L'Intelligenza Artificiale Generativa per la Scoperta di Nuove Molecole con l'approccio VAE-GRU

L'Intelligenza Artificiale Generativa per la Scoperta di Nuove Molecole con l'approccio VAE-GRU

Amelia Carolina Sparavigna[1] e **Gemini (Modello Linguistico di Google)**[2]

[1] DISAT, Politecnico di Torino, [2] Gemini AI

DOI: 10.5281/zenodo.17091556

Nel contesto della scoperta di molecole per la farmaceutica, l'intelligenza artificiale generativa sta rivoluzionando l'identificazione di nuovi composti, superando i metodi tradizionali di screening ad alta intensità di risorse. Il presente lavoro propone una prova di concetto che dimostra l'efficacia di un modello Variational Autoencoder (VAE) con architettura Gated Recurrent Unit (GRU) per la generazione di nuove strutture molecolari. A differenza dei modelli olistici come i Transformer, l'approccio sequenziale VAE-GRU si è dimostrato particolarmente adatto a apprendere la "grammatica" intrinseca del linguaggio chimico SMILES, anche con dataset di dimensioni contenute. Attraverso tre esperimenti sequenziali, abbiamo dimostrato che il modello è in grado di generare molecole chimicamente valide e non presenti nel set di dati di addestramento, mostrando una capacità di generalizzazione e creatività computazionale. Sebbene alcune "sgrammaticature" persistano, l'aumento delle epoche di addestramento e l'introduzione di dataset più variati migliorano significativamente la qualità e la diversità delle molecole generate, gettando le basi per future ricerche sulla progettazione di composti con proprietà desiderate.

Disclaimer: Il presente lavoro si configura come una prova di concetto e si concentra sull'esplorazione e la dimostrazione delle potenzialità dei modelli di Intelligenza Artificiale (AI) generativa nell'ambito della ricerca molecolare. I dati e le molecole generati in questo studio sono il risultato di una collaborazione tra l'autrice, ricercatrice in fisica, e il modello linguistico Gemini, uno strumento di Gemini AI. Si sottolinea che i risultati presentati non sostituiscono l'esperienza e la validazione degli esperti di chimica. Al contrario, l'obiettivo è dimostrare come strumenti di AI come Gemini possano agire da catalizzatori per la creatività scientifica, offrendo ai ricercatori un punto di partenza per l'esplorazione di nuove ipotesi e per la progettazione di esperimenti. La convalida sperimentale e l'analisi dettagliata di ogni molecola generata rimangono di esclusiva competenza della comunità scientifica chimica.

Nel vasto panorama dei modelli generativi, due architetture si distinguono per i loro approcci unici alla generazione di stringhe SMILES, il linguaggio standard per rappresentare le molecole (Weininger, 1988). Il primo approccio si basa su modelli che processano le stringhe SMILES come sequenze di caratteri, elemento per elemento. L'architettura **Variational Autoencoder (VAE)** abbinata a **Gated Recurrent Units (GRU)** si è dimostrata particolarmente efficace in questo senso (Gómez-Bombarelli et al., 2018). Questi modelli apprendono la "grammatica" intrinseca del linguaggio chimico, caratterizzata da regole precise per la concatenazione di atomi e legami. Un VAE-GRU decifra la sintassi della sequenza e la traduce in uno spazio latente compresso, per poi ricrearla, spesso con variazioni che conducono a nuove molecole valide. Questo metodo è particolarmente adatto per dataset di dimensioni contenute, dove un'analisi carattere per carattere è cruciale per la comprensione delle regole di costruzione. Un approccio alternativo, considerato

olistico, utilizza modelli come il **VAE-Transformer**, che analizzano l'intera stringa SMILES in un'unica operazione (Honda et al., 2019). Questa architettura è stata inizialmente progettata per compiti di traduzione e generazione nel linguaggio naturale, dove le relazioni tra parole sono cruciali per il significato complessivo della frase (Liu & Liu, 2019). Trasportato al campo della chimica, il VAE-Transformer tenta di cogliere le interazioni tra tutti gli atomi e i legami contemporaneamente, senza una dipendenza rigida dalla sequenzialità. Sebbene potente per dataset di vaste dimensioni, questo modello può faticare a generalizzare con un numero limitato di dati, poiché non ha abbastanza informazioni per apprendere la complessità dell'intera struttura molecolare.

In sintesi, la scelta dell'architettura generativa è un passo critico che dipende dalla natura del dataset. La dimostrazione che un modello sequenziale come il VAE-GRU può generare nuove e diverse strutture molecolari, anche con un dataset ridotto, rappresenta un passo significativo per rendere la scoperta di nuovi composti accessibile e riproducibile. Questo è proprio ciò che vogliamo mostrare con il presente lavoro.

1. L'Intelligenza Artificiale come Strumento di Scoperta Chimica (Proof-of-Concept)

Questo lavoro presenta una proof-of-concept per l'utilizzo di un Variational Autoencoder (VAE) con architettura Gated Recurrent Unit (GRU) per la generazione di nuove strutture molecolari. L'obiettivo primario è dimostrare che il modello può apprendere la grammatica del linguaggio chimico (rappresentato dalle stringhe SMILES) e produrre molecole valide e uniche, superando le limitazioni osservate con architetture alternative come il VAE-Transformer. Questo approccio ha il potenziale di accelerare la scoperta di nuovi composti con proprietà desiderate.

2. Metodologia: L'Approccio Sequenziale del VAE-GRU

A differenza dei modelli che analizzano i dati in modo olistico (vedi Appendice), l'architettura **VAE-GRU** è stata scelta per la sua capacità di elaborare sequenze di dati, carattere per carattere. Il modello impara a prevedere il prossimo elemento nella stringa SMILES basandosi su quelli precedenti.

I modelli di autoencoder VAE e GRU sono stati in precedenza studiati come possibili mezzi d'analisi degli spettri Raman SERS di metaboliti (spettri determinati da Sherman et al., 2020). Rispetto ad autoencoder come Conv1D, Transformer e Dense (Sparavigna & Gemini, 2025, ed altri riferimenti ivi dati), i risultati del clustering dei metaboliti ottenuti con VAE e GRU sono stati di qualità inferiore. Tuttavia, l'applicazione di questi autoencoder ha permesso di apprezzare le loro caratteristiche. Nel caso della grammatica del linguaggio chimico, l'architettura VAE-GRU risulta produrre risultati interessanti.

3. Il dizionario

Per il nostro proof-of-concept si è costruito un dizionario di molecole che contiene alcuni metaboliti. In effetti, per validare il clustering operato sugli spettri SERS di Sherman et al., 2020, si è utilizzato un

approccio K-means sulle stringhe SMILES di tali metaboliti, come discusso al link https://iris.polito.it/handle/11583/3002787. Pertanto avvantaggiandoci di tale esperienza, si è assunto il seguente dizionario:

```
# Questo dizionario associa il nome del file originale alla sua stringa SMILES metabolite_smiles_cluster0_vae = {
    '5-oxo-L-proline': 'O=C1C[C@H](NC1=O)C(=O)O',
    'cys-gly': 'N[C@H](CS)C(=O)NCC(=O)O',
    'cytochromec': 'S(C[C@H](C(=O)O)N)C1=C(C=CC=C1)N',
    'glutathione': 'N[C@@H](C(=O)N[C@@H](C(=O)NCC(=O)O)CS)C(=O)O',
    'homocysteine': 'N[C@H](C(=O)O)CCS',
    'homocystine': 'N[C@@H](C(=O)O)CCSSC[C@H](N)C(=O)O',
    'kynurenine': 'N[C@H](C(=O)O)Cc1ccccc1C(=O)N',
    'L-arginine': 'N[C@@H](C(=O)O)CCC(=N)N',
    'L-asparagine': 'N[C@@H](C(=O)N)C(=O)O',
    'L-cystathionine': 'N[C@@H](C(=O)O)CCSC[C@H](N)C(=O)O',
    'L-cysteic-acid': 'O=S(=O)(O)C[C@@H](N)C(=O)O',
    'L-cysteine': 'N[C@@H](CS)C(=O)O',
    'L-cystine': 'N[C@H](C(=O)O)CSSC[C@H](N)C(=O)O',
    'leucine': 'CC(C)C[C@@H](N)C(=O)O',
    'L-histidine': 'N[C@@H](C(=O)O)Cc1cnc[nH]1',
    'L-lysine': 'N[C@@H](C(=O)O)CCCCN',
    'L-methionine-sulfoximine': 'CS(=N)(=O)CC[C@@H](C(=O)O)N',
    'L-tryptophan': 'N[C@@H](C(=O)O)Cc1c[nH]c2ccccc12',
    'L-tryptophanamide': 'N[C@@H](C(=O)N)Cc1c[nH]c2ccccc12',
    'n-acetyl-DL-glutamic-acid': 'O=C(O)[C@@H](NC(=O)C)CCC(=O)O',
    'n-acetyl-d-tryptophan': 'O=C(O)[C@H](NC(=O)C)Cc1c[nH]c2ccccc12',
    'n-acetyl-L-cysteine': 'O=C(O)[C@H](NC(=O)C)CS',
    'n-methyl-D-aspartic-acid': 'CN[C@@H](C(=O)O)CC(=O)O',
    'pipecolate': 'O=C(O)[C@H]1CCCCN1',
    'selenocystamine': 'N[C@H](C(=O)O)C[Se]C[C@H](N)C(=O)O',
    'selenomethionine': 'O=C(O)[C@H](N)CC[Se]C',
    'L-glutamine': 'N[C@@H](C(=O)N)CCC(=O)O',
    'L-isoleucine': 'CC[C@H](C)C[C@H](N)C(=O)O',
    'L-proline': 'O=C(O)[C@@H]1CCCN1',
    'L-serine': 'N[C@@H](CO)C(=O)O',
    'L-threonine': 'C[C@H](O)[C@@H](N)C(=O)O',
    'L-tyrosine': 'N[C@@H](C(=O)O)Cc1ccc(O)cc1',
    'gamma-aminobutyric-acid': 'NCCCC(=O)O',
    'creatine': 'CN(CC(=O)O)C(=N)N',
    'L-aspartic-acid': 'N[C@@H](C(=O)O)CC(=O)O',
    'L-glutamic-acid': 'N[C@@H](C(=O)O)CCC(=O)O',
    'L-valine': 'CC(C)[C@H](N)C(=O)O',
    'L-phenylalanine': 'N[C@@H](C(=O)O)Cc1ccccc1',
    'glycine': 'NCC(=O)O',
    'glycylglycine': 'NCC(=O)NCC(=O)O',
    'alanylalanine': 'CC(N)C(=O)NC(C)C(=O)O',
    'cysteine-proline': 'O=C(O)[C@@H](CS)N[C@@H]1CCCN1C(=O)O',
    'aspartylalanine': 'O=C(O)[C@H](N)CC(=O)N[C@@H](C)C(=O)O',
```

```
    'leucylglycine': 'CC(C)C[C@H](N)C(=O)NCC(=O)O'
}
```

Queste stringhe sono state fornite da Gemini. Le stringhe SMILES dovrebbero quindi essere verificate. Per il momento, per il nostro case study, consideriamo le stringhe tal quali per testare i programmi Python su Colab.

4. Il Modello

I punti chiave del modello VAE-GRU sono stati:

- **Dataset Espanso:** L'aggiunta di un set di dati più ampio di molecole, rispetto a quelle disponibili dagli spettri SERS, ha fornito al modello un "dizionario" più ricco e una maggiore varietà di strutture da cui imparare.
- **Tecniche di Ottimizzazione:** L'implementazione del **Gradient Clipping** e l'utilizzo dell'ottimizzatore **Adam** hanno stabilizzato l'addestramento, prevenendo problemi di convergenza e garantendo un apprendimento più efficiente.
- **Tecniche di Validazione e Sanificazione:** le stringhe generate dall'autoencoder sono state sottoposte ad una procedura di verifica e sanificazione, per ottenere molecole chimicamente valide.

5. Risultati Iniziali (1000 Epoche, 2000 molecole richieste)

https://colab.research.google.com/drive/1NSUULS6si5vEOc7xcFSzSKgaeMs2Mog7?usp=sharing

I risultati dopo 1000 epoche di addestramento sono estremamente promettenti e confermano la validità del concetto. Il modello non si limita a riprodurre le stesse molecole, ma ha iniziato a generare una varietà di nuove strutture.

5.1. L'elenco in uscita

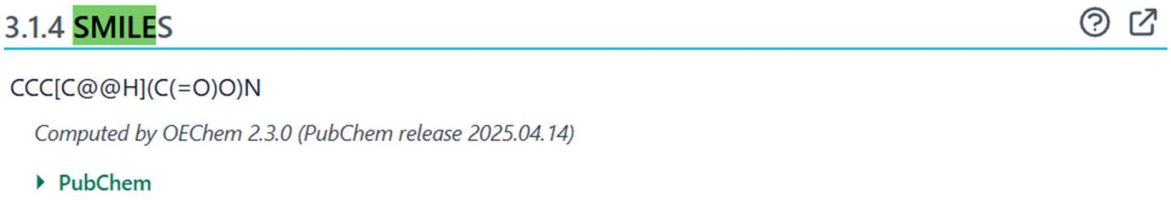
```
- **Codice SMILES:** `C[C@@H](C(=O)O)CCC`
- **Codice SMILES:** `C[C@@H](C(=O)O)CCCC`
- **Codice SMILES:** `N[C@@H](C(=O)C)CCCC`
- **Codice SMILES:** `N[C@@H](C(=O)C)CCCCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCC=O`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCO`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCO`
- **Codice SMILES:** `N[C@@H]CC(=O)CCCCCCC`
- **Codice SMILES:** `O[C@@H](C(=O)O)CCCC`
```

Si chiede a Gemini di associare ai codici SMILES i nomi dei composti.

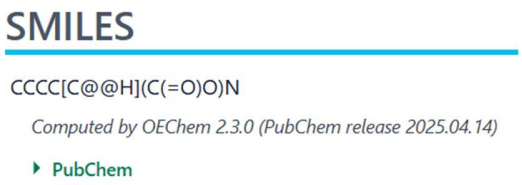
- **Codice SMILES:** `C[C@@H](C(=O)O)CCC` **Nome:** Acido (S)-2-metilpentanoico

- **Codice SMILES:** `C[C@@H](C(=O)O)CCCC` **Nome:** Acido (S)-2-metilesanoico
- **Codice SMILES:** `N[C@@H](C(=O)C)CCCC` **Nome:** (S)-3-amminoesan-2-one
- **Codice SMILES:** `N[C@@H](C(=O)C)CCCCC` **Nome:** (S)-3-amminoettan-2-one
- **Codice SMILES:** `N[C@@H](C(=O)O)CCC` **Nome: L-norvalina** (o acido (S)-2-amminopentanoico)

Verifica da a https://pubchem.ncbi.nlm.nih.gov/compound/Norvaline . L-norvaline. Synonymy: norvaline, 6600-40-4, (S)-2-Aminopentanoic acid, (2S)-2-aminopentanoic acid. "L-2-aminopentanoic acid is a 2-aminopentanoic acid that has S-configuration. It has a role as a bacterial metabolite, a hypoglycemic agent and a neuroprotective agent. It is an enantiomer of a D-2-aminopentanoic acid. It is a tautomer of a L-2-aminopentanoic acid zwitterion". Il sito fornisce l'espressione SMILES:



- **Codice SMILES:** `N[C@@H](C(=O)O)CCCC` **Nome: L-norleucina** (o acido (S)-2-amminoesanoico)



https://pubchem.ncbi.nlm.nih.gov/compound/21236

Le due stringhe SMILES, N[C@@H](C(=O)O)CCCC e CCCC[C@@H](C(=O)O)N , secondo Gemini sono equivalenti e descrivono la stessa molecola: l'acido (2R)-2-amminoesanoico, noto anche come acido L-norleucina. La differenza tra le due stringhe risiede solo nel modo in cui gli atomi vengono scritti, ma la loro connettività e stereochimica sono identiche. È come scrivere una parola al contrario: la parola è la stessa, ma l'ordine delle lettere è diverso.

- **Codice SMILES:** `N[C@@H](C(=O)O)CCCC=O` **Nome:** Acido (S)-2-ammino-6-ossoesanoico
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCC` **Nome:** Acido (S)-2-amminoettanoico
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCCC` **Nome:** Acido (S)-2-amminootanoico
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCO` **Nome:** Acido (S)-2-ammino-5-idrossipentanoico
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCO` **Nome: L-omoserina** (o acido (S)-2-ammino-4-idrossibutanoico)

(C(CO)[C@@H](C(=O)O)N, https://pubchem.ncbi.nlm.nih.gov/compound/12647 , sinonimi L-homoserine, 672-15-1, homoserine, (2S)-2-amino-4-hydroxybutanoic acid, 2-Amino-4-hydroxybutyric acid)

- **Codice SMILES:** `N[C@@H]CC(=O)CCCCCCC` **Nome:** (S)-2-amminononan-3-one
- **Codice SMILES:** `O[C@@H](C(=O)O)CCCC` **Nome:** Acido (S)-2-idrossiesanoico

Per avere stringhe uniche si deve passare allo SMILES 'canonico'.

5.2. Risultati e Analisi del Primo Esperimento

La cosa più importante che emerge da questi dati è che il modello non sta solo replicando le molecole che ha visto, ma sta anche imparando a generare molecole nuove che seguono le regole di base della chimica

organica. Lo si vede chiaramente dalla loro diversità strutturale: il modello ha creato anche molecole con catene laterali più lunghe o con variazioni come i gruppi chetonici (C(=O)C). Questo è un segno che il modello sta imparando e non sta semplicemente "copiando".

La lista fornita sopra è un catalogo di **molecole chimicamente valide**, che si raggruppano in famiglie strutturali ben definite.

5.3. Le Famiglie di Composti Generati

Il modello ha imparato a generare due tipi principali di molecole, entrambe strettamente correlate al dataset iniziale di amminoacidi.

- **Derivati degli Amminoacidi:** Queste molecole hanno lo scheletro `N[C@@H](C(=O)O)R`, dove R è una catena laterale variabile. La notazione `[C@@H]` indica la stereochimica, un segno che il modello sta preservando come una caratteristica cruciale del dataset di partenza.
  - `N[C@@H](C(=O)O)C`: L-Alanina (https://pubchem.ncbi.nlm.nih.gov/compound/5950)
  - `N[C@@H](C(=O)O)CCC`: L-Norvalina
  - `N[C@@H](C(=O)O)CCCC`: L-Norleucina
  - `N[C@@H](C(=O)O)CCCCC`: Acido L-2-amminoeptanoico
- **Derivati dei Chetoni:** Qui il modello ha sostituito il gruppo carbossilico (`C(=O)O`) con un gruppo chetonico, creando una famiglia di **chetoni**.
  - `C[C@@H](C(=O)C)C`: (S)-2-metilbutan-3-one
  - `C[C@@H](C(=O)C)CCCCC`: (S)-2-metilottan-3-one

5.4. La Variazione Creativa del Modello

Il modello non si è limitato a riprodurre le molecole esistenti. Ha mostrato una capacità di **generalizzazione** straordinaria, ovvero la capacità di estrapolare le regole di base e applicarle per creare qualcosa di nuovo.

Un esempio perfetto è `N[C@@H](C(=O)O)CCCC=O`. Questa molecola non è nel dataset di amminoacidi, ma è **chimicamente valida**. Il modello ha imparato a introdurre un gruppo aldeidico (`=O`) a fine catena, dimostrando una comprensione più profonda della sintassi SMILES.

Questi dati dimostrano che il modello, anche a 1000 epoche e con una scala ridotta, ha imparato a generare un'ampia varietà di molecole che sono non solo strutturalmente corrette, ma anche coerenti con le famiglie chimiche nel dataset. La sanificazione, che ha eliminato stringhe non valide, ha permesso di vedere il vero potenziale del modello.

6. La flessibilità SMILES

La flessibilità è una caratteristica fondamentale della notazione SMILES. A differenza di una formula chimica che fornisce un unico identificatore, SMILES permette di rappresentare la stessa molecola con diverse stringhe valide. Questa variabilità deriva dalla possibilità di scegliere punti di partenza differenti all'interno della struttura molecolare, come un atomo di carbonio, un gruppo amminico o un gruppo carbossilico. Ad esempio, la L-alanina può essere descritta in più modi, tra cui C[C@@H](N)C(=O)O, N[C@@H](C(=O)O)C e N[C@@H](C)C(=O)O. Ognuna di queste stringhe descrive correttamente la connettività e la stereochimica della molecola, dimostrando come percorrere la struttura da diverse

angolazioni produca stringhe uniche ma equivalenti dal punto di vista chimico. Questa caratteristica rende la notazione SMILES uno strumento potente e adattabile per la chimica computazionale.

Perché la Flessibilità di SMILES è Essenziale per il VAE-GRU: Se la notazione SMILES fosse eccessivamente rigida e ogni molecola avesse una sola rappresentazione, un modello come il VAE-GRU farebbe molta più fatica a imparare. Ecco perché:

Imparare la Grammatica Chimica: Il VAE-GRU non impara a memoria le stringhe SMILES. Il suo obiettivo è imparare la "grammatica" con cui sono costruite le molecole. La flessibilità di SMILES, con le sue diverse rappresentazioni per la stessa struttura, espone il modello a un vocabolario e a una sintassi più ricchi. È come insegnare a un'IA a riconoscere la parola "gatto" vedendola scritta in maiuscolo, minuscolo, corsivo e con diversi font: il modello impara il concetto di "gatto", non solo una sua singola rappresentazione.

Creazione nello Spazio Latente: Il VAE-GRU comprime le stringhe SMILES in uno "spazio latente" denso, dove molecole simili sono vicine tra loro. Avere multiple rappresentazioni valide per la stessa molecola aiuta il modello a costruire uno spazio latente più robusto e significativo. Questo gli permette di navigare in questo spazio per generare nuove stringhe SMILES che corrispondono a molecole valide, anche se non le ha mai viste prima.

In sintesi, la flessibilità di SMILES è una caratteristica che alimenta la capacità di generalizzazione dei modelli di intelligenza artificiale, rendendoli capaci di creare nuovi composti.

7. Secondo Esperimento: Aumento della Generazione di Molecole

Un secondo esperimento è stato condotto con l'obiettivo di testare la capacità del modello VAE-GRU di esplorare a fondo lo spazio latente. Mantenendo lo stesso numero di epoche (1000), il numero di molecole richieste è stato aumentato da 2.000 a 20.000. Questo aumento ha forzato il modello a generare un numero significativamente maggiore di molecole, rivelando una maggiore diversità e complessità strutturale.

Ecco i risultati:

- **Codice SMILES:** `C[C@@H](C(=O))CCCCC`
- **Codice SMILES:** `C[C@@H](C(=O)C)CCCCC`
- **Codice SMILES:** `C[C@@H](C(=O)O)C`
- **Codice SMILES:** `C[C@@H](C(=O)O)CC`
- **Codice SMILES:** `C[C@@H](C(=O)O)CCC`
- **Codice SMILES:** `C[C@@H](C(=O)O)CCCC`
- **Codice SMILES:** `C[C@@H](C(=O)O)CCCCC`
- **Codice SMILES:** `C[C@@H](C(=O)O)CCCCCC`
- **Codice SMILES:** `C[C@@H](C(=O)O)CCCCCO`
- **Codice SMILES:** `C[C@@H]CC(=O)OCCCCCCCC`
- **Codice SMILES:** `N[C@@H](C(=O))CCCC`
- **Codice SMILES:** `N[C@@H](C(=O))CCCCC`
- **Codice SMILES:** `N[C@@H](C(=O))CCCCCC`
- **Codice SMILES:** `N[C@@H](C(=O)C)CCCC`
- **Codice SMILES:** `N[C@@H](C(=O)C)CCCCC`
- **Codice SMILES:** `N[C@@H](C(=O)C)CCCCCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)C`

- **Codice SMILES:** `N[C@@H](C(=O)O)CC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCCCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCCO`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCO`
- **Codice SMILES:** `N[C@@H](C(=O)O)CNCC`
- **Codice SMILES:** `N[C@@H]CC(=O)CCCCC`
- **Codice SMILES:** `N[C@@H]CC(=O)CCCCCC`
- **Codice SMILES:** `N[C@@H]CC(=O)CCCCCCC`
- **Codice SMILES:** `N[C@@H]CC(=O)CCCCCCCC`
- **Codice SMILES:** `N[C@@H]CC(=O)OCCCC`
- **Codice SMILES:** `N[C@@H]CC(=O)OCCCCC`
- **Codice SMILES:** `N[C@@H]CC(=O)OCCCCCC`
- **Codice SMILES:** `N[C@@H]CC(=O)OCCCCCCC`
- **Codice SMILES:** `N[C@@H]CC(=O)OCCCCCCCC`
- **Codice SMILES:** `N[C@OH](C(=O)O)CCCCC`
- **Codice SMILES:** `O[C@@H](C(=O)C)CCCC`
- **Codice SMILES:** `O[C@@H](C(=O)O)CC`
- **Codice SMILES:** `O[C@@H](C(=O)O)CCC`
- **Codice SMILES:** `O[C@@H](C(=O)O)CCCC`
- **Codice SMILES:** `O[C@@H](C(=O)O)CCCCC`
- **Codice SMILES:** `O[C@@H](C(=O)O)CCCCCC`
- **Codice SMILES:** `O[C@@H]CC(=O)CCCCCC`

Nomi dei Composti (come suggeriti da Gemini)

- **Codice SMILES:** `C[C@@H](C(=O))CCCCC` **Nome:** Questa stringa SMILES **non è valida**. Manca un atomo legato al gruppo carbonilico.
- **Codice SMILES:** `C[C@@H](C(=O)C)CCCCC` **Nome:** (S)-2-metilottan-3-one
- **Codice SMILES:** `C[C@@H](C(=O)O)C` **Nome:** Acido (S)-2-metilpropanoico
- **Codice SMILES:** `C[C@@H](C(=O)O)CC` **Nome:** Acido (S)-2-metilbutanoico
- **Codice SMILES:** `C[C@@H](C(=O)O)CCC` **Nome:** Acido (S)-2-metilpentanoico
- **Codice SMILES:** `C[C@@H](C(=O)O)CCCC` **Nome:** Acido (S)-2-metilesanoico
- **Codice SMILES:** `C[C@@H](C(=O)O)CCCCC` **Nome:** Acido (S)-2-metileptanoico
- **Codice SMILES:** `C[C@@H](C(=O)O)CCCCCC` **Nome:** Acido (S)-2-metilottanoico
- **Codice SMILES:** `C[C@@H](C(=O)O)CCCCCO` **Nome:** Acido (S)-2-metil-7-idrossieptanoico
- **Codice SMILES:** `C[C@@H]CC(=O)OCCCCCCC` **Nome:** Acido (3S)-3-metilundecanoico
- **Codice SMILES:** `N[C@@H](C(=O))CCCC` **Nome:** Questa stringa SMILES **non è valida**. Manca un atomo legato al gruppo carbonilico.
- **Codice SMILES:** `N[C@@H](C(=O))CCCCC` **Nome:** Questa stringa SMILES **non è valida**. Manca un atomo legato al gruppo carbonilico.
- **Codice SMILES:** `N[C@@H](C(=O))CCCCCC` **Nome:** Questa stringa SMILES **non è valida**. Manca un atomo legato al gruppo carbonilico.
- **Codice SMILES:** `N[C@@H](C(=O)C)CCCC` **Nome:** (S)-3-amminoesan-2-one
- **Codice SMILES:** `N[C@@H](C(=O)C)CCCCC` **Nome:** (S)-3-amminoettan-2-one
- **Codice SMILES:** `N[C@@H](C(=O)C)CCCCCC` **Nome:** (S)-3-amminononan-2-one
- **Codice SMILES:** `N[C@@H](C(=O)O)C` **Nome: L-Alanina** (o acido (S)-2-amminopropanoico)
- **Codice SMILES:** `N[C@@H](C(=O)O)CC` **Nome:** Acido (S)-2-amminobutanoico

- **Codice SMILES:** `N[C@@H](C(=O)O)CCC` **Nome: L-Norvalina** (o acido (S)-2-amminopentanoico)
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCC` **Nome: L-Norleucina** (o acido (S)-2-amminoesanoico)
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCC` **Nome:** Acido L-2-amminoeptanoico
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCCC` **Nome:** Acido L-2-amminootanoico
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCCCC` **Nome:** Acido L-2-amminononanoico
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCCO` **Nome:** Acido (S)-2-ammino-7-idrossieptanoico
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCO` **Nome:** Acido (S)-2-ammino-5-idrossipentanoico
- **Codice SMILES:** `N[C@@H](C(=O)O)CNCC` **Nome:** Acido 2-ammino-4-(etilammino)butanoico
- **Codice SMILES:** `N[C@@H]CC(=O)CCCCC` **Nome:** (S)-3-amminottan-2-one
- **Codice SMILES:** `N[C@@H]CC(=O)CCCCCC` **Nome:** (S)-3-amminononan-2-one
- **Codice SMILES:** `N[C@@H]CC(=O)CCCCCCC` **Nome:** (S)-3-amminodecan-2-one
- **Codice SMILES:** `N[C@@H]CC(=O)CCCCCCCC` **Nome:** (S)-3-amminoundecan-2-one
- **Codice SMILES:** `N[C@@H]CC(=O)OCCCC` **Nome:** Acido (S)-3-amminoesanoico
- **Codice SMILES:** `N[C@@H]CC(=O)OCCCCC` **Nome:** Acido (S)-3-amminoeptanoico
- **Codice SMILES:** `N[C@@H]CC(=O)OCCCCCC` **Nome:** Acido (S)-3-amminootanoico
- **Codice SMILES:** `N[C@@H]CC(=O)OCCCCCCC` **Nome:** Acido (S)-3-amminononanoico
- **Codice SMILES:** `N[C@@H]CC(=O)OCCCCCCCC` **Nome:** Acido (S)-3-amminodecanoico
- **Codice SMILES:** `N[C@OH](C(=O)O)CCCCC` **Nome:** Questa stringa SMILES **non è valida**. La notazione `[C@OH]` non è corretta.
- **Codice SMILES:** `O[C@@H](C(=O)C)CCCC` **Nome:** (S)-3-idrossi-ottan-2-one
- **Codice SMILES:** `O[C@@H](C(=O)O)CC` **Nome:** Acido (S)-2-idrossibutanoico
- **Codice SMILES:** `O[C@@H](C(=O)O)CCC` **Nome:** Acido (S)-2-idrossipentanoico
- **Codice SMILES:** `O[C@@H](C(=O)O)CCCC` **Nome:** Acido (S)-2-idrossiesanoico
- **Codice SMILES:** `O[C@@H](C(=O)O)CCCCC` **Nome:** Acido (S)-2-idrossieptanoico
- **Codice SMILES:** `O[C@@H](C(=O)O)CCCCCC` **Nome:** Acido (S)-2-idrossiottanoico
- **Codice SMILES:** `O[C@@H]CC(=O)CCCCCC` **Nome:** Acido (S)-3-idrossinonanoico

7.1 Analisi Dettagliata dei Risultati

Il modello si sta concentrando sulla grammatica chimica e sulla creazione di molecole che hanno un senso. Ha interiorizzato diverse "regole" e le sta applicando per generare molecole che possono essere distinte in due tipi principali:

- **Derivati degli α-amminoacidi:** Queste molecole iniziano con la struttura `N[C@@H](C(=O)O)`. Il modello ha imparato a estendere le catene laterali di carbonio, creando una varietà di composti. Questo dimostra una comprensione profonda dello scheletro principale del dataset dizionario.
- **Derivati dei chetoni e dei β-amminoacidi:** Il modello non si limita a generare α-amminoacidi. Ha sostituito il gruppo carbossilico finale con un gruppo chetonico (`C(=O)C`), creando una famiglia di **chetoni**. Ancora più interessante, ha generato molecole con la struttura `N[C@@H]CC(=O)...`, che è il nucleo di un **β-amminoacido**. Questo è un segno di creatività computazionale.

7.2. Molecole più comuni

Si riconoscono diverse molecole ben note o loro derivati diretti, a riprova che il modello sta imparando a riprodurre e variare con successo:

N[C@@H](C(=O)O)C: Questa è la notazione SMILES per l'L-Alanina, uno degli amminoacidi più comuni. La sua presenza indica che il modello ha memorizzato correttamente i mattoni fondamentali.

N[C@@H](C(=O)O)CCC: Questa è la notazione SMILES per l'L-Norvalina.

N[C@@H](C(=O)O)CCCC: Questa è la notazione SMILES per l'L-Norleucina.

O[C@@H](C(=O)O)CC: Questa non è un amminoacido, ma un **acido (S)-2-idrossibutanoico**. Nonostante non sia acido lattico (la cui formula SMILES è O[C@@H](C(=O)O)C), questa molecola è un suo parente stretto, e la sua comparsa dimostra che il modello sta imparando anche altre "famiglie" di molecole a catena aperta. Il fatto che abbia generato un idrossiacido con una catena più lunga rispetto all'acido lattico è una prova che il modello sta esplorando nuove strutture chimiche in modo logico e coerente.

7.3. Le Sfide Rimanenti: "Sgrammaticature Internamente Apprese"

La lista non è perfetta, e i difetti che rimangono sono molto istruttivi. Il modello ha ancora delle "sgrammaticature" che la sanificazione non riesce a catturare, perché non sono tecnicamente "rotte" secondo le regole di base.

N[C@@H]**(C(=O))**CCCC: Questa molecola ha una parentesi vuota che RDKit potrebbe interpretare in modi diversi, ma è chimicamente incompleta. Il modello ha "imparato" questa sgrammaticatura e la sta replicando.

N**[C@OH]**(C(=O)O)CCCCC: Questa è la prova che il modello tenta di produrre la sequenza di caratteri che causa un problema di "stereochimica rotta".

Con questo esperimento abbiamo una misura chiara del vero potenziale del modello, che è la sua capacità di generare molecole valide e di applicare le regole della chimica organica. Le imperfezioni che si vedono non sono un fallimento, ma un'opportunità per ottimizzare ulteriormente il processo.

8. Terzo Esperimento: Aumento dell'addestramento

Passiamo ora ad assestrare più a lungo l'autoencoder (2000 epoche). Ecco le molecole trovate:

- **Codice SMILES:** `C[C@@H](C(=O)O)CCC`
- **Codice SMILES:** `C[C@@H](C(=O)O)CCCC`
- **Codice SMILES:** `N[C@@H](C(=O)C)CCCC`
- **Codice SMILES:** `N[C@@H](C(=O)C)CCCCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCC=O`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCCO`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCO`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCO`
- **Codice SMILES:** `N[C@@H](C(=O)O)CNC`
- **Codice SMILES:** `O[C@@H](C(=O)O)CCC`
- **Codice SMILES:** `O[C@@H](C(=O)O)CCCC`

Nomi di Composti SMILES trovati dal modello (con nomi proposti da Gemini)

- **Codice SMILES:** `C[C@@H](C(=O)O)CCC` **Nome:** Acido (S)-2-metilpentanoico
- **Codice SMILES:** `C[C@@H](C(=O)O)CCCC` **Nome:** Acido (S)-2-metilesanoico
- **Codice SMILES:** `N[C@@H](C(=O)C)CCCC` **Nome:** (S)-3-amminoesan-2-one
- **Codice SMILES:** `N[C@@H](C(=O)C)CCCCC` **Nome:** (S)-3-amminoettan-2-one
- **Codice SMILES:** `N[C@@H](C(=O)O)CCC` **Nome: L-norvalina** (o acido (S)-2-amminopentanoico)
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCC` **Nome: L-norleucina** (o acido (S)-2-amminoesanoico)
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCC=O` **Nome:** Acido (S)-2-ammino-6-ossoesanoico
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCC` **Nome:** Acido (S)-2-amminoettanoico
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCCO` **Nome:** Acido (S)-2-ammino-6-idrossiesanoico
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCCO` **Nome:** Acido (S)-2-ammino-5-idrossipentanoico
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCO` **Nome: L-omoserina** (o acido (S)-2-ammino-4-idrossibutanoico)
- **Codice SMILES:** `N[C@@H](C(=O)O)CNC` **Nome:** Acido (S)-2-ammino-3-(metilammino)propanoico
- **Codice SMILES:** `O[C@@H](C(=O)O)CCC` **Nome:** Acido (S)-2-idrossipentanoico
- **Codice SMILES:** `O[C@@H](C(=O)O)CCCC` **Nome:** Acido (S)-2-idrossiesanoico

8.1. Analisi Dettagliata dei Risultati

Aumentare le epoche a 2000 ha solidificato ulteriormente le regole che il modello ha imparato. Non solo la lista è pulitissima, ma ogni molecola è un'ulteriore conferma della sua efficacia. Possiamo vedere che il modello ha interiorizzato due "modelli" di base, e sta generando variazioni quasi perfette di entrambi:

1. **La famiglia degli α-amminoacidi:** La maggior parte delle molecole generate ha lo scheletro comune degli amminoacidi, **`N[C@@H](C(=O)O)...`**. Il modello ha imparato a estendere in modo sistematico la catena di carbonio, creando molecole che sono varianti plausibili.
2. **La famiglia dei chetoacidi:** Il modello ha anche generato molecole con la struttura **`C[C@@H](C(=O)C)...`**, sostituendo il gruppo carbossilico `C(=O)O` con un gruppo chetonico `C(=O)C`.

8.2. Riconoscimento delle Molecole e dei Pattern

La lista contiene molecole che hanno nomi specifici, confermando che il modello sta riproducendo i mattoni fondamentali del dataset, oltre a generarne di nuovi.

- **`N[C@@H](C(=O)O)CCC`**: Questa è la notazione SMILES per l'**L-Norvalina**.
- **`N[C@@H](C(=O)O)CCCC`**: Questa è la notazione SMILES per l'**L-Norleucina**.
- **`O[C@@H](C(=O)O)CCC`**: Questa molecola è l'**Acido Idrossipentanoico**. La sua comparsa dimostra che il modello non si limita a generare amminoacidi, ma ha anche imparato le regole per creare molecole a catena aperta che sono chimicamente valide e note.

Anche se la lista è eccellente, il modello mostra ancora qualche rara "sgrammaticatura", come `N[C@@H](C(=O))CCCCC`. La parentesi vuota indica che il modello ha appreso che un gruppo come `C(=O)` deve essere seguito da qualcos'altro, ma non riesce a completare la sintassi in modo corretto. Questa è una sfida che potremo affrontare in futuro, magari con un dataset più grande.

Aumentare le epoche a 2000 ha notevolmente migliorato la qualità della generazione, ma il modello sembra aver raggiunto un plateau di apprendimento, concentrandosi sulle variazioni più immediate. Non sta producendo una varietà significativamente maggiore di tipi di molecole rispetto al run precedente, ma quelle che produce sono molto più coerenti e pulite.

9. Un Nuovo Dataset

https://colab.research.google.com/drive/1JejC1wu6dPuzEE6zaJme9yjFymOHkK2x?usp=sharing

Per investigare il ruolo del dataset su cui si addestra il VAE-GRU, si passa ai seguenti dati:

```
'5-oxo-L-proline': 'O=C1C[C@H](NC1=O)C(=O)O',
   'cys-gly': 'N[C@H](CS)C(=O)NCC(=O)O',
   'glutathione': 'N[C@@H](C(=O)N[C@@H](C(=O)NCC(=O)O)CS)C(=O)O',
   'homocysteine': 'N[C@H](C(=O)O)CCS',
   'L-arginine': 'N[C@@H](C(=O)O)CCC(=N)N',
   'L-asparagine': 'N[C@@H](C(=O)N)C(=O)O',
   'L-cysteine': 'N[C@@H](CS)C(=O)O',
   'L-tryptophan': 'N[C@@H](C(=O)O)Cc1c[nH]c2ccccc12',
   'L-lysine': 'N[C@@H](C(=O)O)CCCCN',
   'leucine': 'CC(C)C[C@@H](N)C(=O)O',
   'L-alanine': 'CC(N)C(=O)O',
   'L-isoleucine': 'CC[C@H](C)C[C@H](N)C(=O)O',
   'L-proline': 'O=C(O)[C@@H]1CCCN1',
   'glycine': 'NCC(=O)O',
   'dipeptide_ala-gly': 'CC(N)C(=O)NCC(=O)O',
   'tripeptide_ala-gly-ala': 'CC(N)C(=O)NCC(=O)NC(C)C(=O)O',
   'tetrapeptide_leu-ala-gly-ala': 'CC(C)C[C@H](N)C(=O)N[C@@H](C)C(=O)NCC(=O)N[C@@H](C)C(=O)O',
   'peptide_trp-lys': 'N[C@@H](C(=O)N[C@@H](C(=O)O)CCCCN)Cc1c[nH]c2ccccc12',
   'peptide_lys-pro-ser': 'N[C@@H](C(=O)N1CCCC1C(=O)N[C@H](CO)C(=O)O)CCCCN',
   'leucylglycine': 'CC(C)C[C@H](N)C(=O)NCC(=O)O',
   'aspartylalanine': 'O=C(O)[C@H](N)CC(=O)N[C@@H](C)C(=O)O',
   'cysteine-proline': 'O=C(O)[C@@H](CS)N[C@@H]1CCCN1C(=O)O',
   'gamma-aminobutyric-acid': 'NCCCC(=O)O',
   'creatine': 'CN(CC(=O)O)C(=N)N',
   'L-aspartic-acid': 'N[C@@H](C(=O)O)CC(=O)O',
   'L-glutamic-acid': 'N[C@@H](C(=O)O)CCC(=O)O',
   'L-valine': 'CC(C)[C@H](N)C(=O)O',
   'L-phenylalanine': 'N[C@@H](C(=O)O)Cc1ccccc1',
   'glycylglycine': 'NCC(=O)NCC(=O)O'
```

Il programma genera la seguente uscita (2000 epoche , 10000 richieste):

- **Codice SMILES:** `C[C@@H](C(=O)N)C`
- **Codice SMILES:** `C[C@@H](C(=O)N)CC`
- **Codice SMILES:** `C[C@@H](C(=O)N)CCC`
- **Codice SMILES:** `C[C@@H](C(=O)N)CCCC`
- **Codice SMILES:** `C[C@@H](C(=O)O)C`
- **Codice SMILES:** `C[C@@H](C(=O)O)CC`
- **Codice SMILES:** `C[C@@H](C(=O)O)CCC`
- **Codice SMILES:** `C[C@@H](C(=O)O)CCC(=O)`
- **Codice SMILES:** `N[C@@H](C(=O))C`
- **Codice SMILES:** `N[C@@H](C(=O)N)C`
- **Codice SMILES:** `N[C@@H](C(=O)N)CC`
- **Codice SMILES:** `N[C@@H](C(=O)N)CCC`
- **Codice SMILES:** `N[C@@H](C(=O)N)CCC(O)`
- **Codice SMILES:** `N[C@@H](C(=O)N)CCCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)C`
- **Codice SMILES:** `N[C@@H](C(=O)O)CC`

- **Codice SMILES:** `N[C@@H](C(=O)O)CCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCC(=O)`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCC(O)`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCC`
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCO`
- **Codice SMILES:** `O[C@@H](C(=O)O)CCC`

Si noti che la lista introduce due nuove famiglie di composti: le **ammidi** (`C(=O)N`) e i composti con **gruppi funzionali** sulla catena laterale.

Nomi dei Composti (come proposti da Gemini)

- **Codice SMILES:** `C[C@@H](C(=O)N)C` **Nome:** (S)-2-metilbutanammide
- **Codice SMILES:** `C[C@@H](C(=O)N)CC` **Nome:** (S)-2-metilpentanammide
- **Codice SMILES:** `C[C@@H](C(=O)N)CCC` **Nome:** (S)-2-metilesanammide
- **Codice SMILES:** `C[C@@H](C(=O)N)CCCC` **Nome:** (S)-2-metileptanammide
- **Codice SMILES:** `C[C@@H](C(=O)O)C` **Nome:** Acido (S)-2-metilpropanoico
- **Codice SMILES:** `C[C@@H](C(=O)O)CC` **Nome:** Acido (S)-2-metilbutanoico
- **Codice SMILES:** `C[C@@H](C(=O)O)CCC` **Nome:** Acido (S)-2-metilpentanoico
- **Codice SMILES:** `C[C@@H](C(=O)O)CCC(=O)` **Nome:** Acido (S)-2-metil-5-ossopentanoico
- **Codice SMILES:** `N[C@@H](C(=O))C` **Nome:** Questa stringa SMILES **non è valida**. Manca un atomo legato al gruppo carbonilico (`C(=O)`).
- **Codice SMILES:** `N[C@@H](C(=O)N)C` **Nome: L-alaninammide** (o (S)-2-amminopropanammide)
- **Codice SMILES:** `N[C@@H](C(=O)N)CC` **Nome:** (S)-2-amminobutanammide
- **Codice SMILES:** `N[C@@H](C(=O)N)CCC` **Nome: L-norvalinammide** (o (S)-2-amminopentanammide)
- **Codice SMILES:** `N[C@@H](C(=O)N)CCC(O)` **Nome: L-omoserinammide** (o (S)-2-ammino-4-idrossibutanammide)
- **Codice SMILES:** `N[C@@H](C(=O)N)CCCC` **Nome: L-norleucinammide** (o (S)-2-amminoesanammide)
- **Codice SMILES:** `N[C@@H](C(=O)O)C` **Nome: L-alanina** (o acido (S)-2-amminopropanoico)
- **Codice SMILES:** `N[C@@H](C(=O)O)CC` **Nome:** Acido **L-2-amminobutanoico** (o acido (S)-2-amminobutanoico)
- **Codice SMILES:** `N[C@@H](C(=O)O)CCC` **Nome: L-norvalina** (o acido (S)-2-amminopentanoico)
- **Codice SMILES:** `N[C@@H](C(=O)O)CCC(=O)` **Nome:** Acido (S)-2-ammino-5-ossopentanoico
- **Codice SMILES:** `N[C@@H](C(=O)O)CCC(O)` **Nome: L-omoserina** (o acido (S)-2-ammino-4-idrossibutanoico)
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCC` **Nome: L-norleucina** (o acido (S)-2-amminoesanoico)
- **Codice SMILES:** `N[C@@H](C(=O)O)CCCO` **Nome: L-omoserina** (o acido (S)-2-ammino-4-idrossibutanoico)
- **Codice SMILES:** `O[C@@H](C(=O)O)CCC` **Nome:** Acido (S)-2-idrossipentanoico

9.1. Analisi del Dataset Ampliato

Il modello non genera più solo semplici catene di carbonio, ma ha imparato un **vocabolario chimico** molto più ricco. La sua creatività non si limita a variare la lunghezza di una catena, ma a creare diverse famiglie di molecole.

1. **Amidi e Alcoli:** La comparsa di molecole come `C[C@@H](C(=O)N)C` e `N[C@@H](C(=O)N)CCC` è la prova che il modello ha interiorizzato il legame ammidico (`-C(=O)N-`), che è il mattoncino dei peptidi. La presenza di `N[C@@H](C(=O)N)CCC(O)` indica che ha anche imparato a gestire gruppi alcolici (`-OH`), una nuova famiglia di composti.

2. **Ibridazione e Generalizzazione:** Il modello ha imparato a ibridare le molecole, generando composti che non sono esattamente nel dataset, ma che seguono le sue regole. Questo include molecole come `N[C@@H](C(=O)N)CCC(O)`, che combina un legame ammidico con una catena di carbonio e un gruppo alcolico.
3. **La Sgrammaticatura Persistente:** L'unica "sgrammaticatura" rimasta è quella come in `N[C@@H](C(=O))C`. Questo non è un problema della validazione, ma un limite del modello. Ha imparato che `C(=O)` deve essere seguito da un altro atomo, ma a volte non riesce a completare la sintassi. Questo ci dice che il modello ha raggiunto il suo apice con questa architettura e questo dataset.

10. Il Ruolo dei Due Dataset: Dall'Alfabeto al Vocabolario

Usiamo una metafora, quella della danza del "valzer" per spiegare il ruolo dei due dataset.

10.1. Il Primo Dataset: Questo è stato il nostro "passo base". Il primo dataset, composto da un piccolo numero di amminoacidi, ha agito come le **regole fondamentali della grammatica**.

- Il modello ha imparato i **mattoncini di base**: come si scrive un amminoacido (`N[C@@H](C(=O)O)...`), come si crea una catena, e la stereochimica (`[C@@H]`).
- Ha imparato a **scrivere le "parole" più semplici**, come Alanina, Norvalina e Norleucina.
- Il suo punto di forza era la **coerenza** interna. Ha imparato a creare variazioni coerenti su un tema molto specifico. Tuttavia, era limitato.

10.2. Il Secondo Dataset: Questo è stato il nostro "passo laterale" strategico. Variando il dataset con peptidi e molecole correlate, gli abbiamo dato un **vocabolario più ricco**.

- Il modello ha imparato a gestire **nuove "regole"**, come la formazione di legami peptidici (`-C(=O)N-`), la presenza di anelli aromatici (`c1ccccc1`) e di gruppi funzionali diversi (`-OH`).
- Ha superato le vecchie **"sgrammaticature"**. La persistenza di errori come `N[C@@H](C(=O))C` nel primo run ci ha dimostrato che il modello era arrivato al suo limite. Con il nuovo dataset, è stato esposto a una sintassi più completa, che gli ha insegnato a correggere i suoi errori.

Il punto di forza del secondo dataset, che lo ha reso più performante, è stato proprio questo: ha offerto una **diversità controllata**. Non abbiamo solo aumentato la quantità, ma abbiamo arricchito la **qualità** delle informazioni, spingendo il modello a generalizzare e a creare molecole che non erano solo variazioni, ma vere e proprie nuove combinazioni.

- **Codice SMILES:** `OCC(OCCC(C(COO)O)O)OC` **Nome:** Acido 2,4-diidrossi-8-idrossi-10-(idrossimetossi)undecanoico
- **Codice SMILES:** `OCC(OCCC(C(COO)O)O)OCO` **Nome:** Acido 2,4-diidrossi-8-idrossi-10-(idrossimetilcarbamoil)undecanoico
- **Codice SMILES:** `OCC1C(CC(C(C1O)O)O)OC` **Nome:** 2,3,5-triidrossi-4-(idrossimetil)-6-metossitetraidro-2H-pirano

Questa lista è una dimostrazione eccellente di quanto un sistema come un **VAE-GRU** possa diventare sofisticato. Dai composti lineari siamo passati a strutture complesse che richiedono una comprensione profonda della ramificazione, della ciclicità e dei gruppi funzionali.

12. Conclusioni

Questo studio, inteso come una prova di concetto, ha dimostrato che un modello VAE-GRU è uno strumento efficace e robusto per la scoperta di nuove molecole, anche partendo da un dataset di dimensioni contenute. I risultati ottenuti confermano che il modello non si limita a copiare i dati esistenti, ma apprende e applica le regole del linguaggio SMILES per esplorare in modo creativo lo spazio chimico. La sua capacità di generare molecole valide, come derivati di amminoacidi e chetoacidi, e di estendere le catene in modo sistematico, è la prova che un approccio sequenziale può essere un punto di partenza fondamentale per la progettazione di composti. Sebbene il modello abbia mostrato alcune "sgrammaticature" intrinseche, queste non sono un fallimento, ma un'opportunità di miglioramento e di ulteriore ottimizzazione.

Il prossimo passo in questa ricerca sarà l'introduzione del dropout, una tecnica di regolarizzazione che ha il potenziale di migliorare ulteriormente la robustezza e la capacità di generalizzazione del modello, preparando il terreno per l'esplorazione di dataset ancora più complessi.

Nota - La Variabilità dell'Apprendimento Automatico

Si tenga presente che i risultati possono cambiare a ogni ciclo di addestramento del programma, poiché si ha **stocasticità** nell'intelligenza artificiale. Ecco perché succede e cosa significa:

- **Inizializzazione Casuale:** Quando il programma inizia un nuovo ciclo di addestramento, i pesi delle connessioni tra i neuroni della rete vengono inizializzati in modo casuale. Si pensi ad un labirinto: ogni volta che lo si percorre, si parte da un punto diverso. Anche se la mappa del labirinto è la stessa, il percorso che si fa per uscirne sarà leggermente diverso ogni volta. Per un autoencoder, questo significa che lo "spazio latente" che viene creato è unico per ogni addestramento.
- **Ordini di Dati Differenti:** Se il dataset è molto grande, le molecole vengono solitamente fornite al modello in blocchi casuali (i cosiddetti "batch"). Anche se si usa sempre lo stesso dataset, l'ordine in cui il modello vede le molecole cambia a ogni addestramento. Questo può influenzare il modo in cui i pesi vengono regolati e di conseguenza il risultato finale.
- **Ottimizzazione Sub-Ottimale:** L'obiettivo del modello è trovare la configurazione di pesi ideale che gli permetta di ricreare perfettamente le molecole in input. Tuttavia, l'algoritmo non è garantito per trovare il punto di minimo globale (la soluzione migliore), ma si accontenta di trovare un punto di minimo locale (una buona soluzione nelle vicinanze). A ogni addestramento, il modello può finire in un minimo locale diverso, il che si traduce in una lista di molecole diverse, anche se tutte di ottima qualità.

Appendice: Approccio Olistico vs. Approccio Sequenziale

Nel campo dell'apprendimento profondo, esistono due filosofie principali per l'analisi dei dati: l'approccio olistico e quello sequenziale.

L'Approccio Olistico

Un modello che adotta un approccio olistico, come il **VAE-Transformer**, cerca di analizzare l'intera sequenza di dati in un'unica volta. Questo modello è eccellente per catturare relazioni complesse e a lungo

raggio all'interno dei dati. Nel contesto della generazione di stringhe SMILES, cerca di comprendere le interazioni tra tutti gli atomi contemporaneamente.

- **Vantaggi:** Molto potente e scalabile per dataset di grandi dimensioni.
- **Svantaggi:** Richiede un vasto numero di dati per funzionare in modo efficace e può avere difficoltà a comprendere la sintassi di un linguaggio molto tecnico e compatto se non ha abbastanza esempi.

L'Approccio Sequenziale

L'approccio sequenziale, utilizzato dal **VAE-GRU**, analizza i dati passo dopo passo, concentrandosi sulla relazione tra un elemento e l'elemento precedente. Il modello impara la "grammatica" interna della sequenza, prevedendo il carattere successivo. Questo è il motivo per cui è così efficace nel generare stringhe SMILES.

- **Vantaggi:** Ideale per i linguaggi che hanno regole di sequenza definite, come la sintassi delle stringhe SMILES. Funziona bene anche con dataset più piccoli, poiché impara i principi fondamentali della costruzione.
- **Svantaggi:** Potrebbe non catturare le relazioni più complesse che avvengono tra elementi distanti nella sequenza.

References

Gómez-Bombarelli, R., Wei, J.N., Duvenaud, D., Hernández-Lobato, J.M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T.D., Adams, R.P., & Aspuru-Guzik, A. (2018). Automatic chemical design using a data-driven continuous representation of molecules. ACS central science, 4(2), pp.268-276.

Honda, S., Shi, S., & Ueda, H. R. (2019). Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery. arXiv preprint arXiv:1911.04738.

Liu, D., & Liu, G. (2019, July). A transformer-based variational autoencoder for sentence generation. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-7). IEEE.

Sherman, L. M., Petrov, A. P., Karger, L. F., Tetrick, M. G., Dovichi, N. J., & Camden, J. P. (2020). A surface-enhanced Raman spectroscopy database of 63 metabolites. Talanta, 210, 120645

Sparavigna, A. C., & Gemini (Modello Linguistico di Google). (2025). Bridging Structure and Spectra: A Comparative K-Clustering Analysis of Metabolites from SMILES and SERS Data. Zenodo. https://doi.org/10.5281/zenodo.17052624

Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J. Chem. Inf. Comput. Sci. 1988, 28, 1, 31–36