PRISMA[1]: Disentanglement e scomposizione in feature monosemantiche delle rappresentazioni latenti nei LLM tramite Sparse Autoencoders

Edoardo Tedesco

1 Introduzione e Problema

L'avvento dei Large Language Models (LLM) ha segnato un punto di svolta nell'elaborazione del linguaggio naturale (Devlin et al. 2018), consentendo alle macchine di apprendere complesse sfumature semantiche. Tuttavia, questa straordinaria capacità predittiva ha un costo: la perdita di interpretabilità. I modelli operano proiettando le parole in spazi vettoriali ad alta dimensionalità (embedding densi), i cui assi non possiedono significato intrinseco. Idealmente, per garantire trasparenza, vorremmo che i singoli neuroni corrispondessero a concetti isolati: un neurone che si attiva solo in presenza di "colore rosso" o "muso di cane". In termini formali, l'obiettivo è ottenere una rappresentazione in cui il numero di dimensioni corrisponda ai **fattori di variazione** dei dati (Wang et al. 2024). Empiricamente, tuttavia, si osserva raramente tale corrispondenza. La discrepanza tra concetti umani e attivazioni neurali è alla base del **problema dell'allineamento** (Elhage et al. 2022): diventa arduo fidarsi di un sistema se non se ne comprendono i meccanismi decisionali. Una risposta teorica è la **Superposition Hypothesis** (Elhage et al. 2022): le reti sfruttano concetti mutuamente esclusivi per rappresentare un numero di feature superiore ai neuroni disponibili (es. "meccanica quantistica" raramente si attiva con "torta al cioccolato"). I modelli "comprimono" le informazioni tramite interferenza controllata. Sebbene efficiente, questo genera *polisemanticità*, rendendo la rete una "black box". Il presente lavoro propone l'implementazione di un metodo per invertire questo processo attraverso il **disentanglement** delle rappresentazioni, proiettando gli embedding densi in spazi semantici sparsi con assi interpretabili (O'Neill et al. 2024). A tale scopo è stata sviluppata **PRISMA**, un'applicazione che, analogamente a un prisma ottico che scompone la luce bianca, isola i concetti atomici costitutivi del testo.

2 PRISMA

2.1 Dagli Autoencoders ai SAE

Un Autoencoder (AE) è una rete neurale non supervisionata che apprende una funzione identità $h(x) \approx x$. È composto da un *encoder* che comprime l'input in una rappresentazione latente z di dimensione inferiore (*bottleneck*), e un *decoder* che ricostruisce l'input da z. Gli AE classici estraggono fattori principali, ma non garantiscono che le dimensioni latenti siano monosemantiche: l'assenza di vincoli espliciti crea uno spazio latente "discontinuo", dove la semantica emerge solo come sottomanifold indotto dai dati. Per ottenere *disentanglement* e feature interpretabili, Prisma utilizza uno **Sparse Autoencoder** (SAE) (O'Neill et al. 2024). A differenza degli AE classici, un SAE usa una rappresentazione latente *overcomplete*: la dimensione latente n è maggiore dell'input d, con expansion factor $\rho = n/d$ tipicamente in [2, 9]. Contestualmente, si impone un forte vincolo di **sparsità**, costringendo solo pochi neuroni ad attivarsi per ogni input. L'intuizione è che l'overcompletezza fornisca molte "direzioni" disponibili, mentre la sparsità forza il modello a selezionarne poche, favorendo feature stabili e interpretabili.

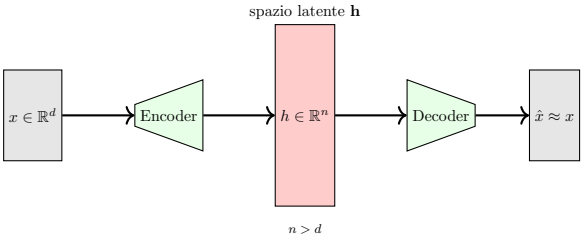

Figura 1: Sparse Autoencoder: spazio latente overcomplete con attivazioni sparse.

2.2 Architettura e Training

Sia $x \in \mathbb{R}^d$ l'input e $h \in \mathbb{R}^n$ la rappresentazione latente con $n > d$. L'encoder mappa l'input tramite $h = \sigma(W_e x + b_e)$ con σ non-linearità (ReLU), mentre il decoder ricostruisce linearmente: $\hat{x} = W_d h + b_d$. La linearità del decoder implica che la ricostruzione è una combinazione lineare dei vettori-feature (colonne di W_d) pesata dalle attivazioni, rendendo ciascuna feature interpretabile come una *direzione* nello spazio degli embedding (O'Neill et al. 2024).

L'obiettivo di addestramento combina una perdita di ricostruzione e un vincolo di sparsità:

$$\mathcal{L} = \frac{1}{d}\|x - \hat{x}\|_2^2 + \lambda \mathcal{L}_{sparse}(h) \tag{1}$$

Il termine $\mathcal{L}_{sparse}$ è implementato tramite un vincolo **Top-K**: durante il forward pass, vengono mantenute solo le k attivazioni maggiori in h, mentre le restanti sono poste a zero (O'Neill et al. 2024).

2.3 Interpretabilità degli assi

Una volta addestrato il SAE, ogni feature può essere interpretata associandole una descrizione testuale. Questa fase avviene tramite un LLM *Interpreter* che, osservando esempi che massimizzano l'attivazione di una feature e contro-esempi, produce un'etichetta semantica (topic/concetto) per quella direzione latente (O'Neill et al. 2024).

3 Analisi: Effective Rank

La matrice delle attivazioni sparse $H \in \mathbb{R}^{N \times n}$ (documenti $\times$ feature) non codifica feature perfettamente indipendenti: esistono co-attivazioni sistematiche tra concetti (es. *febbre* $\leftrightarrow$ *polmonite*), che riducono i gradi di libertà effettivi. Per quantificare questa *dimensione effettiva* adottiamo l'**Effective Rank** (Roy e Vetterli 2007), definito

[1] Acronimo per ***P**rojection of **R**epresentations for **I**nterpretability via **S**parse **M**onosemantic **A**utoencoders.*

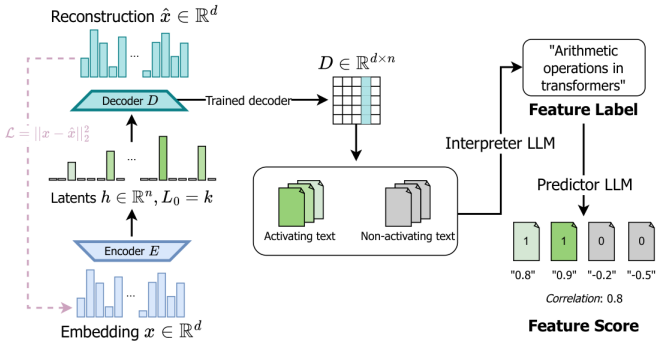

Figura 2: Processo di training e feature labelling del SAE (O'Neill et al. 2024).

tramite l'entropia dello spettro singolare normalizzato:

$$ER(H) = \exp\left(-\sum_i p_i \log p_i\right), \quad p_i = \frac{\sigma_i}{\|\sigma\|_1} \tag{2}$$

dove $\{\sigma_i\}$ sono i valori singolari di H. Intuitivamente, ER è alto se la varianza è distribuita su molte direzioni (codice isotropo), basso se concentrata in poche (codice compresso).

Stimiamo ER al variare dell'expansion factor e confrontiamo con un'*ipotesi nulla* (SAE addestrato su input casuali). Definiamo il **Semantic Compression Ratio**:

$$SCR(\%) = 100 \cdot \frac{ER_{\text{null}} - ER_{\text{real}}}{ER_{\text{null}}} \tag{3}$$

Empiricamente, aumentando la capacità latente, ER cresce in entrambi i casi ma *sistematicamente meno* sui dati reali, indicando che la semantica impone vincoli di co-attivazione che riducono la dimensionalità effettiva.

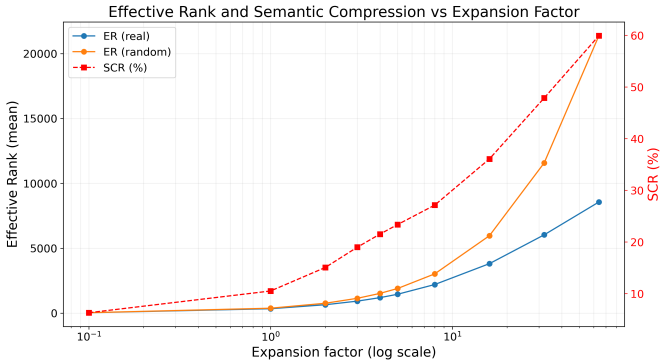

Figura 3: Effective Rank vs expansion factor (real vs null) e Semantic Compression Ratio.

4 Conclusioni

Il lavoro con Prisma dimostra che l'interpretabilità dei LLM può essere recuperata invertendo il processo di *superposition*. Proiettando le attivazioni in uno spazio overcomplete e forzando la sparsità, i SAE isolano feature monosemantiche interpretabili come concetti atomici. Il contributo centrale è l'osservazione che tali concetti non sono indipendenti. L'analisi dell'Effective Rank rivela che lo spazio latente ha dimensionalità effettiva sistematicamente inferiore rispetto all'ipotesi nulla. Questa riduzione, quantificata dal Semantic Compression Ratio, riflette **vincoli semantici reali** che legano i concetti in pattern di co-attivazione. In altri termini, **la semantica si manifesta come riduzione dei gradi di libertà** nello spazio dei concetti estratti. I SAE non eliminano questa struttura di dipendenza: la rendono osservabile e misurabile.

Riferimenti bibliografici

Devlin, Jacob et al. (2018). «Bert: Pre-training of deep bidirectional transformers for language understanding». In: *arXiv preprint arXiv:1810.04805*.

Elhage, Nelson et al. (2022). «Toy Models of Superposition». In: *Transformer Circuits Thread*. URL: `https://transformer-circuits.pub/2022/toy_model/index.html`.

O'Neill, Charles et al. (2024). «Disentangling Dense Embeddings with Sparse Autoencoders». In: *arXiv preprint arXiv:2408.00657*.

Roy, Olivier e Martin Vetterli (2007). «The effective rank: A measure of effective dimensionality». In: *15th European Signal Processing Conference*. IEEE, pp. 606–610.

Wang, Xin et al. (2024). *Disentangled Representation Learning*. arXiv: `2211.11695 [cs.LG]`. URL: `https://arxiv.org/abs/2211.11695`.