ORIENS

AI ACADEMY

FINE-TUNING DI LLM

Partners in Growth

SOMMARIO

Oriens Consulting S.r.l. a socio unico
Via Zamenhof 200, Vicenza 36100
+39 0444 1834081 - C.F. e P. IVA: 03801360243
info@oriens.consulting – oriens.consulting

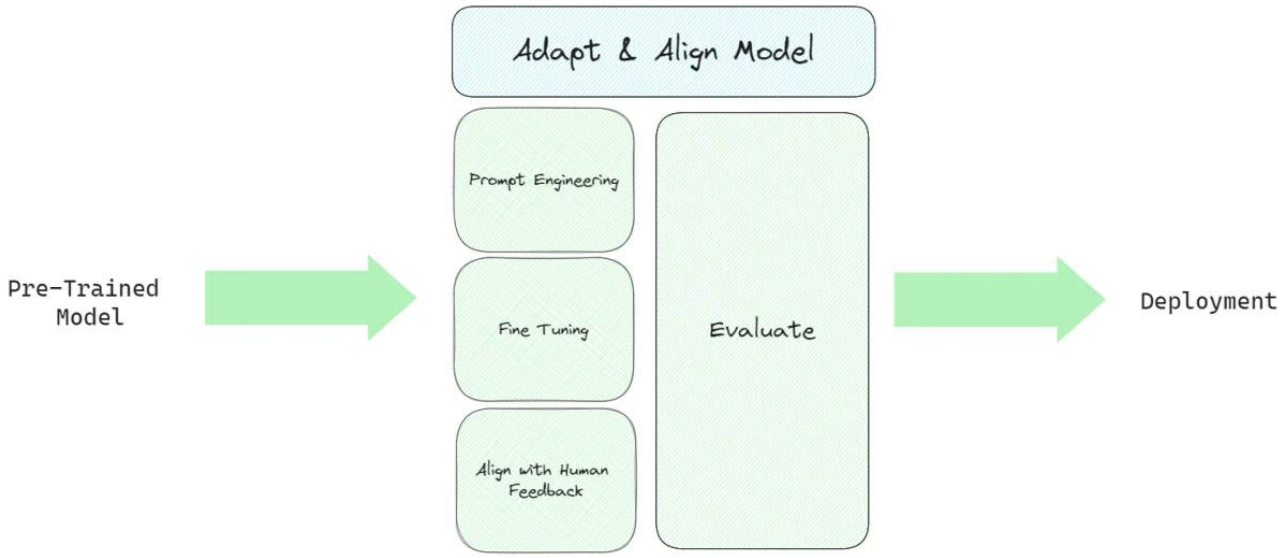
L'APPROCCIO FINE-TUNING

Un aspetto importante per sfruttare il potenziale di questi LLM risiede nel processo di messa a punto, una strategia che consente la personalizzazione di modelli pre-addestrati per soddisfare compiti specifici con precisione. È attraverso questa messa a punto che questi modelli possono veramente allinearsi ai requisiti individuali, offrendo soluzioni innovative e su misura per esigenze specifiche.

La messa a punto del LLM è più di un miglioramento tecnico; è un aspetto cruciale dello sviluppo del modello LLM che consente un'applicazione più specifica e raffinata in vari compiti. La messa a punto regola i modelli pre-addestrati per adattarsi meglio a set di dati specifici, migliorandone le prestazioni in attività particolari e garantendo un'applicazione più mirata. Mette in evidenza la straordinaria capacità degli LLM di adattarsi ai nuovi dati, dimostrando la flessibilità che è vitale nel crescente interesse per le applicazioni di intelligenza artificiale.

La messa a punto di modelli linguistici di grandi dimensioni apre molte opportunità, consentendo loro di eccellere in compiti specifici che vanno dall'analisi del sentiment alle revisioni della letteratura medica. Adattando il modello base a un caso d'uso specifico, sblocchiamo nuove possibilità, migliorando l'efficienza e la precisione del modello. Inoltre, facilita un utilizzo più economico delle risorse di sistema, poiché la messa a punto richiede meno potenza di calcolo rispetto all'addestramento di un modello da zero.

Il ciclo di vita, illustrato nella figura seguente, è caratterizzato dall'integrazione di input e output di istruzioni, accoppiati con esempi di ragionamento passo passo. Questo approccio facilita il modello nel generare risposte che non solo sono rilevanti ma anche perfettamente allineate con le istruzioni specifiche in esso inserite. È durante questa fase che i modelli preaddestrati vengono adattati per risolvere compiti e casi d'uso distinti, utilizzando set di dati personalizzati per migliorarne la funzionalità.



La messa a punto di un singolo compito si concentra sull'affinamento delle competenze del modello in un compito specifico, come il riepilogo. Questo approccio è particolarmente utile per ottimizzare i flussi di lavoro che coinvolgono documenti sostanziali o thread di conversazione, inclusi documenti legali e ticket di assistenza clienti. Sorprendentemente, questa messa a punto può ottenere miglioramenti significativi delle prestazioni con un insieme relativamente piccolo di esempi, che vanno da 500 a 1000, in contrasto con i miliardi di token utilizzati nella fase di pre-addestramento.

Il viaggio alla comprensione della messa a punto del LLM inizia con la comprensione degli elementi fondamentali che costituiscono i LLM. Al centro di questi modelli c'è il architettura del trasformatore, una rete neurale che sfrutta i meccanismi di auto-attenzione per dare priorità al contesto delle parole rispetto alla loro vicinanza in una frase. Questo approccio innovativo facilita una comprensione più profonda delle relazioni distanti tra i token nell'input.

Mentre esploriamo le complessità dei trasformatori, incontriamo un processo in più fasi che inizia con il codificatore. Questa fase iniziale prevede la tokenizzazione

dell'input e la creazione di vettori di incorporamento che rappresentano l'input e la sua posizione nella frase. Le fasi successive prevedono una serie di calcoli utilizzando matrici note come domanda, Valore e Le, culminando in un punteggio di auto-attenzione che detta l'attenzione su diverse parti della frase e vari token.

La messa a punto rappresenta una fase critica nello sviluppo degli LLM, un processo che implica apportare sottili aggiustamenti per ottenere risultati più desiderabili. Questa fase, sebbene essenziale, presenta una serie di sfide, comprese le esigenze computazionali e di archiviazione legate alla gestione di un vasto numero di parametri. Il parametro Efficient Fine-Tuning (PEFT) offre tecniche per ridurre il numero di parametri da ottimizzare, semplificando così il processo di formazione.

Pre-formazione LLM: stabilire una base solida

Nelle fasi iniziali dello sviluppo LLM, la pre-formazione è al centro dell'attenzione, utilizzando trasformatori sovraparametrizzati come architettura fondamentale. Questo processo implica la modellazione del linguaggio naturale in vari modi come bidirezionale, autoregressivo o sequenza per sequenza su corpora non supervisionati su larga scala. L'obiettivo qui è quello di creare una base che possa essere messa a punto in seguito per specifici compiti a valle attraverso l'introduzione di obiettivi specifici per compito

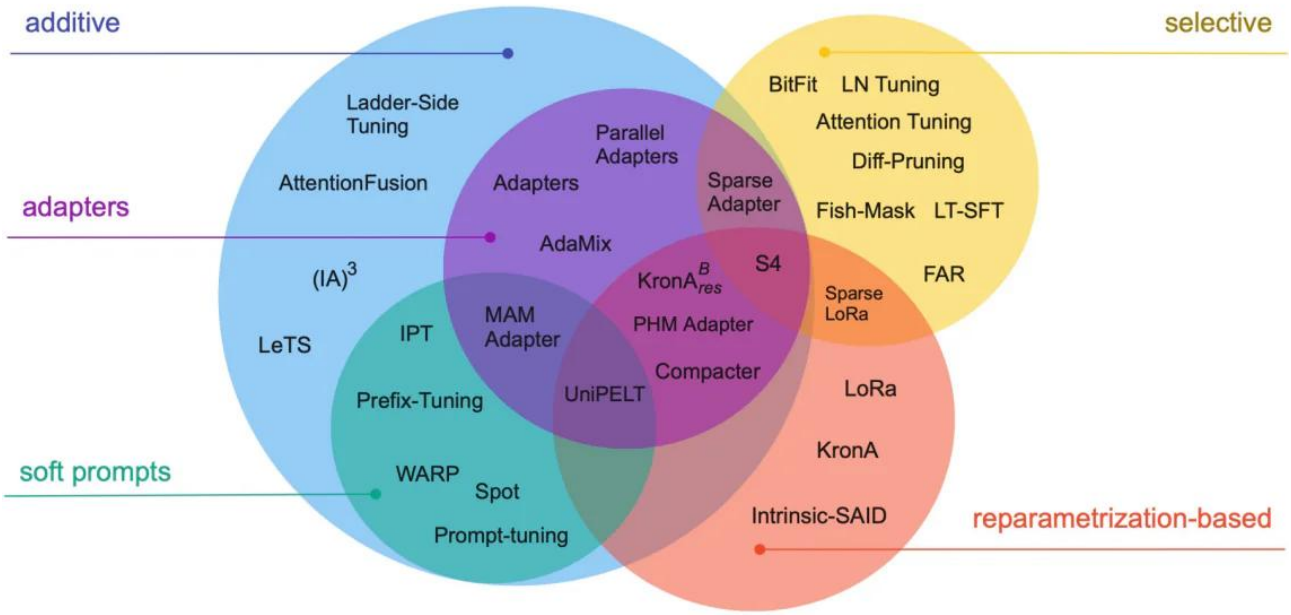
Una tendenza degna di nota in questo ambito è l'inevitabile aumento della scala dei LLM pre-formati, misurata dal numero di parametri. I dati empirici mostrano costantemente che modelli più grandi abbinati a più dati producono quasi sempre prestazioni migliori. Ad esempio, il GPT-3, con i suoi 175 miliardi di parametri, ha stabilito un punto di riferimento nella generazione di un linguaggio naturale di alta qualità e nell'esecuzione competente di un'ampia gamma di attività zero-shot.

Messa a punto: il percorso verso l'adattamento del modello

Dopo la pre-formazione, il LLM viene sottoposto a perfezionamento per adattarsi a compiti specifici. Nonostante le prestazioni promettenti mostrate dall'apprendimento in contesto in LLM pre-addestrati come GPT-3, la messa a punto rimane superiore in contesti specifici delle attività. Tuttavia, l'approccio prevalente di messa a punto completa dei parametri presenta sfide, tra cui elevate esigenze computazionali e di memoria, soprattutto quando si ha a che fare con modelli su larga scala.

Per modelli linguistici di grandi dimensioni con oltre un miliardo di parametri, la gestione efficiente della RAM della GPU è fondamentale. Un singolo parametro del modello con precisione completa a 32 bit richiede 4 byte di spazio, che si traducono in un requisito di 4 GB di RAM GPU solo per caricare un modello da 1 miliardo di parametri. L'effettivo processo di addestramento richiede ancora più memoria per ospitare vari componenti tra cui stati e gradienti dell'ottimizzatore, richiedendo potenzialmente fino a 80 GB di RAM GPU per un modello di questa scala.

Per superare i limiti della RAM della GPU, viene utilizzata la quantizzazione, una tecnica che riduce la precisione dei parametri del modello, diminuendo così i requisiti di memoria. Ad esempio, modificando la precisione da 32 bit a 16 bit è possibile dimezzare la memoria necessaria sia per caricare che per addestrare il modello



Nel processo di messa a punto completa dei modelli linguistici di grandi dimensioni, è importante disporre di una configurazione computazionale in grado di gestire in modo efficiente non solo il peso sostanziale dei modelli, che per i modelli più avanzati stanno ora raggiungendo dimensioni di centinaia di gigabyte, ma anche di gestire una serie di altri elementi critici. Questi includono l'allocazione della memoria per gli stati dell'ottimizzatore, la gestione dei gradienti, le attivazioni successive e la facilitazione della memoria temporanea durante le varie fasi della procedura di addestramento.

Metodo additivo

Questo tipo di ottimizzazione può aumentare il modello pre-addestrato con parametri o livelli aggiuntivi, concentrandosi sull'addestramento solo dei parametri appena aggiunti. Nonostante l'aumento del conteggio dei parametri, questi metodi migliorano i tempi di formazione e l'efficienza dello spazio. Il metodo additivo è ulteriormente suddiviso in sottocategorie:

- Adattatori RF: Incorporando sottostrati post-trasformatori di piccole reti completamente connesse, con esempi degni di nota AdaMix, KronAe Compattatore.
- Suggerimenti morbidi: Ottimizzazione di un segmento degli incorporamenti di input del modello attraverso la discesa del gradiente, con IPT, sintonizzazione del prefissoe WARP ne sono esempi importanti.
- Altri approcci additivi: Include tecniche come LeTS, AttentionFusion e Ladder-Side Tuning

Metodo selettivo

I PEFT selettivi mettono a punto un numero limitato di strati superiori in base al tipo di strato e alla struttura del modello interno. Questa categoria include metodi come BitFit e a un LN tuning, che si concentra sulla messa a punto di elementi specifici come i pregiudizi del modello o righe particolari.

Metodo basato sulla riparametrizzazione

Questi metodi utilizzano rappresentazioni di basso rango per ridurre il numero di parametri addestrabili, il più noto dei quali è l'adattamento di basso rango o LoRA. Questo metodo sfrutta una semplice scomposizione della matrice di basso rango per parametrizzare l'aggiornamento del peso, dimostrando un'efficace messa a punto nei sottospazi di basso rango.

1) LoRA (adattamento di basso rango)

LoRA è emersa come una tecnica PEFT innovativa, introdotta in un articolo da Edward J. Hu e altri nel 2021. Opera all'interno della categoria di riparametrizzazione, congelando i pesi originali dell'LLM e integrando nuove matrici di basso rango addestrabili in ogni strato dell'architettura Transformer. Questo approccio non solo riduce il numero di parametri addestrabili, ma diminuisce anche il tempo di addestramento e le risorse computazionali necessarie, presentando così un'alternativa

più efficiente alla messa a punto completa.

Per comprendere i meccanismi di LoRA, è necessario rivisitare l'architettura del trasformatore in cui il prompt di input viene sottoposto a tokenizzazione e conversione in vettori di incorporamento. Questi vettori attraversano i segmenti codificatore e/o decodificatore del trasformatore, incontrando reti di autoattenzione e feed-forward i cui pesi sono pre-addestrati.

LoRA utilizza il concetto di Decomposizione valore singolare (SVD). Essenzialmente, SVD seziona una matrice in tre matrici distinte, una delle quali è una matrice diagonale che ospita valori singolari. Questi valori singolari sono fondamentali in quanto misurano il significato delle diverse dimensioni nelle matrici, con valori più grandi che indicano maggiore importanza e quelli più piccoli che denotano significato minore.

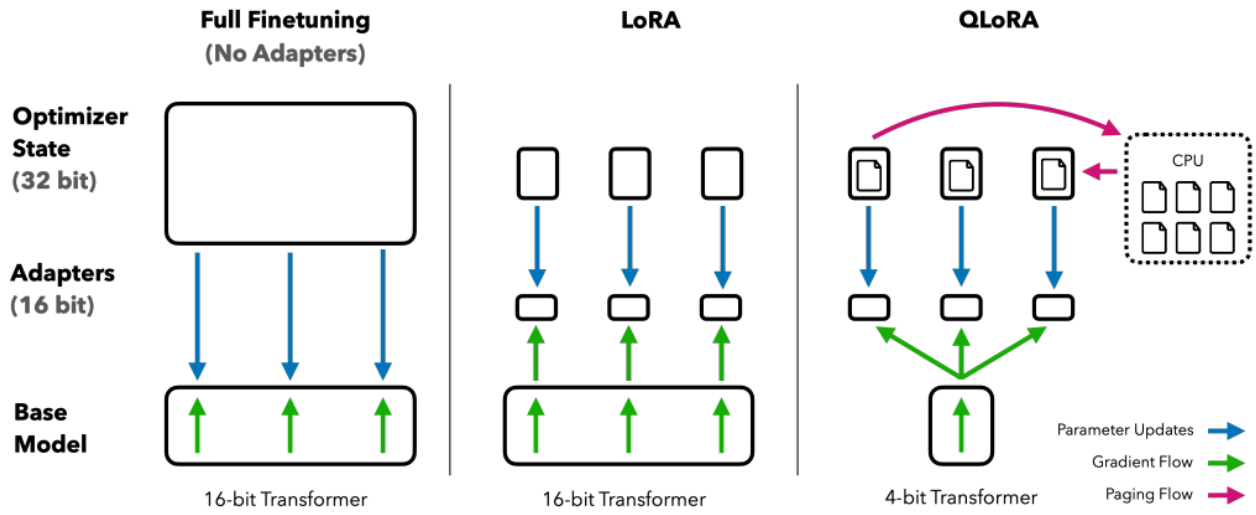
Questo approccio consente a LoRA di mantenere le caratteristiche essenziali dei dati riducendone al contempo la dimensionalità, ottimizzando quindi il processo di messa a punto.

LoRA interviene in questo processo, congelando tutti i parametri del modello originale e introducendo una coppia di “matrici di scomposizione dei ranghi” accanto ai pesi originali. Queste matrici più piccole, denominate A e B, vengono sottoposte ad addestramento attraverso l'apprendimento supervisionato.

L'elemento centrale di questa strategia è il parametro chiamato rango ("r"), che determina la dimensione delle matrici di rango basso. Una meticolosa selezione di "r" può produrre risultati impressionanti, anche con un valore più piccolo, creando così una matrice di basso rango con meno parametri da addestrare. Questa strategia è stata implementata in modo efficace utilizzando librerie open source come HuggingFace Transformers, facilitando la messa a punto di LoRA per varie attività con notevole efficienza.

2) QLoRA: aumentare l'efficienza di LoRA

Basandosi sulle fondamenta gettate da LoRA, QLoRA riduce ulteriormente i requisiti di memoria. Presentato da Tim Dettmers e altri nel 2023, combina l'adattamento di basso rango con la quantizzazione, utilizzando un formato di quantizzazione a 4 bit chiamato Normale Float or nf4. La quantizzazione è essenzialmente un processo che trasferisce i dati da una rappresentazione più informativa a una con meno informazioni. Questo approccio mantiene l'efficacia dei metodi di regolazione fine a 16 bit, dequantizzando i pesi da 4 bit a 16 bit come richiesto durante i processi computazionali



QLoRA sfrutta NumericFloat4 (nf4), prendendo di mira ogni livello dell'architettura del trasformatore, e introduce il concetto di doppia quantizzazione per ridurre ulteriormente l'impronta di memoria richiesta per la messa a punto. Ciò si ottiene eseguendo la quantizzazione sulle costanti già quantizzate, una strategia che evita i tipici picchi di memoria del checkpoint del gradiente attraverso l'utilizzo di ottimizzatori paginati e una gestione unificata della memoria.

guanaco, che è un insieme ottimizzato per QLORA, stabilisce un punto di riferimento

nelle soluzioni chatbot open source. Le sue prestazioni, convalidate attraverso valutazioni sistematiche umane e automatizzate, sottolineano la sua posizione dominante ed efficienza nel settore.

Le versioni 65B e 33B di Guanaco, perfezionate utilizzando una versione modificata del OASST1 set di dati, emergono come formidabili contendenti a modelli rinomati come ChatGPT e persino GPT-4.

Ottimizzazione utilizzando l'apprendimento per rinforzo dal feedback umano

L'apprendimento per rinforzo dal feedback umano (RLHF) entra in gioco quando si perfezionano i modelli linguistici pre-addestrati per allinearli più strettamente ai valori umani. Questo concetto è stato introdotto da Open AI nel 2017 gettando le basi per un riepilogo avanzato dei documenti e lo sviluppo di Istruisci GPT.

Al centro di RLHF c'è il paradigma dell'apprendimento per rinforzo, un tipo di tecnica di apprendimento automatico in cui un agente impara come comportarsi in un Industria XNUMX eseguendo azioni e ricevere premi. È un ciclo continuo di azione e a un feedback, dove l'agente è incentivato a fare scelte che produrranno la ricompensa più alta.

Traducendo questo nel regno dei modelli linguistici, il agente Europe è modello stesso, operante all'interno del Industria XNUMX di una determinata finestra di contesto e prendere decisioni basate su stato, che è definito dai token correnti nella finestra di contesto. IL "spazio di azione" comprende tutti i potenziali token tra cui il modello può scegliere, con l'obiettivo di selezionare il token che si allinea più strettamente alle preferenze umane.

Il processo RLHF sfrutta ampiamente il feedback umano, utilizzandolo per formare un

modello di ricompensa. Questo modello svolge un ruolo cruciale nel guidare il modello pre-addestrato durante il processo di messa a punto, incoraggiandolo a generare risultati più in linea con i valori umani. Si tratta di un processo dinamico e iterativo, in cui il modello apprende attraverso una serie di "lanciamenti", un termine usato per descrivere la sequenza di stati e azioni che portano a una ricompensa nel contesto della generazione del linguaggio.

Uno dei notevoli potenziali di RLHF è la sua capacità di favorire la personalizzazione degli assistenti IA, adattandoli per rispondere alle preferenze dei singoli utenti, che si tratti del loro senso dell'umorismo o delle routine quotidiane. Apre strade per la creazione di sistemi di intelligenza artificiale che non siano solo tecnicamente competenti ma anche emotivamente intelligenti, in grado di comprendere e rispondere alle sfumature della comunicazione umana.

Tuttavia, è essenziale notare che RLHF non è una soluzione infallibile. I modelli sono ancora suscettibili di generare risultati indesiderati, un riflesso dei dati vasti, spesso non regolamentati e distorti su cui sono formati.

APPROCCIO RAG

L'approccio RAG (Retrieval-Augmented Generation) è uno dei metodi più potenti e pratici per creare un LLM aziendale senza dover addestrare da zero un nuovo modello. RAG è un'architettura che unisce due componenti:

1. Retrieval (recupero): cerca le informazioni più rilevanti in una base di conoscenza aziendale (documenti, database, FAQ, ecc.)
2. Generation (generazione): le passa a un LLM generico (es. GPT-4), che le usa per produrre una risposta contestualizzata