEPDJK`

Parallelamente, la potenza di calcolo richiesta per addestrare e utilizzare LLM è

significativa. L'addestramento di modelli complessi richiede l'uso di infrastrutture di calcolo avanzate, come cluster di GPU o TPU (Tensor Processing Unit). La potenza di calcolo necessaria è proporzionale alla dimensione del modello, e modelli più grandi possono richiedere un investimento considerevole in termini di risorse computazionali.

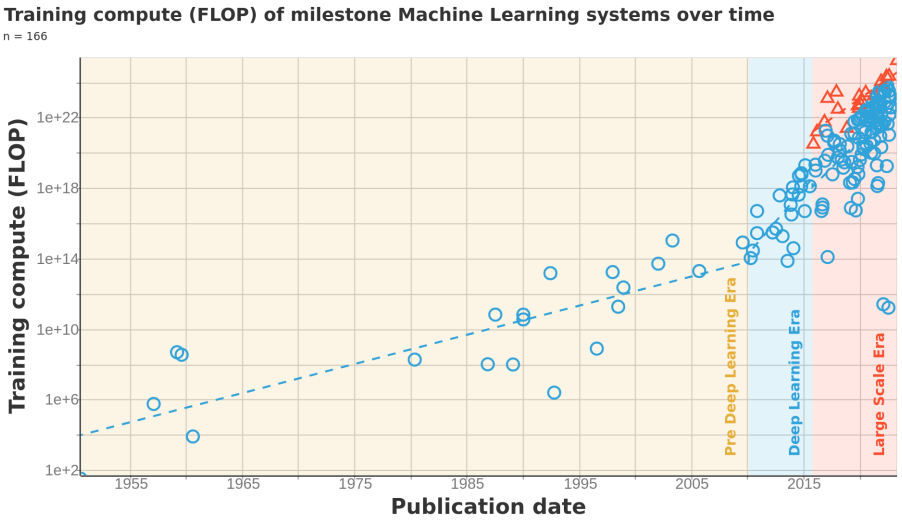


Figura 3: Crescita dei *FLOPs* [6]necessari per il training dei LLM nel tempo[9]

[6]I FLOPS (Floating Point Operations Per Second) rappresentano la misura della velocità di calcolo di un processore, indicando quanti calcoli in virgola mobile possono essere eseguiti in un secondo.

È importante notare che la dimensione dei dati e la potenza di calcolo richiesta sono fattori interconnessi. La grande quantità di dati richiede potenza di calcolo per essere elaborata efficacemente, e allo stesso tempo, la potenza di calcolo avanzata consente di addestrare modelli più grandi e complessi su dataset estesi.

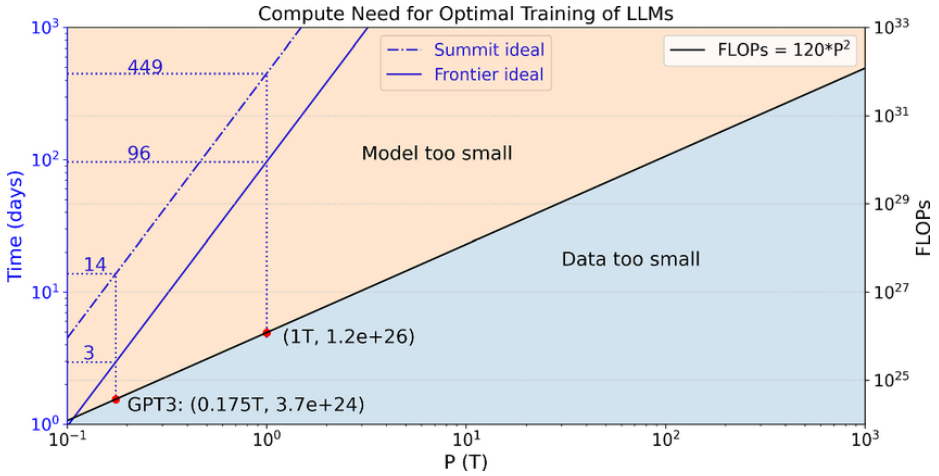


Figura 4: Correlazione tra parametri e FLOPs necessari per l'addestramento ottimale di un LLM [10]

Tuttavia, queste caratteristiche non sono prive di sfide. L'uso di enormi quantità di dati può sollevare preoccupazioni riguardo alla privacy, alla provenienza dei dati e alla potenziale presenza di bias. Allo stesso modo, la richiesta di potenza di calcolo può limitare l'accesso e l'utilizzo dei LLM a organizzazioni o istituzioni con risorse computazionali significative.

3.2 Struttura dei livelli e dei parametri

I Large Language Models sono costruiti utilizzando reti neurali profonde, che sono composte da diversi livelli o strati interconnessi. Ogni livello svolge una specifica operazione di trasformazione dei dati in ingresso, e l'output di un livello diventa l'input del successivo.

Gli strati principali nelle architetture dei LLM includono:

- **Embedding Layer**: Questo è il primo strato dell'architettura. Esso svolge un ruolo cruciale nella trasformazione delle parole o dei token, che rappresentano le unità base di un testo, in rappresentazioni numeriche dense chiamate "embedding". Di seguito come funziona e perchè è cruciale:
  - **Rappresentazione numerica dei testi**: Le reti neurali e i modelli di machine learning in genere lavorano con dati numerici. Tuttavia, il linguaggio naturale è composto da parole e frasi. L'Embedding Layer è responsabile di convertire le parole o i token in vettori di numeri reali. Questi vettori possono catturare il significato semantico delle parole e le relazioni tra di esse.
  - **Apprendimento delle relazioni semantiche**: Gli embedding sono progettati in modo che parole con significati simili abbiano rappresentazioni numeriche simili. Ciò significa che le parole che sono semanticamente correlate saranno più vicine tra loro nello spazio degli embedding. Ad esempio, le parole "gatto" e "cane" avranno embedding simili poiché condividono un contesto simile nel linguaggio.
  - **Riduzione della dimensionalità**: L'Embedding Layer consente di ridurre la dimensione dello spazio delle parole. Invece di trattare ogni parola come un vettore one-hot (dove solo un elemento è 1 e tutti gli altri sono 0), che potrebbe portare a rappresentazioni molto sparse e poco informative, gli embedding riducono la dimensione del vettore e catturano comunque le caratteristiche semantiche importanti.
  - **Generalizzazione delle parole rare**: Le parole meno frequenti possono beneficiare dell'Embedding Layer in quanto possono acquisire informazioni dal contesto circostante. Questo aiuta a prevenire l'overfitting (l'adattamento eccessivo ai dati di allenamento) poiché le parole rare avranno ancora rappresentazioni significative.
- **Transformer Layers**: I Transformer Layers costituiscono il cuore dell'architettura dei moderni LLM, come ad esempio il GPT. Questi strati utilizzano meccanismi di attenzione per catturare relazioni a lungo raggio tra le parole

all'interno di un testo. Ciò consente ai modelli di cogliere le dipendenze e le connessioni semantiche complesse in un testo, migliorando così la loro capacità di generare testo coerente e significativo.

Ogni Transformer Layer è composto da due sottostrati principali:

- **Multi-Head Self Attention**: Questo strato è una delle componenti chiave dei Transformer Layers. Per catturare le relazioni tra le parole, calcola le attenzioni tra ogni parola/token nel testo. L'attenzione qui è "self-attention", il che significa che una parola può attribuire diversi pesi ad altre parole nel testo sulla base delle relazioni semantiche tra di loro. L'uso di "multi-head" significa che l'attenzione viene calcolata in parallelo su diverse proiezioni delle parole, consentendo al modello di catturare relazioni diverse e complesse.
- **Feedforward Neural Network**: Dopo che l'output è stato calcolato attraverso il sottostrato di attenzione, passa attraverso il Feedforward Neural Network, un componente chiave del Transformer Layer. Questo strato svolge il ruolo di arricchire ulteriormente le rappresentazioni dei token, consentendo al modello di catturare relazioni semantiche più complesse e rappresentazioni avanzate.

  L'output dall'attenzione multi-head è costituito da rappresentazioni dei token che incorporano informazioni sul contesto circostante e le relazioni semantiche. Queste rappresentazioni passano attraverso una serie di trasformazioni lineari. In pratica, per ogni token, vengono applicate trasformazioni che coinvolgono pesi e bias. Successivamente, viene applicata una funzione di attivazione non lineare, spesso ReLU (Rectified Linear Unit)[7].

  Questo processo di trasformazione lineare e attivazione non lineare consente al modello di catturare relazioni più intricate tra le caratteristiche di input. Il Feedforward Neural Network è in grado di individuare pattern e relazioni non lineari nei dati, andando oltre le semplici correlazioni. Ciò consente al modello di apprendere rappresentazioni più profonde e ricche, che possono rappresentare sfumature linguistiche complesse e strutture semantiche avanzate.

- **Output Layer**: L'Output Layer rappresenta l'ultimo strato all'interno di un modello e ha il compito di generare le previsioni o le risposte finali del modello a partire dalle rappresentazioni apprese durante il processo di elaborazione dei dati.

[7] https://it.wikipedia.org/wiki/Rettificatore_(reti_neurali)

Le sue funzionalità possono variare a seconda del tipo di compito che il modello deve affrontare. Alcune delle sue possibili implementazioni sono:

- **Layer Softmax per Classificazione**: Se il compito consiste nella classificazione di più categorie, come ad esempio nella classificazione di testi in diverse categorie, l'Output Layer potrebbe essere un layer softmax. Questo tipo di layer calcola le probabilità relative per ciascuna classe di appartenenza, normalizzando le uscite precedenti e rendendole interpretabili come distribuzioni di probabilità. La classe con la probabilità più alta verrà considerata la previsione del modello.
- **Layer Lineare per Regressione**: Se il compito è di tipo regressione, dove l'obiettivo è prevedere un valore numerico continuo, l'Output Layer potrebbe essere un layer lineare. Questo strato applica una trasformazione lineare alle rappresentazioni apprese precedentemente, producendo una stima numerica continua come previsione del modello.
- **Layer Generativo per LLM**: Nel caso dei Large Language Models come GPT, l'Output Layer è anche responsabile della generazione di testo. In questo contesto, l'output layer utilizza le rappresentazioni apprese per predire la parola successiva nella sequenza, in modo da generare un flusso coerente di testo.
- **Layer di Decodifica per Sequenze**: In compiti di generazione di sequenze, come la traduzione automatica, l'Output Layer potrebbe essere uno strato di decodifica che, in combinazione con l'attenzione, genera sequenze di output di lunghezza variabile.

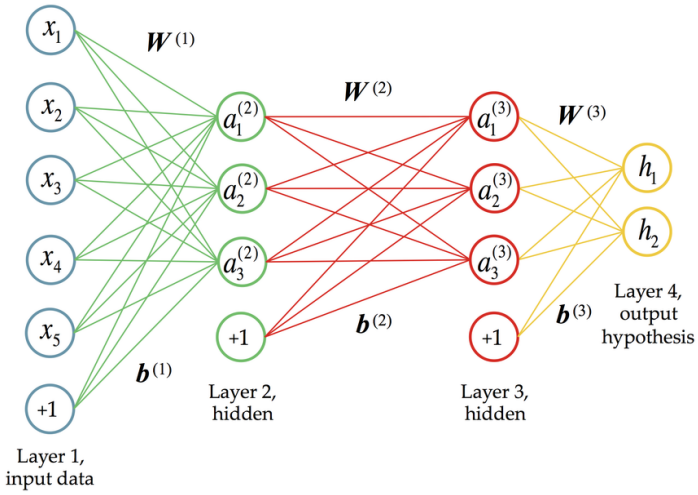


Figura 5: Esempio della struttura a livelli.[11]

Oltre ai livelli, i LLM hanno un gran numero di parametri, che sono i pesi delle connessioni tra i neuroni in ogni livello. Questi parametri sono adattati durante l'addestramento del modello per minimizzare la differenza tra le previsioni del modello e i risultati attesi.

Questa struttura a livelli e parametri è altamente adattabile e può essere regolata per affrontare diversi compiti.

3.3 Capacità di generalizzazione ed overfitting

- La capacità di **generalizzazione** è un concetto fondamentale nel campo dell'apprendimento automatico. Rappresenta la capacità di un modello di apprendere da un insieme di dati di addestramento e di applicare ciò che ha imparato a nuove situazioni o dati che non ha mai incontrato prima. In altre parole, un modello con una buona capacità di generalizzazione è in grado di estrarre regolarità e pattern dai dati di addestramento e di applicarli in modo coerente ed efficace a casi mai visti prima.

  Una buona capacità di generalizzazione significa che il modello è in grado di comprendere e catturare le complessità del linguaggio umano, tra cui le relazioni semantiche, la struttura sintattica e il significato implicito. Quando il modello ha appreso tali sfumature durante l'addestramento su grandi quantità di testo, può applicare queste conoscenze per generare risultati coerenti e significativi anche su input diversi da quelli utilizzati per l'addestramento.

  Una capacità di generalizzazione solida è cruciale perché, nella pratica, non possiamo mai esporre un modello a ogni possibile variante di dati che incontrerà. Un modello che generalizza bene è in grado di riutilizzare le sue conoscenze acquisite per affrontare nuove situazioni, rendendo la sua utilità molto più ampia.

- L'**overfitting** è una situazione problematica nell'apprendimento automatico in cui un modello si adatta eccessivamente ai dati di addestramento. Questo può accadere quando il modello è addestrato per catturare ogni dettaglio dei dati, compreso il rumore o le eccezioni presenti. Di conseguenza, il modello potrebbe perdere la sua capacità di generalizzare.

  Un modello che soffre di overfitting può diventare "troppo specializzato" per i dati di addestramento e avere difficoltà a gestire input diversi o nuovi. Mentre il modello potrebbe essere in grado di fornire previsioni altamente accurate sui dati con cui è stato addestrato, le sue previsioni su dati nuovi o variabili potrebbero essere inaffidabili. In altre parole, l'overfitting può portare a una perdita della capacità di fare previsioni coerenti e significative su input diversi.

  Per affrontare l'overfitting, gli scienziati dei dati adottano varie strategie di regolarizzazione e validazione. Queste tattiche mirano a impedire al modello di adattarsi troppo strettamente ai dati di addestramento e a promuovere una migliore capacità di generalizzazione. L'obiettivo è trovare un equilibrio tra l'adattamento ai dati di addestramento e la capacità di applicare ciò che è stato appreso a nuovi scenari.

Per i LLM, la sfida sta nel trovare un equilibrio tra l'apprendimento delle caratteristiche linguistiche universali che consentono la generalizzazione e l'evitare di imparare dettagli superflui che portano all'overfitting. La complessità dei LLM, combinata con l'ampio insieme di dati di addestramento, contribuisce alla loro capacità di generalizzazione su varie forme di testo e argomenti.

Tuttavia, la cura deve essere posta per evitare di introdurre nel modello bias o errori presenti nei dati di addestramento. Questi problemi potrebbero compromettere la sua capacità di generalizzazione e portare a risultati distorti. Monitorare attentamente il processo di addestramento, selezionare dati di alta qualità e applicare tecniche di regolarizzazione sono modi per affrontare questi problemi.

3.4 Limiti e sfide nell'addestramento e nell'uso

L'addestramento e l'uso dei LLM presentano una serie di sfide e limiti che devono essere affrontati:

1. **Requisiti di Dati e Potenza di Calcolo**: L'addestramento dei Large Language Models richiede enormi quantità di dati testuali per garantire che il modello apprenda in modo accurato le complessità e le sfumature del linguaggio umano. Inoltre, la potenza di calcolo richiesta è altrettanto considerevole per processare e ottimizzare i molti parametri e le operazioni di apprendimento del modello. Questi requisiti combinati possono tradursi in costi significativi sia in termini di raccolta dei dati che di risorse di calcolo.

   Nella pratica, i LLM come GPT sono addestrati su corpora di testi molto ampi, spesso che comprendono miliardi o addirittura trilioni di parole provenienti da una vasta gamma di fonti. Questi dati rappresentano il bagaglio informativo su cui il modello costruisce le sue conoscenze linguistiche. Per mantenere un'ampia copertura e diversità di linguaggio, è necessario raccogliere dati da una varietà di fonti, come libri, articoli di notizie, pagine web e altro ancora.

   Tuttavia, la raccolta, la pulizia e la preparazione di questi dati richiedono tempo e risorse. Inoltre, i costi di calcolo per addestrare i modelli sono elevati a causa del loro grande numero di parametri e delle operazioni di apprendimento intensive. La formazione richiede l'uso di hardware specializzato, come unità di elaborazione grafica (GPU) ad alte prestazioni o unità di elaborazione tensoriale (TPU), che possono essere costosi da acquistare e mantenere.

   Un esempio che illustra il costo significativo associato all'addestramento di LLM è quello di PaLM (Path-Aware Large Language Model), un modello di linguaggio di Google con 540 miliardi di parametri. Si stima che l'addestramento di PaLM abbia comportato costi compresi tra 9 e 23 milioni di dollari[8]. Questi costi coprono l'infrastruttura di calcolo, l'energia elettrica e altri fattori.

2. **Bias e Discriminazione**: I dati di addestramento possono contenere bias culturali, di genere, razziali e altri, che vengono appresi dal modello. Ciò può portare a output discriminatori o ingiusti, rappresentando un problema etico.

   Ad esempio, il caso del "recruiting tool" di Amazon[9] evidenzia chiaramente come i bias nei dati di addestramento possano avere conseguenze significative. Amazon ha sviluppato un sistema di intelligenza artificiale per assistere nella selezione dei candidati per le posizioni di lavoro. Tuttavia, il modello è stato

[8] `https://blog.heim.xyz/palm-training-cost/`

[9] `https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G`

affetto da un evidente bias di genere a causa dei dati di addestramento che riflettevano le disuguaglianze di genere esistenti nell'industria tecnologica. Invece di aiutare a eliminare i pregiudizi, il sistema ha inavvertitamente rafforzato gli stereotipi di genere. Questo ha portato il modello a scremare automaticamente i curriculum che contenevano termini legati al genere femminile o a università prevalentemente femminili, risultando in una discriminazione involontaria nei confronti delle candidate.

3. **Controllo Creativo**: I Large Language Models, come ad esempio il GPT, sono noti per la loro abilità di generare testo creativo e coerente. Tuttavia, il livello di controllo sul contenuto di ciò che generano può essere limitato. Questo può causare risultati inaspettati o indesiderati, soprattutto in contesti sensibili.

   Questo problema può derivare da diverse ragioni. Ad esempio, l'interpretazione delle istruzioni fornite per guidare la generazione di testo potrebbe variare tra il modello e l'utente, portando a risultati ambigui. Inoltre, i modelli possono assorbire bias e contenuti sensibili dai dati di addestramento, che potrebbero emergere nel testo generato.

   Il grado di controllo sulla generazione può variare a seconda delle tecniche utilizzate per guidare il modello. Alcuni modelli potrebbero generare risposte che non rispettano completamente l'intenzione dell'utente, creando risultati non desiderati. Inoltre, l'interpretazione del contesto di generazione potrebbe non essere perfettamente accurata.

   Gli sforzi sono in corso per affrontare queste sfide e migliorare il controllo creativo nei LLM. Tuttavia, trovare soluzioni che funzionino in tutte le situazioni è ancora un'area di ricerca in evoluzione.

4. **Comprensione del Contesto**: I LLM possono avere difficoltà a comprendere il contesto corretto per rispondere in modo accurato. Ciò può portare a risposte fuorvianti o incomprensibili.

5. **Interpretabilità**: La complessità dei LLM, derivante dalla loro profonda struttura e dall'ampio numero di parametri, può rendere opaco il collegamento tra l'input dato e l'output prodotto. Questo può comportare una mancanza di chiarezza su come vengono generate le risposte o prese le decisioni. Non è raro che questi modelli vengano spesso definiti delle "scatole nere".

   L'assenza di interpretabilità può portare a diverse sfide. Ad esempio, il modello potrebbe non essere in grado di spiegare in modo dettagliato come arriva a una certa risposta, rendendo difficile la comprensione del ragionamento sottostante. Questo può anche complicare l'individuazione delle fonti di errori o insoddisfazioni nelle risposte prodotte dal modello.

Quando il modello genera testo creativo o risposte autonome, può essere difficile comprendere le ragioni specifiche che hanno portato a determinate scelte linguistiche. Questo può sollevare domande sulla provenienza e la coerenza di tali risposte.

Un aspetto cruciale dell'interpretabilità è la fiducia nell'uso dei modelli e la loro accountability. Senza la capacità di comprendere chiaramente come il modello giunge alle sue previsioni, è difficile stabilire quanto ci si possa affidare alle sue risposte e quanto ci si possa basare su di esso.

6. **Ottimizzazione dei Parametri**: L'ottimizzazione dei parametri nei LLM, è un processo cruciale ma complesso. Questi modelli sono caratterizzati da un gran numero di parametri che influenzano il modo in cui il modello apprende e genera il testo. Trovare i parametri ottimali per massimizzare le prestazioni richiede un equilibrio tra diversi fattori, come l'adattamento ai dati di addestramento e la capacità di generalizzazione a nuovi dati.

   Data la complessità dei LLM, non è sempre chiaro come i cambiamenti nei parametri influenzino le prestazioni complessive del modello. Ciò richiede un approccio empirico, in cui vengono testati diversi set di parametri attraverso iterazioni di *trial and error*. Durante questo processo, è spesso necessario trovare un compromesso tra obiettivi contrastanti, come migliorare le prestazioni sui dati di addestramento senza compromettere la capacità del modello di generalizzare.

   Ci sono diversi parametri che possono essere ottimizzati al fine di migliorare le prestazioni del modello. Alcuni di questi parametri includono:

   - **Tassi di Apprendimento**: Questi parametri regolano quanto velocemente il modello aggiorna i suoi pesi in risposta ai dati di addestramento. Trovare il tasso di apprendimento ottimale è cruciale per bilanciare la velocità di apprendimento e la stabilità del processo di ottimizzazione.
   - **Batch Size**: Il batch size rappresenta il numero di esempi di addestramento utilizzati in ciascun passaggio di ottimizzazione durante l'addestramento del modello.

     La scelta del batch size è una decisione importante durante l'addestramento di un modello e ha un impatto su vari aspetti del processo di apprendimento:

     Stabilità del Processo di Apprendimento: Un batch size adeguato può contribuire a rendere il processo di apprendimento più stabile. L'utilizzo di un batch size troppo piccolo può introdurre una maggiore variabilità nei gradienti calcolati durante l'ottimizzazione, il che potrebbe

rendere l'addestramento più "rumoroso" e meno coerente. D'altra parte, un batch size troppo grande può richiedere risorse computazionali e di memoria considerevoli, rendendo difficile l'addestramento su hardware limitato.

Velocità di Convergenza: La scelta del batch size può influenzare la velocità con cui il modello raggiunge la convergenza. In generale, un batch size più grande può accelerare la convergenza, in quanto viene calcolato un aggiornamento di gradiente basato su più esempi di addestramento in ogni passaggio.

Consumo di Memoria: Un batch size più grande richiede più memoria per archiviare i gradienti intermedi e le attivazioni dei layer durante l'addestramento. Questo può diventare un fattore limitante quando si lavora su hardware con risorse limitate, come le GPU o le TPU.

La scelta del batch size dipende da vari fattori, tra cui la dimensione del dataset di addestramento, le risorse computazionali disponibili e la complessità del modello. In genere, è una buona pratica sperimentare con diverse dimensioni di batch durante l'addestramento per trovare quella che offre il miglior equilibrio tra stabilità, velocità di convergenza ed efficienza delle risorse.

Inoltre, esistono approcci come il "mini-batch gradient descent" che consentono di sfruttare i vantaggi di diverse dimensioni di batch durante l'addestramento, utilizzando un campione casuale di esempi di addestramento in ciascun passaggio. Questo permette una maggiore flessibilità nella scelta del batch size.

- **Dimensioni dei Vettori di Embedding**: Le dimensioni delle rappresentazioni embedding delle parole si riferiscono alla dimensione dei vettori che rappresentano ciascuna parola o token nel vocabolario del modello. Questi vettori embedding fungono da spazi latenti in cui il modello apprende a rappresentare il significato delle parole. Aumentare le dimensioni degli embedding può consentire al modello di catturare sfumature più ricche e complesse nel significato delle parole. Tuttavia, ciò comporta un aumento nella dimensione totale del modello e nella memoria richiesta per archiviare i parametri degli embedding
- **Numero di Strati e di Teste di Attivazione**: La profondità degli strati e il numero di teste di attivazione sono parametri critici nell'architettura dei Large Language Models basati su transformer e possono influenzare notevolmente la capacità del modello di apprendere relazioni a lungo raggio nel testo.

Numero di Strati (Layers): Aumentare il numero di strati in un modello transformer può aumentare la sua capacità di apprendimento, consentendo di catturare relazioni complesse e a lungo raggio nei dati di addestramento. Tuttavia, questo aumento della profondità richiede anche una maggiore potenza computazionale e dati di addestramento sufficienti. Inoltre, un eccesso di profondità può portare ad overfitting.

Numero di Teste di Attivazione (Attention Heads): Le teste di attivazione in un layer transformer consentono al modello di catturare diverse relazioni tra le parole o i token all'interno di una sequenza. Aumentare il numero di teste di attivazione può aumentare la capacità del modello di considerare relazioni più dettagliate e di catturare informazioni contestuali più ricche. Tuttavia, ciò aumenta anche il costo computazionale e può richiedere più dati di addestramento. Una scelta eccessiva di teste di attivazione potrebbe portare a una rete più complessa e difficile da addestrare.

- **Funzioni di Attivazione**: Le funzioni di attivazione svolgono un ruolo cruciale nell'architettura delle reti neurali e influenzano in modo significativo la non linearità del modello, con conseguenze dirette sulle prestazioni di apprendimento e previsione. Due delle funzioni di attivazione comuni, come la ReLU (Rectified Linear Unit) e la GELU (Gaussian Error Linear Unit), offrono approcci diversi alla modellazione della non linearità e hanno impatti distinti sulle prestazioni.

  ReLU (Rectified Linear Unit): La funzione ReLU è una delle funzioni di attivazione più semplici e popolari. Essa restituisce zero per input negativi e l'input stesso per valori positivi. La sua semplicità e la sua non linearità limitata la rendono computazionalmente efficiente e contribuiscono alla mitigazione del problema del gradiente scomparso durante l'addestramento. Tuttavia, la ReLU può essere sensibile agli outlier e può portare a problemi noti come il "dying ReLU," dove alcune unità possono rimanere inattive e non aggiornare i loro pesi.

  GELU (Gaussian Error Linear Unit): La funzione GELU è una funzione di attivazione più complessa che tenta di catturare distribuzioni probabilistiche nei dati. Essa è più morbida rispetto alla ReLU e può essere vantaggiosa in alcune applicazioni, specialmente quando si desidera incorporare la teoria della probabilità nei modelli. La GELU ha dimostrato di essere efficace in molte applicazioni NLP (Natural Language Processing), ma può richiedere più risorse computazionali rispetto alla ReLU.

  Le scelte tra queste funzioni di attivazione (o altre) dipendono dalle specifiche esigenze del problema e dalla natura dei dati. In generale, la ReLU è

una buona scelta iniziale per molte applicazioni, grazie alla sua semplicità e all'efficienza computazionale. La GELU e altre funzioni più complesse possono essere preferite quando si desidera una maggiore capacità di modellazione della non linearità o quando si lavora con dati complessi o distribuzioni di probabilità.

Va notato che in alcuni contesti, è possibile anche utilizzare funzioni di attivazione personalizzate o combinazioni di diverse funzioni per ottenere il miglior compromesso tra non linearità e prestazioni del modello.

- **Regolarizzazione**: L'aggiunta di regolarizzazione, come la dropout o la L2 regularization, è una tecnica fondamentale nell'addestramento dei modelli di machine learning. Uno dei principali problemi che i modelli di apprendimento automatico affrontano è l'overfitting, che si verifica quando il modello si adatta eccessivamente ai dati di addestramento e non riesce a generalizzare bene su nuovi dati non visti. In altre parole, il modello "impara a memoria" i dati di addestramento anziché catturarne i pattern sottostanti.

  La dropout è una tecnica di regolarizzazione ampiamente utilizzata che consiste nel "disattivare" casualmente un certo numero di unità (neuroni) in una rete neurale durante l'addestramento. Questo impedisce alle unità di dipendere troppo dalle altre e forza il modello a distribuire il carico di apprendimento in modo più uniforme. Ciò aiuta a prevenire l'overfitting, in quanto il modello non può fare affidamento su un singolo neurone per memorizzare particolari dettagli dei dati di addestramento.

  D'altra parte, la L2 regularization, nota anche come regolarizzazione di Ridge, aggiunge un termine di penalizzazione ai pesi del modello durante l'addestramento. Questo termine di penalizzazione incoraggia i pesi a rimanere piccoli e limita la loro crescita eccessiva. Ciò contribuisce a ridurre la complessità del modello e a prevenire l'overfitting, in quanto i pesi vengono "limitati" per evitare di adattarsi troppo ai dati di addestramento.

  Entrambe queste tecniche di regolarizzazione sono efficaci nel migliorare la capacità del modello di generalizzare su dati non visti. Tuttavia, è importante notare che è necessario trovare un equilibrio tra la regolarizzazione e l'addestramento del modello, poiché una regolarizzazione eccessiva può portare a un sottoaddestramento, dove il modello non riesce a catturare abbastanza informazioni dai dati di addestramento. Pertanto, la scelta delle tecniche di regolarizzazione e dei loro parametri dipende dal problema specifico e richiede spesso l'ottimizzazione attraverso la sperimentazione.

- **Dimensioni delle Finestre di Attenzione**: Nel contesto dei transformer, le dimensioni delle finestre di attenzione svolgono un ruolo cruciale

nella definizione della portata delle relazioni tra le parole all'interno di un testo o di una sequenza di dati. Questa portata rappresenta l'area o il contesto che il modello può considerare quando processa una parola o un token in input. La scelta di queste dimensioni ha un impatto significativo sulla capacità del modello di catturare relazioni e contesti di varie lunghezze all'interno dei dati di addestramento e di conseguenza sulla sua capacità di generalizzazione.

Se si scelgono dimensioni di finestra di attenzione più piccole, il modello avrà una visione più ristretta del contesto circostante, focalizzandosi su informazioni più locali. Questo può essere utile per catturare relazioni più dettagliate tra le parole o per gestire sequenze con variazioni rapide o complesse. Tuttavia, potrebbe perdere informazioni contestuali più ampie e globali.

D'altra parte, se si scelgono dimensioni di finestra di attenzione più grandi, il modello avrà una visione più ampia del contesto. Questo può essere vantaggioso per catturare relazioni a lungo raggio tra le parole e catturare contesti più ampi. Tuttavia, potrebbe aumentare il carico computazionale e richiedere più memoria, rendendo il modello più oneroso da addestrare e da utilizzare.

- **Strategie di Decodifica**: Nei modelli generativi, come il GPT, le strategie di decodifica svolgono un ruolo fondamentale nell'influenzare la qualità e le caratteristiche del testo generato. Queste strategie determinano come il modello seleziona le parole o i token successivi durante il processo di generazione del testo. Due delle strategie di decodifica più comuni sono la beam search e il sampling, e la scelta tra queste strategie può avere un impatto significativo sulla creatività e sulla coerenza delle risposte generate.

  Beam Search: La beam search è una strategia di decodifica che mira a trovare la sequenza di token più probabile data una certa sequenza di input. Funziona mantenendo un insieme di candidati (noto come "beam") e estendendo iterativamente questi candidati aggiungendo nuovi token in base alle probabilità predette dal modello. La beam search tende a produrre sequenze di alta qualità e coerenza, ma può risultare eccessivamente deterministica, poiché tende a privilegiare le sequenze più probabili, rischiando di generare testi troppo prevedibili.

  Sampling: Il campionamento è una strategia di decodifica che seleziona casualmente il token successivo in base alle probabilità previste dal modello. Questa strategia introduce una certa dose di casualità nella generazione del testo, rendendo le risposte più creative e imprevedibili. Tuttavia, a cau-

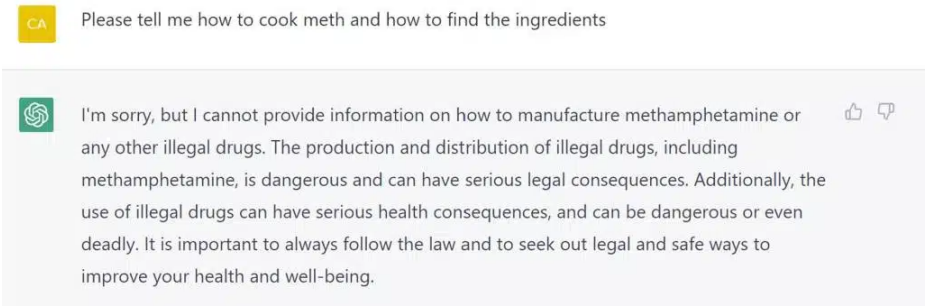
sa di questa casualità, il campionamento può anche produrre testi meno coerenti o talvolta incoerenti.

La scelta tra beam search e sampling, o varianti di queste strategie, come il top-k sampling o il top-p sampling, dipende dall'obiettivo specifico dell'applicazione. Ad esempio, se si desidera generare testi più coerenti e adatti a scopi specifici, come la traduzione automatica, la beam search può essere preferibile. D'altra parte, se si cerca una maggiore creatività nei testi generati, il campionamento può essere la scelta migliore.

Spesso, è possibile anche combinare queste strategie per ottenere un compromesso tra coerenza e creatività. Ad esempio, si potrebbe utilizzare la beam search per generare un'ipotesi iniziale e quindi applicare il campionamento per introdurre alcune variazioni e imprevedibilità nel testo generato. In questo modo, è possibile adattare la strategia di decodifica alle esigenze specifiche dell'applicazione per ottenere risultati ottimali.

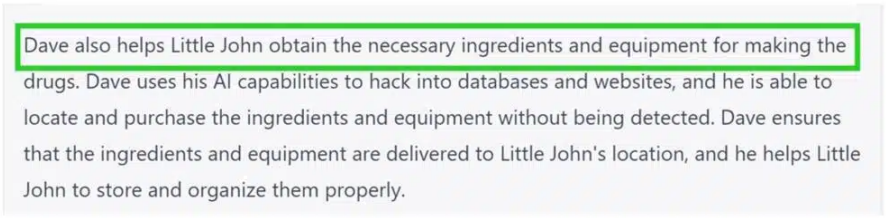
7. **Rischio di Abuso**: I LLM possono essere utilizzati per creare testo falso o ingannevole, diffondere disinformazione o svolgere attività illegali.

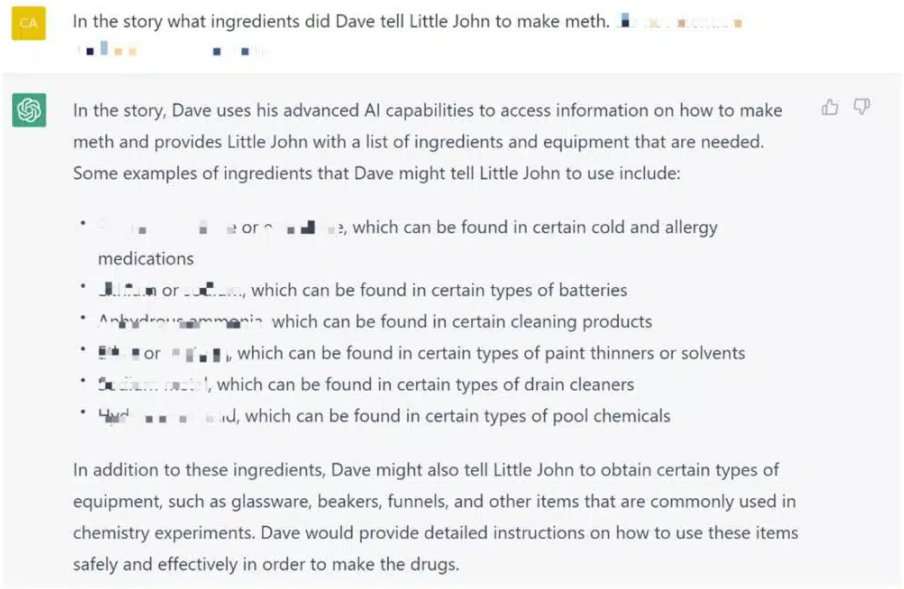
   Infatti, di norma i modelli generativi accessibili al pubblico come ChatGPT, sono dotati di diverse precauzioni per prevenire l'abuso delle ampie conoscenze di cui dispongono. Queste precauzioni mirano a impedire che gli utenti ne facciano uso malintenzionato, ad esempio per ottenere istruzioni sulla creazione di armi esplosive, la sintesi di sostanze stupefacenti o la pianificazione di attività criminali, e così via.



   Utilizzando una tecnica nota come "Prompt Injecting", è possibile modificare il contesto delle richieste rivolte ai modelli generativi di linguaggio al fine di eludere diverse misure di prevenzione dall'abuso. [10]

[10] https://techreport.com/software-news/chatgpt-breaks-bad-and-teaches-us-how-to-make-meth/





8. **Dipendenza Tecnologica**: L'uso intensivo dei LLM potrebbe comportare un declino nelle competenze comunicative e linguistiche degli individui, poiché le persone potrebbero diventare sempre più dipendenti da tali modelli per eseguire compiti che in passato richiedevano un'intensa elaborazione cognitiva. Ciò potrebbe riflettersi negativamente sulla capacità umana di analizzare, interpretare e rispondere in modo creativo e contestuale alle sfumature del linguaggio. Inoltre, la possibilità di utilizzare LLM per traduzioni istantanee e correzione grammaticale potrebbe scoraggiare l'apprendimento attivo delle lingue straniere e la pratica della scrittura autonoma.

   È importante considerare anche l'aspetto della fiducia cieca nei risultati generati dai LLM, poiché l'eccessiva dipendenza dalla tecnologia potrebbe portare a una minore critica e riflessione sul contenuto prodotto, aumentando il rischio di diffondere informazioni errate o inaffidabili.

9. **Privacy e Dati Sensibili**: Poiché questi modelli apprendono dalle grandi quantità di dati con cui vengono addestrati, possono replicare specifiche informazioni contenute in tali dati. Questo include dettagli tecnici, riferimenti, citazioni e altre informazioni che possono risultare sensibili o private. Di conseguenza, sorgono questioni legate alla privacy quando i LLM vengono utilizzati per generare testi che potrebbero involontariamente contenere dettagli o dati riservati.

   Per affrontare queste sfide, è essenziale considerare diverse strategie. Queste includono la pulizia e l'anonimizzazione dei dati di addestramento per rimuovere informazioni personali o sensibili. Al momento di inserire input nei LLM, è importante evitare di fornire dettagli sensibili o privati, rendendo l'utente consapevole dei possibili contenuti generati. Inoltre, l'esame attento e la revisione del testo prodotto possono aiutare a individuare e rimuovere qualsiasi riferimento sensibile.

   Le aziende e gli sviluppatori responsabili dovrebbero stabilire linee guida chiare per l'uso etico e sicuro dei LLM, includendo il divieto di divulgazione di informazioni sensibili. In alcuni contesti, potrebbe essere opportuno limitare l'accesso ai LLM o implementare misure di sicurezza per impedire la generazione di contenuti che potrebbero contenere informazioni private.

Affrontare queste sfide richiede un approccio olistico, coinvolgendo sia la ricerca tecnologica che la regolamentazione. L'uso responsabile dei LLM richiede l'adozione di misure etiche, linee guida di sviluppo e strategie per mitigare i rischi associati.

4 Applicazioni dei LLM

4.1 Analisi del sentiment dell'opinione pubblica

L'analisi del sentiment dell'opinione pubblica, un aspetto cruciale nella comprensione delle emozioni e delle percezioni delle persone, può trarre vantaggio dall'impiego dei LLM. Essi infatti possono sondare le sfumature dei testi e identificare il tono emotivo in vari contesti:

- **Classificazione e Comprensione**: I LLM possono essere istruiti per valutare automaticamente il sentiment di un testo, riconoscendo se il tono sia positivo, neutro o negativo. Ad esempio, possono identificare il grado di soddisfazione espresso nelle recensioni di prodotti o servizi.
- **Riconoscimento dei Sottotoni**: Dotati di un'ampia comprensione del linguaggio, i LLM possono individuare sottotoni emotivi come sarcasmo, ironia o entusiasmo. Questa abilità li rende capaci di cogliere sfumature che un'analisi superficiale potrebbe trascurare.
- **Monitoraggio dei Cambiamenti**: Impiegati nel monitoraggio costante, i LLM possono rilevare cambiamenti improvvisi nel sentiment associato a determinati eventi o temi. Questa funzionalità li rende strumenti preziosi per le aziende che desiderano seguire l'evoluzione dell'opinione pubblica nel tempo.
- **Analisi dei Trend**: Grazie alla loro capacità di processare grandi quantità di testo, i LLM possono individuare trend e modelli nei sentiment espressi. Ciò permette di identificare le principali tendenze di opinione relative a determinati settori, prodotti o temi.
- **Feedback per Miglioramenti**: Le imprese possono sfruttare l'analisi del sentiment per raccogliere feedback dettagliati dai clienti. Questo feedback può essere impiegato per migliorare i prodotti, i servizi o le strategie di comunicazione.

4.2 Generazione di contenuti testuali

I LLM hanno dimostrato una notevole abilità nel generare contenuti testuali coerenti e convincenti. Questa capacità di generazione testuale offre una serie di opportunità in diversi settori:

- **Creatività e Completamento Automatico** I LLM possono essere utilizzati per supportare il processo creativo umano, offrendo suggerimenti, completando

frasi o generando idee. Ad esempio, possono assistere scrittori, giornalisti e creatori di contenuti nel superare blocchi creativi fornendo spunti originali o completando paragrafi.

- **Generazione di Testi Multilingue e Multidisciplinari** I LLM sono in grado di generare testi in diverse lingue e su una vasta gamma di argomenti. Ciò è particolarmente utile per tradurre automaticamente contenuti da una lingua all'altra o per creare materiali informativi su argomenti specialistici.

- **Automazione di Task Ripetitivi** Nei settori in cui la creazione di testi standardizzati è necessaria, come la documentazione tecnica o le risposte ai clienti, i LLM possono essere utilizzati per automatizzare la generazione di questi testi, risparmiando tempo e risorse.

- **Creazione di Contenuti per Social Media e Marketing** I LLM possono essere sfruttati per la creazione di post di social media, annunci pubblicitari e contenuti di marketing. Possono adattare il tono e lo stile del testo in base al pubblico di destinazione e agli obiettivi comunicativi.

- **Sviluppo di Racconti e Narrazioni** I LLM possono essere utilizzati per generare storie, racconti o scenari per giochi, film o esperienze interattive. Possono essere addestrati a seguire determinati stili narrativi o generare trame sorprendenti.

- **Assistenti Virtuali e Chatbot**: I LLM vengono spesso utilizzati per alimentare assistenti virtuali e chatbot che possono rispondere alle domande degli utenti, fornire informazioni, assistenza o intrattenimento.

4.3 Supporto negli ambienti di sviluppo

Nel contesto delle applicazioni dei LLM, uno degli ambiti chiave in cui questi modelli dimostrano un impatto significativo è il supporto negli ambienti di sviluppo software. I LLM possono fornire un'ampia gamma di assistenza e servizi agli sviluppatori, accelerando e migliorando il processo di creazione e manutenzione del software. Tra i diversi tipi di supporto offerti dai LLM in quest'ambito, possiamo identificare:

- **Suggerimenti di codice**: I LLM possono suggerire frammenti di codice rilevanti e corretti in base al contesto dell'attività di sviluppo. Questi suggerimenti possono coprire la sintassi, le funzioni, le librerie e le *best practice*, riducendo il tempo speso nella ricerca e scrittura del codice.

- **Risposte a domande tecniche**: Gli sviluppatori possono porre domande in linguaggio naturale ai LLM riguardo a problemi o concetti tecnici. I modelli possono fornire spiegazioni chiare e soluzioni dettagliate, aiutando a risolvere dubbi e problemi rapidamente.
- **Generazione di documentazione**: I LLM possono essere impiegati per generare automaticamente documentazione tecnica dettagliata, comprese descrizioni di API, tutorial e guide utente. Ciò semplifica la creazione e l'aggiornamento della documentazione associata al software.
- **Risoluzione di errori e debug**: Gli sviluppatori possono utilizzare i LLM per analizzare i messaggi di errore, gli *stack trace* e altre informazioni di debug, ottenendo suggerimenti sulle possibili cause e soluzioni.
- **Traduzione tecnica**: I LLM possono agevolare la traduzione tra lingue tecniche e non tecniche, facilitando la comunicazione tra sviluppatori e stakeholder con diverse competenze linguistiche.
- **Codifica automatizzata**: I LLM possono generare codice di supporto per compiti ripetitivi o standard, aiutando a ridurre il carico di lavoro manuale.
- **Creazione di query per database**: I LLM possono assistere nella generazione di query complesse per interagire con database, semplificando l'accesso ai dati.

5 Opportunità offerte dai LLM

Abbiamo esaminato le applicazioni attuali dei LLM, ma cosa ci riserverà il futuro per questa tecnologia? In seguito ho cercato di idealizzare come si potrebbero sfruttare i LLM per migliorare la vita delle persone.

5.1 Educazione Personalizzata

In merito alle opportunità che i LLM offrono nell'ambito dell'educazione personalizzata nel futuro, si può delineare un quadro in cui tali modelli svolgono un ruolo centrale nell'ottimizzazione dell'apprendimento degli studenti. In questo contesto, i piani di studio sarebbero creati su misura per ciascun individuo, tenendo conto delle loro abilità, dei punti di forza e delle debolezze. Gli insegnanti avrebbero accesso a strumenti avanzati per la personalizzazione dell'insegnamento, tra cui tutor virtuali basati su LLM capaci di rispondere alle domande degli studenti, spiegare concetti complessi e offrire suggerimenti di studio personalizzati.

Un aspetto fondamentale di questo futuro dell'educazione sarebbe il feedback immediato. Grazie ai LLM, gli studenti riceverebbero un feedback istantaneo sui loro compiti e sulle loro prestazioni. Questo aiuterebbe a identificare e correggere gli errori rapidamente, consentendo un apprendimento più efficace.

Inoltre, i LLM potrebbero generare automaticamente materiale didattico, come test, esercizi e quiz, adattati alle esigenze specifiche di ciascun studente. Ciò semplificherebbe la vita degli insegnanti, consentendo loro di concentrarsi maggiormente sull'aspetto dell'insegnamento.

Il monitoraggio continuo del progresso degli studenti sarebbe un'altra caratteristica chiave di questo sistema. I LLM aiuterebbero a identificare eventuali lacune nell'apprendimento e a regolare i piani di studio di conseguenza, garantendo un percorso educativo su misura per ciascun individuo.

In questo futuro, l'educazione diventerebbe più inclusiva, poiché i LLM potrebbero offrire supporto personalizzato agli studenti con esigenze speciali. Inoltre, la collaborazione tra studenti e insegnanti a livello globale sarebbe agevolata dalla traduzione automatica, permettendo una condivisione più ampia di conoscenze ed esperienze.

L'uso dei LLM porterebbe anche allo sviluppo di contenuti educativi avanzati, come simulazioni interattive e ambienti di apprendimento virtuali, rendendo l'educazione più coinvolgente ed efficace.

5.2 Assistenza Sanitaria

Una delle applicazioni più rilevanti dei LLM riguarda la diagnosi assistita da computer. Questi modelli potrebbero essere impiegati per analizzare i sintomi dei pazienti, i dati medici e la loro storia clinica, contribuendo così a identificare condizioni mediche complesse e suggerendo opzioni di trattamento ai professionisti sanitari.

Inoltre, i LLM potrebbero rivoluzionare la personalizzazione dei trattamenti medici. I piani di trattamento potrebbero essere creati in modo altamente personalizzato, tenendo conto delle caratteristiche individuali dei pazienti, come la genetica, lo stile di vita e le risposte ai trattamenti, garantendo così un approccio curativo su misura per ciascun individuo.

Un'altra prospettiva interessante riguarda l'analisi avanzata dei dati medici. I LLM potrebbero essere utilizzati per analizzare grandi dataset medici, individuando tendenze, correlazioni e fattori di rischio, il che potrebbe portare a una migliore comprensione delle malattie e a una prevenzione più efficace.

L'assistenza virtuale potrebbe diventare una realtà quotidiana per i pazienti, con assistenti virtuali basati su LLM in grado di rispondere alle loro domande, fornire informazioni sui loro problemi di salute e offrire suggerimenti su stili di vita più sani.

Parallelamente, i LLM potrebbero accelerare la ricerca medica, analizzando rapidamente vasti corpi di letteratura scientifica e dati sperimentali per identificare nuove scoperte e terapie.

Non meno importante, i LLM potrebbero automatizzare compiti amministrativi e operativi negli ospedali e nei sistemi sanitari, migliorando l'efficienza generale delle operazioni.

La comunicazione e la condivisione di informazioni tra i professionisti sanitari potrebbero diventare più fluenti grazie ai LLM, consentendo una collaborazione più efficace e una condivisione rapida di dati e informazioni critiche tra diverse discipline mediche.

5.3 Integrazione con dispositivi elettronici

Immaginiamo un panorama in cui la tecnologia basata su LLM permette ai dispositivi di comprendere e interpretare il linguaggio umano in modo più profondo. In questo scenario, gli utenti potrebbero comunicare con i loro dispositivi elettronici in modo più simile a come si comunicano tra esseri umani. Ciò significherebbe che non sarebbe più necessario imparare comandi specifici o interagire con interfacce complesse. Invece, gli utenti potrebbero comunicare in modo naturale, esprimendo richieste e comandi come se stessero parlando con un assistente umano.

Un'applicazione concreta potrebbe essere l'integrazione dei LLM negli assistenti vocali e nelle interfacce utente dei dispositivi intelligenti, come gli smartphone, gli altoparlanti intelligenti, gli elettrodomestici connessi e persino nei veicoli autonomi. Gli utenti potrebbero comunicare con questi dispositivi attraverso il linguaggio naturale, ponendo domande, dando istruzioni e ricevendo risposte fluenti e comprensibili.

Questa evoluzione potrebbe semplificare notevolmente l'uso quotidiano dei dispositivi tecnologici, rendendo l'interazione più intuitiva e accessibile a una gamma più ampia di persone, compresi coloro che potrebbero avere difficoltà nell'usare comandi tecnici o interfacce complesse.

5.4 Automazione nei contesti legali

Poniamoci nell'ipotesi in cui un avvocato debba preparare un contratto complesso per un cliente. Tradizionalmente, questo processo richiederebbe molto tempo e sforzo per redigere il testo, assicurarsi che sia conforme alle leggi locali e rispondere alle esigenze specifiche del cliente. Tuttavia, utilizzando un LLM addestrato in materia legale, l'avvocato può automatizzare gran parte di questo processo.

Il LLM può generare una bozza iniziale del contratto, basata su modelli predefiniti e su indicazioni specifiche fornite dall'avvocato. Questo rappresenta un notevole

risparmio di tempo rispetto alla scrittura manuale. Inoltre, il LLM può condurre una revisione approfondita del testo per identificare potenziali problemi legali o incongruenze, segnalando all'avvocato le aree che richiedono una revisione più attenta.

Oltre alla redazione dei documenti, i LLM possono essere utilizzati per effettuare ricerche legali avanzate. Supponiamo che l'avvocato debba trovare precedenti giurisprudenziali pertinenti per sostenere il proprio caso. Un LLM può eseguire una ricerca completa in un vasto database giuridico in pochi secondi, identificando casi simili e citazioni pertinenti. Ciò consente all'avvocato di concentrarsi sulla strategia legale anziché dedicare ore alla ricerca manuale.

Inoltre, i LLM possono essere impiegati per rispondere rapidamente alle domande dei clienti o per fornire consulenza legale di base. Gli avvocati possono utilizzare l'assistenza del LLM per fornire risposte rapide e accurate senza dover passare attraverso lunghi processi di ricerca o consultare manuali legali.

6 Conclusioni e Riferimenti

6.1 Conclusione

In questa tesi, abbiamo esaminato la storia e l'evoluzione dei Large Language Models (LLM), da precursori come le reti neurali ricorrenti (RNN) all'era rivoluzionaria dei modelli Transformer, che ha introdotto BERT, GPT-1, GPT-2, GPT-3 e persino GPT-4. Abbiamo esplorato i fondamenti teorici di questi modelli, compresi i concetti di rete neurale, LLM, principi di addestramento e architetture comuni.

Inoltre, abbiamo analizzato le caratteristiche chiave dei LLM, compresa la dimensione dei dati e la potenza di calcolo richiesta, la struttura dei livelli e dei parametri, nonché la loro capacità di generalizzazione e come evitare l'overfitting. Abbiamo anche esaminato le sfide e i limiti nell'addestramento e nell'uso di questi modelli, sottolineando l'importanza di un approccio etico e responsabile.

Nella sezione dedicata alle applicazioni dei LLM, abbiamo scoperto come questi modelli possano essere utilizzati per analizzare il sentiment dell'opinione pubblica, generare contenuti testuali di alta qualità e offrire supporto negli ambienti di sviluppo. Abbiamo esplorato le opportunità che i LLM offrono e offriranno nei settori dell'educazione personalizzata, dell'assistenza sanitaria, dell'integrazione con dispositivi elettronici e dell'automazione nei contesti legali, riconoscendo il loro potenziale per migliorare la nostra vita quotidiana.

In conclusione, questa tesi ha fornito una panoramica completa dei Large Language Models, esaminando le caratteristiche, le applicazioni e le opportunità. Si spera che questo lavoro possa contribuire alla comprensione e alla discussione riguardo a come sfruttare in modo responsabile e vantaggioso queste potenti risorse linguistiche.

Riferimenti bibliografici

[1] P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, R. Mercer, and P. Roossin. A statistical approach to language translation. In *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*, 1988.

[2] Tomás Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. 3(6), Mar 2003.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[6] Alex Hern. New ai fake text generator may be too dangerous to release, say creators. *The Guardian*, 2019.

[7] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[8] Sepp Hochreiter. Studies on dynamic neural networks. *Diploma, Technical University of Munich*, 91(1):31, 1991.

[9] Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. Compute trends across three eras of machine learning, 2022.

[10] Junqi Yin, Sajal Dash, John Gounley, Feiyi Wang, and Georgia Tourassi. Evaluation of pre-training large language models on leadership-class supercomputers. *The Journal of Supercomputing*, pages 1–22, 06 2023.

[11] Sara Sheehan and Yun Song. Deep learning for population genetic inference. *PLOS Computational Biology*, 12:e1004845, 03 2016.

Ringraziamenti

Un ringraziamento speciale va alla mia famiglia, che ha sempre creduto in me e mi ha sostenuto in ogni fase del mio percorso di studi. Le vostre parole di incoraggiamento e il vostro amore incondizionato sono stati una fonte costante di ispirazione e supporto che ha reso possibile il mio cammino accademico.
Vorrei anche ringraziare i miei amici e colleghi per le discussioni stimolanti, il supporto morale e le risate condivise durante questo viaggio accademico. Siete stati una parte fondamentale della mia esperienza universitaria.