

# HW11

---

112652011 廖晨鈞

## Unanswered Questions

---

### 1. In what scenario does the Gradient descent method or the Newton's method perform better than the other one? What's the differences?

Ans:

Scenario for Gradient Descent (GD):

- Best for: High-dimensional problems (e.g., Deep Learning) or non-convex functions.
- Why: It is computationally cheap per iteration ( $O(n)$ ) because it only calculates the first derivative. Inverting the Hessian matrix for Newton's method is too expensive ( $O(n^3)$ ) for large variables.

Scenario for Newton's Method:

- Best for: Low-dimensional problems where high precision is required and the function is strictly convex.
- Why: It takes far fewer steps to reach the minimum because it considers curvature.

Key Differences:

- Information Used: GD uses First-order (slope/gradient); Newton's uses Second-order (curvature/Hessian).
- Convergence Speed: GD has Linear convergence (slow, zig-zags near the optimum); Newton's has Quadratic convergence (extremely fast near the optimum).

Reference:

Nocedal, J., & Wright, S. J. (2006). Numerical Optimization (2nd ed.). Springer. (Chapter 3) <https://link.springer.com/book/10.1007/978-0-387-40065-5>

### 2. What is the geometric meaning of the assumption that simplifies GDA to LDA (i.e., $\Sigma_0 = \Sigma_1 = \Sigma$ )?

Ans:

- Geometric Meaning:

- \* It assumes that the probability distributions of the two classes have the exact same shape, size, and orientation.
- \* Visually, the data clusters form ellipsoids that are identical in curvature but are located at different positions (different means,  $\mu_0 \neq \mu_1$ ).

- Result:

- \* Because the "shapes" are identical, the quadratic terms in the probability equations cancel each other out.

- \* This forces the decision boundary to be a straight line (or a flat hyperplane in higher dimensions), hence the name Linear Discriminant Analysis (LDA). If  $\Sigma_0 \neq \Sigma_1$ , the boundary would be curved (Quadratic).

Reference:

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning* (2nd ed.). Springer. (Section 4.3)

<https://hastie.su.domains/ElemStatLearn/>

## Toy model

---

為了實現 De Novo Design (從頭藥物設計) 的雛形，我們使用 Variational Autoencoder (VAE，變分自編碼器) 作為核心架構。

與傳統機器學習不同，VAE 是分子生成領域的經典模型，其核心能力在於學習數據的潛在空間 (Latent Space)，進而具備創造新分子的能力。

---

## 1. 問題設計 (Problem Design)

### 說明與對應目標

我們的模型需要在一個連續的數學空間中導航，找到代表高療效藥物的座標，然後將其解碼為結構。VAE 正是這種「連續空間導航」的雛形。它允許我們在兩個已知藥物之間進行插值 (Interpolation)，找出它們的中間型態，這是新化學結構發現的邏輯基礎。

### 定義

\* 輸入 (Input)：藥物分子的 SMILES 字串 (例如：CC(=O)OC1=CC=CC=C1C(=O)O)。這是化學結構的 1D 序列表示法。

\* 輸出 (Output)：重構或生成的 SMILES 字串。

\* 任務目標 (Task)：

1. 壓縮：將離散的化學字串壓縮成連續的常態分佈 (Gaussian Distribution)。
2. 生成：從這個分佈中隨機採樣 (Sampling)，解碼出合乎語法且具有化學意義的新字串。

\* 資料形式 (Data Format)：One-hot 編碼的字元序列。

## 2. 模型與方法 (Model & Method)

我們選擇LSTM-based Variational Autoencoder (VAE)作為Toy Model的模型，理由如下：

1. 生成能力：傳統 Autoencoder 只能重構看過的數據，但 VAE 學習的是數據的機率分佈。表示我們可以從潛在空間的空白處採樣，創造出訓練集中不存在的分子。
2. 連續性：VAE 強制潛在空間呈現高斯分佈，這使得相似的分子在空間中距離相近，讓我們能進行分子演化的模擬。
3. 序列處理：使用 LSTM (長短期記憶網絡) 處理 SMILES 的語法依賴性 (如括號配對、環的閉合標記)。

## 3. 討論 (Discussion)

從這個簡化問題中學到了什麼？

1. 透過 VAE，我們確實能將離散的化學分子映射到連續的幾何空間中。結果應該能顯示，結構相似的藥物 (如含有苯環的分子) 在這個空間中距離會比較近。
2. 在「分子變形」的實驗中，我們可能會看到中間步驟出現一些無意義的字串。這表示化學空間雖然被壓縮了，但並不是所有潛在空間的點解碼後都是合法的化學分子。

## 一些困難

1. SMILES 的語法脆弱性 (Syntactic Fragility)：

這個 Toy Model 最明顯的問題是生成的 SMILES 經常會語法錯誤 (例如括號不匹配 `c((...))`)。SMILES 雖然是 1D 的簡便表示，但對於生成模型來說，很容易產生無效結構 (Invalid SMILES)。

> 未來的 AI 藥物設計正在轉向 Graph Neural Networks (GNN) 和 3D 幾何深度學習，直接在圖結構或 3D 空間中生成，而不是處理1D字串。

2. 多目標優化的缺失 (Lack of Multi-objective Optimization)：

這個 VAE 只是學會了生成看起來像藥物的字串，但並不知道該藥物的性質，例如親和力、毒性等。

> 未來的系統必須在生成過程中結合 Reinforcement Learning (RL)，將「親和力」、「合成可行性」和「毒性」作為獎勵函數 (Reward Function)，引導藥物模型的生成方向。