Large Language Models

Architettura, Addestramento e Impatti Applicativi

[Il tuo nome]

Anno Accademico 2024–2025

Indice

Capitolo 1

Introduzione

1.1 Introduzione

Contesto scientifico

Motivazioni

Obiettivi della tesi

Breve descrizione dei capitoli

Capitolo 2

Autoencoders

2.1 Introduzione

In questo capitolo vengono introdotti gli *autoencoders*, una classe di modelli di apprendimento non supervisionato ampiamente utilizzata per l'apprendimento di rappresentazioni latenti dei dati. Dopo averne presentato la formulazione generale e i principi di funzionamento, verranno discussi i principali limiti degli autoencoders classici, in particolare in termini di capacità di apprendere rappresentazioni interpretabili.

Successivamente, il capitolo introduce gli *Sparse Autoencoders*, una estensione degli autoencoders tradizionali che impone vincoli di sparsità sullo spazio latente, favorendo il disentanglement delle feature e l'interpretabilità delle rappresentazioni apprese. Questi modelli costituiscono il fondamento teorico delle metodologie utilizzate nel resto del lavoro di tesi.

2.2 Autoencoders: definizione e formulazione generale

2.2.1 Apprendimento non supervisionato

Gli autoencoders sono modelli di apprendimento non supervisionato, in quanto non richiedono etichette associate ai dati di input durante la fase di addestramento. Si consideri un dataset di addestramento S_T costituito da M

osservazioni non etichettate $\mathbf{x}_i$, con $i = 1, \ldots, M$:

$$S_T = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}. \tag{2.1}$$

In generale, ciascuna osservazione appartiene allo spazio $\mathbb{R}^n$, ovvero $\mathbf{x}_i \in \mathbb{R}^n$. L'obiettivo di un autoencoder è apprendere una rappresentazione dei dati tale da permettere la ricostruzione dell'input nel modo più accurato possibile, minimizzando una misura dell'errore di ricostruzione.

L'interesse verso questo tipo di modelli risiede nel fatto che la rappresentazione latente appresa può essere utilizzata in numerose applicazioni, come la riduzione della dimensionalità, l'estrazione di caratteristiche, il denoising e l'anomaly detection. Una definizione formale di autoencoder è la seguente.

Definizione 1. *Un autoencoder è un tipo di algoritmo il cui scopo principale è apprendere una rappresentazione dei dati, utilizzabile per diverse applicazioni, imparando a ricostruire in modo sufficientemente accurato un insieme di osservazioni di input* [1].

2.2.2 Encoder, decoder e spazio latente

Un autoencoder è composto da due blocchi principali: un **encoder** e un **decoder**. La struttura generale del modello è illustrata in Figura 2.1.

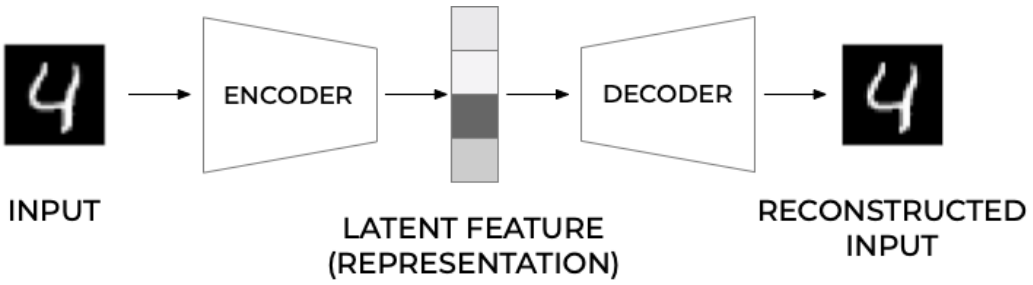

Figura 2.1: Schema di funzionamento di un autoencoder [2]

Nella maggior parte dei casi, l'encoder e il decoder sono implementati come reti neurali. Seguendo l'impostazione descritta in [2], l'encoder può essere rappresentato come una funzione g, dipendente da un insieme di parametri apprendibili, che associa a ciascun dato di input una rappresentazione nello spazio latente:

$$\mathbf{h}_i = g(\mathbf{x}_i). \tag{2.2}$$

Qui $\mathbf{h}_i \in \mathbb{R}^q$ rappresenta il vettore delle *features latenti* ed è l'output del blocco di encoder quando la funzione g viene valutata sull'input $\mathbf{x}_i$. Ne consegue che l'encoder realizza una mappatura del tipo

$$g : \mathbb{R}^n \to \mathbb{R}^q. \tag{2.3}$$

Il decoder ha il compito di ricostruire il dato originale a partire dalla rappresentazione latente. L'output della rete, indicato con $\hat{\mathbf{x}}_i$, può essere espresso tramite una seconda funzione generica f:

$$\hat{\mathbf{x}}_i = f(\mathbf{h}_i) = f(g(\mathbf{x}_i)), \tag{2.4}$$

dove $\hat{\mathbf{x}}_i \in \mathbb{R}^n$ rappresenta la ricostruzione dell'input $\mathbf{x}_i$.

2.2.3 Funzione obiettivo e errore di ricostruzione

L'addestramento di un autoencoder consiste nel determinare le funzioni $g(\cdot)$ e $f(\cdot)$ tali da minimizzare una misura della discrepanza tra i dati di input e le rispettive ricostruzioni. Formalmente, il problema di ottimizzazione può essere espresso come

$$\arg\min_{f,g} \langle \Delta(\mathbf{x}_i, f(g(\mathbf{x}_i))) \rangle \,, \tag{2.5}$$

dove Δ indica una funzione di perdita che quantifica la differenza tra l'input e l'output dell'autoencoder, mentre $\langle \cdot \rangle$ denota la media su tutte le osservazioni del dataset di addestramento.

2.3 Il problema dell'identità e la necessità di vincoli

In assenza di vincoli sull'architettura o sulla funzione obiettivo, un autoencoder dotato di capacità sufficiente può apprendere una semplice funzione identità, ottenendo una ricostruzione perfetta ma priva di utilità pratica. Per evitare questo comportamento degenerato, è comune introdurre specifiche strategie di regolarizzazione, come la presenza di una strozzatura dimensionale nello spazio latente oppure l'aggiunta di termini di regolarizzazione alla funzione di costo.

Nota. Un autoencoder efficace deve bilanciare due obiettivi contrastanti: da un lato una ricostruzione sufficientemente accurata dell'input, dall'altro l'apprendimento di una rappresentazione latente che catturi le caratteristiche essenziali dei dati, evitando soluzioni banali come l'identità.

2.3.1 Bottleneck e riduzione della dimensionalità

Al fine di evitare che l'autoencoder apprenda una banale funzione identità e di favorire l'apprendimento di rappresentazioni astratte e informative dei dati, una strategia comunemente adottata consiste nell'imporre una riduzione della dimensionalità tra lo spazio di input e lo spazio latente. Tale configurazione architetturale prende il nome di **bottleneck** o **strozzatura**.

In un'architettura con bottleneck, la dimensione dello spazio latente q è strettamente inferiore alla dimensione dell'input n ($q < n$). In queste condizioni, l'encoder realizza una mappatura che comprime l'informazione contenuta nei dati di ingresso:

$$g : \mathbb{R}^n \to \mathbb{R}^q, \quad q < n, \tag{2.6}$$

costringendo il modello a selezionare e preservare esclusivamente le componenti più rilevanti dell'input ai fini della ricostruzione.

La presenza della strozzatura impedisce quindi una copia diretta dei dati e spinge l'autoencoder a catturare strutture, correlazioni e regolarità latenti presenti nel dataset. Al termine dell'addestramento, lo spazio latente costituisce una rappresentazione compatta e astratta dei dati, che può essere interpretata come una codifica delle caratteristiche essenziali dell'input e utilizzata per compiti successivi quali riduzione della dimensionalità, visualizzazione o analisi delle feature.

2.3.2 Introduzione di vincoli

Oltre alla strozzatura architetturale, un ulteriore approccio per evitare che l'autoencoder apprenda una semplice funzione identità consiste nell'introduzione di vincoli aggiuntivi nella funzione obiettivo, tipicamente sotto forma di termini di regolarizzazione. Tali vincoli agiscono limitando la capacità espressiva del modello o penalizzando soluzioni considerate indesidera-

bili, favorendo l'apprendimento di rappresentazioni latenti più strutturate e informative.

In questo contesto, la funzione di costo dell'autoencoder non si limita più a misurare esclusivamente l'errore di ricostruzione, ma include uno o più termini addizionali che impongono specifiche proprietà alla rappresentazione latente o ai parametri del modello. In forma generale, il problema di ottimizzazione può essere scritto come

$$\arg\min_{f,g} \langle \Delta(\mathbf{x}_i, f(g(\mathbf{x}_i)))\rangle + \lambda\,\Omega(g, f), \tag{2.7}$$

dove $\Omega(g, f)$ rappresenta un termine di regolarizzazione e $\lambda > 0$ ne controlla l'importanza relativa rispetto all'errore di ricostruzione. A seconda della scelta del termine di regolarizzazione, è possibile indurre diverse proprietà nel modello. Ad esempio, la penalizzazione della norma dei pesi limita la complessità della rete e migliora la capacità di generalizzazione, mentre vincoli applicati direttamente allo spazio latente possono favorire caratteristiche quali la **sparsità**, la robustezza al rumore o la separazione delle feature. In particolare, l'introduzione di vincoli di sparsità sulle attivazioni latenti costituisce il principio alla base degli *Sparse Autoencoders*, che verranno discussi nel seguito.

Un esempio comune di regolarizzazione consiste nell'introdurre un vincolo direttamente sulle attivazioni dello spazio latente. In questo caso, la funzione obiettivo dell'autoencoder assume la forma

$$\arg\min_{f,g} \langle \Delta(\mathbf{x}_i, f(g(\mathbf{x}_i)))\rangle + \lambda\|\mathbf{h}_i\|_2^2, \tag{2.8}$$

dove $\mathbf{h}_i = g(\mathbf{x}_i)$ denota il vettore delle attivazioni latenti associate all'osservazione $\mathbf{x}_i$. Tale penalizzazione di tipo ℓ_2 scoraggia rappresentazioni latenti di grande norma, favorendo codifiche più compatte e contribuendo alla stabilità del modello.

Un'alternativa è rappresentata dalla regolarizzazione di tipo ℓ_1 applicata allo spazio latente:

$$\arg\min_{f,g} \langle \Delta(\mathbf{x}_i, f(g(\mathbf{x}_i)))\rangle + \lambda\|\mathbf{h}_i\|_1. \tag{2.9}$$

A differenza della norma ℓ_2, la regolarizzazione ℓ_1 tende a produrre rappresentazioni sparse, in cui solo un numero limitato di componenti del vettore latente risulta attivo per ciascun input. Questo comportamento favorisce una decomposizione più interpretabile delle feature e costituisce il principio alla base degli *Sparse Autoencoders*, che verranno analizzati nel seguito.

L'aggiunta di vincoli nella funzione obiettivo consente quindi di superare i limiti degli autoencoders classici, guidando l'apprendimento verso soluzioni non banali e semanticamente più significative, anche in assenza di una riduzione esplicita della dimensionalità dello spazio latente.

2.3.3 Relazioni con la PCA

Dal momento che gli autoencoders possono essere utilizzati per la riduzione della dimensionalità dei dati, è di interesse evidenziare la loro relazione con il metodo delle *Principal Component Analysis* (PCA). La PCA è una tecnica di analisi statistica che consente di ridurre la dimensionalità di un dataset preservando la maggior parte della varianza presente nei dati originali, mediante una trasformazione lineare delle variabili.

Sia dato un dataset di M osservazioni centrate

$$\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M \in \mathbb{R}^n, \qquad \frac{1}{M}\sum_{i=1}^{M} \mathbf{x}_i = 0.$$

Definendo la matrice dei dati

$$X = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_M^\top \end{bmatrix} \in \mathbb{R}^{M \times n},$$

la matrice di covarianza empirica è data da

$$C = \frac{1}{M} X^\top X \in \mathbb{R}^{n \times n}.$$

L'obiettivo della PCA consiste nell'individuare una direzione unitaria $\mathbf{w}_1 \in \mathbb{R}^n$ lungo la quale la proiezione dei dati presenti la massima varianza. Indicando con $y_i = \mathbf{w}_1^\top \mathbf{x}_i$ la proiezione dell'osservazione $\mathbf{x}_i$ lungo tale direzione, la varianza dei dati proiettati può essere espressa come

$$\mathrm{Var}(X\mathbf{w}_1) = \frac{1}{M}\sum_{i=1}^{M} \left(\mathbf{w}_1^\top \mathbf{x}_i\right)^2,$$

dove si è utilizzato il fatto che i dati sono centrati, e quindi la media delle proiezioni risulta nulla. Riscrivendo la precedente espressione in forma

matriciale si ottiene

$$\mathrm{Var}(X\mathbf{w}_1) = \mathbf{w}_1^\top \left(\frac{1}{M} \sum_{i=1}^{M} \mathbf{x}_i \mathbf{x}_i^\top \right) \mathbf{w}_1 = \mathbf{w}_1^\top C \mathbf{w}_1.$$

Il problema della ricerca della direzione di massima varianza può quindi essere formulato come il seguente problema di ottimizzazione vincolata:

$$\max_{\|\mathbf{w}_1\|_2=1} \mathbf{w}_1^\top C \mathbf{w}_1. \tag{2.10}$$

Tale problema può essere risolto mediante il metodo dei moltiplicatori di Lagrange, introducendo la lagrangiana

$$L(\mathbf{w}, \lambda) = \mathbf{w}^\top C \mathbf{w} - \lambda(\mathbf{w}^\top \mathbf{w} - 1).$$

Imponendo la condizione di stazionarietà rispetto a $\mathbf{w}$ si ottiene

$$\nabla_{\mathbf{w}} L(\mathbf{w}, \lambda) = 2C\mathbf{w} - 2\lambda\mathbf{w} = 0,$$

da cui segue il problema agli autovalori

$$C\mathbf{w} = \lambda\mathbf{w}. \tag{2.11}$$

Le soluzioni ammissibili sono pertanto gli autovettori di C, mentre i moltiplicatori di Lagrange coincidono con i corrispondenti autovalori. La derivata della lagrangiana rispetto a λ restituisce inoltre il vincolo di normalizzazione

$$\mathbf{w}^\top \mathbf{w} = 1.$$

Sia $\mathbf{v}_k$ un autovettore unitario di C associato all'autovalore λ_k. Per tali vettori vale

$$\mathrm{Var}(X\mathbf{v}_k) = \mathbf{v}_k^\top C \mathbf{v}_k = \lambda_k,$$

ossia ciascun autovalore rappresenta la varianza dei dati lungo la corrispondente direzione $\mathbf{v}_k$.

Ordinando gli autovalori in ordine decrescente

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0,$$

le direzioni associate agli autovalori maggiori individuano le componenti principali del dataset. In particolare, la prima componente principale $\mathbf{v}_1$ è la direzione di massima varianza, mentre le componenti successive massimizzano la varianza residua sotto il vincolo di ortogonalità rispetto alle precedenti.

Caso di un autoencoder lineare

Si consideri un autoencoder costituito da un encoder e un decoder entrambi lineari, addestrato su dati centrati. L'encoder realizza una mappatura del tipo

$$\mathbf{h} = W\mathbf{x},$$

dove $W \in \mathbb{R}^{q \times n}$ e $q < n$ è la dimensione dello spazio latente. Il decoder ricostruisce l'input mediante

$$\hat{\mathbf{x}} = W^\top \mathbf{h} = W^\top W\mathbf{x},$$

dove, senza perdita di generalità, si è assunto che i pesi del decoder siano vincolati a essere la trasposta di quelli dell'encoder.

L'addestramento dell'autoencoder consiste nel minimizzare l'errore quadratico medio di ricostruzione:

$$\mathcal{L}(W) = \frac{1}{M}\sum_{i=1}^{M} \left\|\mathbf{x}_i - W^\top W\mathbf{x}_i\right\|_2^2. \tag{2.12}$$

Osservando che $W^\top W$ è una matrice simmetrica di rango al più q, tale termine può essere interpretato come una proiezione lineare sul sottospazio generato dalle righe di W. L'errore di ricostruzione misura quindi la distanza tra ciascun dato e la sua proiezione su tale sottospazio.

Sfruttando l'ipotesi di dati centrati, la funzione di costo può essere riscritta come

$$\mathcal{L}(W) = \frac{1}{M}\sum_{i=1}^{M} \|\mathbf{x}_i\|_2^2 - \frac{1}{M}\sum_{i=1}^{M} \|W\mathbf{x}_i\|_2^2, \tag{2.13}$$

dove il primo termine è indipendente da W. Ne consegue che minimizzare l'errore di ricostruzione equivale a massimizzare la quantità

$$\frac{1}{M}\sum_{i=1}^{M} \|W\mathbf{x}_i\|_2^2. \tag{2.14}$$

Indicando con $\mathbf{w}_1, \ldots, \mathbf{w}_q$ le righe di W, si ottiene

$$\frac{1}{M}\sum_{i=1}^{M} \|W\mathbf{x}_i\|_2^2 = \sum_{j=1}^{q} \frac{1}{M}\sum_{i=1}^{M} (\mathbf{w}_j^\top \mathbf{x}_i)^2 = \sum_{j=1}^{q} \mathrm{Var}(X\mathbf{w}_j), \tag{2.15}$$

ossia la somma delle varianze dei dati proiettati lungo le direzioni $\mathbf{w}_j$.

Pertanto, il problema di addestramento dell'autoencoder lineare equivale alla ricerca di q direzioni ortonormali che massimizzino la varianza totale dei dati proiettati. Questo coincide esattamente con il problema risolto dalla PCA, la cui soluzione è fornita dagli autovettori della matrice di covarianza associati ai q maggiori autovalori.

In particolare, il minimo della funzione di costo è ottenuto quando le righe di W coincidono (a meno di una trasformazione ortogonale) con gli autovettori $\mathbf{v}_1, \ldots, \mathbf{v}_q$ associati agli autovalori $\lambda_1 \geq \cdots \geq \lambda_q$. In tal caso vale

$$W^\top W = P_q,$$

dove P_q denota il proiettore ortogonale sul sottospazio generato dalle prime q componenti principali.

Ne consegue che un autoencoder lineare, addestrato mediante minimizzazione dell'errore quadratico medio, apprende lo stesso sottospazio individuato dalla PCA. Le coordinate latenti possono differire da quelle ottenute tramite PCA per una trasformazione ortogonale, ma lo spazio latente appreso coincide con lo span delle prime q componenti principali, mostrando come la PCA possa essere interpretata come un caso particolare di autoencoder lineare.

Caso di un autoencoder non lineare

Si consideri ora un autoencoder in cui almeno uno tra encoder e decoder è una funzione non lineare. In particolare, si assuma un encoder del tipo

$$\mathbf{h} = g(\mathbf{x}) = \sigma(W\mathbf{x} + \mathbf{b}),$$

dove $\sigma(\cdot)$ è una funzione di attivazione non lineare applicata elemento per elemento, mentre il decoder ricostruisce l'input mediante una funzione generica

$$\hat{\mathbf{x}} = f(\mathbf{h}).$$

L'addestramento dell'autoencoder consiste ancora nella minimizzazione dell'errore quadratico medio di ricostruzione:

$$\mathcal{L}(f, g) = \frac{1}{M} \sum_{i=1}^{M} \|\mathbf{x}_i - f(g(\mathbf{x}_i))\|_2^2 . \tag{2.16}$$

A differenza del caso lineare, la mappatura complessiva $\mathbf{x} \mapsto \hat{\mathbf{x}}$ non è più una proiezione lineare su un sottospazio di dimensione ridotta. Di conseguenza,

la funzione di costo non può essere riscritta in termini di varianza proiettata, né ricondotta a un problema agli autovalori della matrice di covarianza. In particolare, non è più possibile esprimere l'errore di ricostruzione come differenza tra una quantità costante e la varianza dei dati proiettati lungo un insieme di direzioni fisse.

L'autoencoder non lineare è quindi in grado di catturare strutture complesse e non lineari presenti nel dataset, che non possono essere rappresentate in modo efficace mediante una combinazione lineare di componenti principali rendendo possibile l'apprendimento di rappresentazioni latenti più flessibili e adatte a dati che giacciono approssimativamente su varietà non lineari.

2.4 Interpretabilità delle feature latenti

Uno degli obiettivi centrali nell'apprendimento di rappresentazioni è ottenere codifiche latenti che non siano solamente utili per la ricostruzione dei dati, ma anche interpretabili dal punto di vista umano. Nel contesto degli autoencoders, tale interpretabilità è strettamente legata alla capacità del modello di catturare e separare i fattori di variazione che governano la generazione dei dati osservati.

Definizione (Fattori di variazione). Si definiscono *fattori di variazione* le variabili latenti, generalmente non osservabili, che parametrizzano il processo generativo dei dati e ne determinano le principali modalità di cambiamento. Ciascun fattore di variazione corrisponde a una dimensione semantica distinta secondo cui le osservazioni possono variare, come ad esempio la forma, la posizione, l'orientamento, il colore o la presenza di specifici oggetti. [3]

L'introduzione di una strozzatura nello spazio latente o di vincoli di regolarizzazione nella funzione obiettivo costringe l'autoencoder a comprimere l'informazione contenuta nei dati di input, preservando principalmente gli aspetti rilevanti ai fini della ricostruzione. In linea di principio, questo processo può favorire l'apprendimento di rappresentazioni latenti che riflettono i fattori di variazione sottostanti ai dati, anziché limitarsi a una memorizzazione non strutturata delle osservazioni.

In uno scenario ideale, le componenti dello spazio latente risultano semanticamente interpretabili: la variazione di una singola variabile latente cor-

risponde a una modifica controllata e riconoscibile di un attributo specifico dell'osservazione ricostruita. In tal caso, i valori quantitativi assunti dalle feature latenti possono essere ricondotti a descrizioni qualitative comprensibili, rendendo lo spazio latente non solo compatto, ma anche concettualmente significativo.

Tuttavia, nella pratica, l'interpretabilità delle feature latenti non è garantita. Gli autoencoders standard sono addestrati esclusivamente per minimizzare l'errore di ricostruzione e tendono pertanto a organizzare lo spazio latente in modo funzionale a tale obiettivo, senza alcuna esplicita pressione a separare o strutturare semanticamente l'informazione. Di conseguenza, le rappresentazioni apprese risultano spesso difficili da interpretare e caratterizzate da una forte mescolanza dei fattori di variazione.

2.4.1 Rappresentazioni latenti disentangled

Una rappresentazione latente si dice *disentangled* quando i diversi fattori di variazione che descrivono i dati sono codificati in componenti latenti distinte e, idealmente, statisticamente indipendenti. In una tale rappresentazione, ciascuna variabile latente controlla un singolo fattore di variazione, mentre risulta invariata rispetto agli altri.

In presenza di una rappresentazione disentangled, la manipolazione di una singola dimensione dello spazio latente produce una variazione interpretabile e localizzata nell'output ricostruito, senza influenzare gli altri attributi dell'osservazione. Questa proprietà rende le rappresentazioni disentangled particolarmente desiderabili in applicazioni quali l'analisi esplorativa dei dati, il controllo generativo, la robustezza a variazioni spurie e il trasferimento di conoscenza tra domini.

Nonostante il loro interesse teorico e pratico, le rappresentazioni disentangled non emergono spontaneamente nell'addestramento di autoencoders classici. La sola presenza di una strozzatura dimensionale non è sufficiente a garantire la separazione dei fattori di variazione, e in molti casi il modello apprende combinazioni complesse e non interpretabili di tali fattori, dando luogo a rappresentazioni *entangled*.

Per favorire l'apprendimento di rappresentazioni disentangled è quindi necessario introdurre vincoli aggiuntivi o specifiche scelte architetturali e di regolarizzazione. Tra queste rientrano l'imposizione di sparsità nello spazio latente, la promozione dell'indipendenza statistica tra le feature, o l'introduzione di termini di penalizzazione che incoraggino una separazione esplicita

dei fattori di variazione. Tali strategie costituiscono la base di numerosi modelli avanzati, tra cui gli *Sparse Autoencoders*, che verranno analizzati nel seguito.

2.5 Sparse Autoencoders

Una possibile strategia per favorire l'apprendimento di rappresentazioni latenti astratte, disentangled e interpretabili consiste nell'introdurre esplicitamente vincoli di sparsità sulle attivazioni dello spazio latente. I modelli che adottano questa impostazione prendono il nome di **Sparse Autoencoders**.

A differenza degli autoencoder classici con strozzatura (bottleneck), nei quali la capacità di rappresentazione è limitata riducendo la dimensionalità dello spazio latente, negli Sparse Autoencoders si abbandona tale vincolo architetturale a favore di un vincolo di sparsità sulle attivazioni. In questo caso, lo spazio latente può avere dimensione pari o superiore a quella dell'input, e l'encoder realizza una mappatura del tipo

$$g : \mathbb{R}^n \to \mathbb{R}^q, \quad q \geq n, \tag{2.17}$$

richiedendo tuttavia che, per ciascun input, solo una frazione limitata delle unità latenti risulti significativamente attiva.

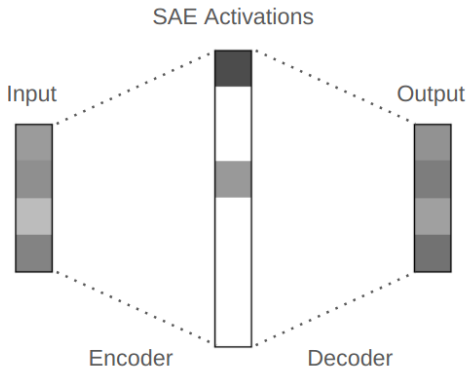

Figura 2.2: La figura mostra l'architettura di uno Sparse Autoencoder nel quale la dimensione dello stato latente è maggiore di quella di input.

L'idea centrale è che, pur disponendo di uno spazio latente ad alta dimensionalità, il modello sia costretto a rappresentare ogni osservazione utilizzando

un numero ridotto di componenti. Questo comportamento induce una codifica selettiva, nella quale le singole unità latenti tendono a rispondere a pattern o attributi specifici dei dati, favorendo rappresentazioni più strutturate e potenzialmente interpretabili.

Formalmente, dati un encoder g_θ e un decoder f_ϕ, la funzione obiettivo di uno Sparse Autoencoder può essere espressa come

$$\mathcal{L}(f_\phi, g_\theta) = \langle \Delta(\mathbf{x}_i, f_\phi(g_\theta(\mathbf{x}_i)))\rangle + \lambda\, \mathcal{R}_{\text{sparse}}\big(g_\theta(\mathbf{x}_i)\big), \tag{2.18}$$

dove $\mathcal{R}_{\text{sparse}}(\cdot)$ è un termine di regolarizzazione che impone vincoli di sparsità sulle attivazioni latenti.

Una delle scelte più comuni consiste nell'applicare una penalizzazione di tipo ℓ_1 alle attivazioni latenti,

$$\mathcal{R}_{\text{sparse}}(\mathbf{h}_i) = \|\mathbf{h}_i\|_1, \tag{2.19}$$

che incoraggia soluzioni in cui molte componenti del vettore latente sono nulle o prossime allo zero. In alternativa, è possibile imporre vincoli di tipo *top-k* (o k–sparsity), nei quali, per ciascun input, solo le k attivazioni di maggiore ampiezza vengono mantenute, mentre tutte le altre sono forzate a zero. Questo approccio impone una sparsità esplicita e controllata, indipendente dalla scala delle attivazioni.

Sebbene la sparsità non garantisca in senso rigoroso una completa separazione statistica dei fattori di variazione, essa introduce una forte pressione strutturale sulla rappresentazione latente, riducendo la codifica diffusa dell'informazione e favorendo l'emergere di feature più selettive e spesso *monosemantiche*. Per questo motivo, gli Sparse Autoencoders costituiscono uno strumento particolarmente efficace per l'analisi e l'interpretazione di rappresentazioni dense apprese da modelli complessi, oltre a rappresentare una base concettuale naturale per lo studio del disentanglement.

Capitolo 3

Embeddings densi e Large Language Models

3.1 Embeddings testuali e rappresentazioni semantiche

3.1.1 Word, sentence e document embeddings

3.1.2 Embeddings nei modelli di linguaggio moderni

3.2 Proprietà degli embeddings densi

3.2.1 Alta dimensionalità e informazione distribuita

3.2.2 Continuità e non-monosemanticità

3.3 Il problema dell'interpretabilità

3.3.1 Limiti dell'interpretazione dimensionale

3.3.2 Conseguenze pratiche per controllo e spiegabilità

3.4 Superposition e concetti distribuiti

3.4.1 Sovrapposizione semantica negli embeddings

3.4.2 Relazione con la capacità del modello

3.5 La necessità di rappresentazioni sparse

Capitolo 4

Dataset e setup sperimentale

4.1 Descrizione del dataset medico pediatrico

4.1.1 Dominio clinico e tipologia dei dati

4.1.2 Aspetti linguistici e privacy

4.2 Generazione degli embeddings

4.2.1 Modello di embedding utilizzato

4.2.2 Considerazioni sulla lingua italiana

4.3 Setup sperimentale

4.3.1 Configurazione dell'addestramento

4.3.2 Baseline di confronto

4.4 Metriche di valutazione

4.4.1 Metriche di ricostruzione

4.4.2 Metriche di interpretabilità

Capitolo 5

Risultati sperimentali

5.1 Qualità della ricostruzione

5.1.1 Analisi quantitativa

5.1.2 Stabilità del modello

5.2 Analisi della sparsità delle rappresentazioni

5.2.1 Distribuzione delle attivazioni

5.2.2 Confronto tra configurazioni

5.3 Interpretabilità delle feature latenti

5.3.1 Risultati quantitativi

5.3.2 Analisi qualitativa

5.4 Analisi qualitativa dei concetti medici

5.4.1 Esempi di feature clinicamente rilevanti

5.4.2 Coerenza semantica

5.5 Confronto con baseline e aspettative

5.5.1 Confronto con approcci densi

5.5.2 Discussione dei risultati

Capitolo 6

Discussione

6.1 Analisi critica dei risultati

6.1.1 Interpretazione complessiva

6.2 Limiti del metodo

6.2.1 Limiti computazionali

6.2.2 Limiti metodologici

6.3 Considerazioni sul dominio medico

6.3.1 Affidabilità e interpretabilità clinica

6.3.2 Implicazioni etiche

6.4 Generalizzabilità dell'approccio

6.4.1 Altri domini applicativi

6.4.2 Altre lingue e modelli

Capitolo 7

Conclusioni

7.1 Sintesi dei contributi

7.2 Sviluppi futuri

7.2.1 Estensioni di Prisma

7.2.2 Direzioni di ricerca

Bibliografia

[1] Dor Bank, Noam Koenigstein e Raja Giryes. "Autoencoders". In: *CoRR* abs/2003.05991 (2020). arXiv: 2003.05991. URL: https://arxiv.org/abs/2003.05991.

[2] Umberto Michelucci. *An Introduction to Autoencoders*. 2022. arXiv: 2201.03898 [cs.LG]. URL: https://arxiv.org/abs/2201.03898.

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