

# 深度學習 Lab1.

## 1 第一題 (50%)

目的: 練習 numpy 矩陣操作、matplotlib 畫圖

矩陣在資訊與數學領域中有廣泛應用，例如線性代數方程求解、線性變換、圖像處理以及機器學習模型的建立。除了這些常見應用外，矩陣也可以用來表示並計算某些線性遞迴關係，其中一個典型例子便是 Fibonacci 數列：

$$\begin{cases} Fib_0 = 0, \\ Fib_1 = 1, \\ Fib_n = Fib_{n-1} + Fib_{n-2}, \quad n \geq 2 \end{cases} \Rightarrow \begin{bmatrix} Fib_{n+1} \\ Fib_n \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} Fib_n \\ Fib_{n-1} \end{bmatrix}$$

Fibonacci 數列有個有趣的數學性質，其名為皮薩諾週期 (Pisano periods)。具體來說，皮薩諾週期是指 Fibonacci 數列中每一項  $Fib_i$  皆  $\text{mod } n, n > 1$  後，其餘數會有週期性。如  $n = 2$  時，由於 Fibonacci 數列前幾項為：0, 1, 1, 2, 3, 5, 8, 13, 21, ...， $\text{mod}$  後的結果為 0, 1, 1, 0, 1, 1, 0, 1, 1, ...，其中的週期為 0, 1, 1，週期長度為 3 (週期開頭一定是 0, 1)。另外可證明  $\text{mod } n$  後的序列週期長度不會超過  $6n$ ，一個週期較長的例子為  $n = 5$ ，其序列為：

0, 1, 1, 2, 3, 0, 3, 3, 1, 4, 0, 4, 4, 3, 2, 0, 2, 2, 4, 1, 0, 1, ... 週期長度為 20

### 第一小題 (30%)

請透過 numpy 計算三次  $n = 5$  的皮薩諾週期序列，即  $Fib_0 \sim Fib_{59}$  每一項  $\text{mod } 5$  (前 60 項)。

- 計算過程請使用矩陣運算，其餘 Fibonacci 數列算法不予以給分 (如遞迴、動態規劃等方式)。
- 計算過程中請保持計算  $\text{mod}$ ，若後面才計算將會扣分。
- 請將皮薩諾週期序列紀錄並收集於 Python list 或 numpy array 中。

### 第二小題 (20%)

請將第一小題計算的皮薩諾週期序列透過 matplotlib 畫成折線圖並觀察其循環現象 (如下圖)。

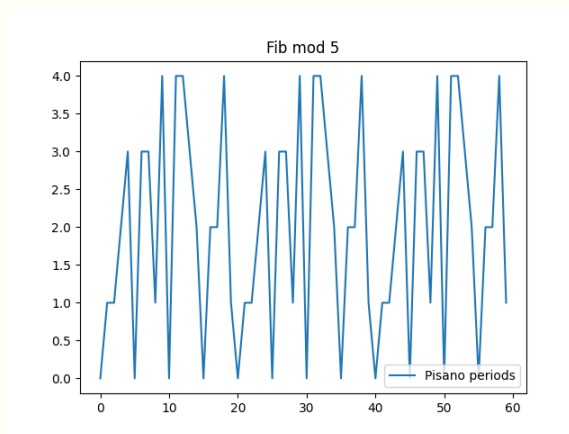


Figure 1: 第二小題圖輸出範例

- 請包含 title、標籤，文字與位置如題目圖片所示。
- 請輸出圖片到與 python 同級的路徑 output/lab1.1.png 中

## HINT:

- 宣告 numpy 的二維矩陣可以使用如 `A = np.array([[1, 0], [0, 1]])`，如此會得到二維單位矩陣
- 矩陣乘法可以使用 `Y = A @ X`，”@” 是 numpy 中矩陣乘法的符號

輸出 json 檔結構圖下圖，可以不用排版，注意 **key 的名稱要相同**。

```
{  
  "mod": 5,  
  "pisano_period": [0, 1, 1, 2, 3, 0, 3, 3, 1, 4, 0, 4, 4, ... (共60項)],  
}
```

## 2 第二題 (50%)

目的: 初探神經網路 (一顆神經元)

機器學習的各種演算法推動了人工智慧在醫療、金融、語音與影像等領域的應用。機器學習包含許多不同的演算法，其中類神經網路（深度學習）是最常用且發展最迅速的一類。雖然現今的生成式模型與大型語言模型十分複雜，但其基礎概念可以從最簡單的單一神經元出發，透過實作逐步理解其運作原理。

### 第一小題 (50%)

使用提供的資料集，利用 numpy 實作一個神經元擬合資料並將回歸直線以及資料點透過 matplotlib 描繪出來，同時記錄訓練過程中的 mse loss 變化。

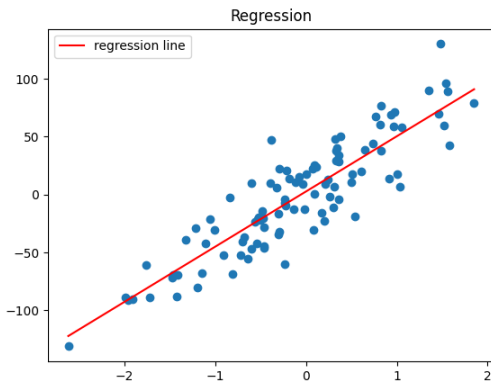


Figure 2: 結果範例圖

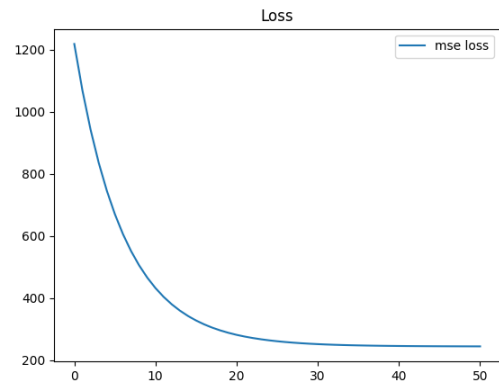


Figure 3: Loss 變化圖

HINT:

### Algorithm 1 Lab1 第二題流程

```
1:  $X, Y \leftarrow$  讀資料                                ▷ 讀完資料時,  $X$  shape = (100, 1),  $Y$  shape = (100,)
2:  $\mathbf{w} \leftarrow \text{random}()$                             ▷  $\mathbf{w}$  shape = (1,)
3:  $b \leftarrow \text{random}()$                                 ▷  $b$  shape = (1,)
4:  $lr \leftarrow 0.001$                                     ▷ 這裡可以自己嘗試, 只要  $lr \in (0, 1)$ 
5:
6:  $lossHistory = []$                                         ▷ 用來記錄 mse loss
7:  $loss \leftarrow$  使用所有的輸入資料計算 mse loss
8:  $lossHistory.append(loss)$                                 ▷ 訓練前先算一次 mse loss
9:
10: for  $epoch \leftarrow 1$  to  $maxEpoch$  do                    ▷  $maxEpoch$  可以設成 50
11:     for  $i \leftarrow 1$  to  $|X|$  do                                ▷ 迭代每一筆資料
12:          $\hat{y} \leftarrow \mathbf{w} \cdot X[i] + b$                 ▷ 計算神經元預測的值。注: 向量與向量之間用內積
13:
14:          $\mathbf{w} \leftarrow \mathbf{w} + lr * (Y[i] - \hat{y}) * X[i]$         ▷ 用梯度下降更新  $\mathbf{w}$ 
15:          $b \leftarrow b + lr * (Y[i] - \hat{y})$                 ▷ 用梯度下降更新  $b$ 
16:     end for
17:      $loss \leftarrow$  使用所有的輸入資料計算 mse loss
18:      $lossHistory.append(loss)$                                 ▷ 訓練後算一次 mse loss
19: end for
20:
21: 畫資料點與利用  $\mathbf{w}, b$  預測的回歸直線, 並輸出圖到檔案中
22: 畫 mse loss 的變化圖, 並輸出圖到檔案中
23: 輸出  $lossHistory$  到檔案中
```

- 利用隨機初始化神經元參數可以使用 `np.random.uniform()`  
<https://numpy.org/doc/stable/reference/random/generated/numpy.random.uniform.html> 使用方式為 `np.random.uniform(<low>, <high>, <size>)`  
 例如需要 `shape = (3,)` 的，則使用 `np.random.uniform(-1, 1, (3,))`，隨機範圍在  $(-1, 1)$   
 需要 `shape = (3, 1)` 的，則使用 `np.random.uniform(-1, 1, (3, 1))`
- 推薦  $w$  跟  $b$  初始數值不要太大，如可使用 `uniform(-1, 1)`
- 學習率不建議過大，可以嘗試使用  $lr = 0.001$
- 計算 mse loss 時記得要使用所有的資料計算
- 回歸直線的畫法可以使用 train data 的最小值與最大值畫直線，或是使用 `numpy.linspace()` 生成區間資料並畫圖，在矩陣或向量中找最大最小值可以使用 `A.max()` 或是 `A.min()`

輸出 json 檔結構圖下圖 (數值僅為範例)，可以不用排版，注意 **key 的名稱要相同**。

```
{
  "final_weight": 0.87,
  "final_bias": -0.12,
  "loss_history": [
    12.53,
    10.87,
    9.45,
    ...,
    0.56
  ],
  "num_epochs": 50,
  "learning_rate": 0.001
}
```

### 3 繳交注意事項

作業繳交期限為 10/9 23:59

繳交規定及評分方式如下，請符合以下要求，否則將斟酌扣分。

#### 3.1 繳交檔案要求

輸出結果檔案格式為 json，圖片存成 png。資料夾結構如下，並以 zip 壓縮後進行上傳：

- 班級\_學號\_姓名\_深度學習\_Lab1
  - Lab1-1
    - \* main.py
    - \* output
      - lab1.1.png
      - lab1.1\_output.json
  - Lab1-2
    - \* main.py
    - \* lab1.2\_train\_data.json [請勿上傳]
    - \* output
      - lab1.2\_loss.png
      - lab1.2\_result.png
      - lab1.2\_output.json
  - readme.txt

當中 output 資料夾中的檔案需要透過 main.py 當中創建。若有需要額外說明事宜，可另行付上 readme.txt。

請勿上傳輸入資料(如 train data)，但是程式中請保持使用相對路徑，讀資料時需假設所有輸入資料與 main.py 同級，若批改作業時程式讀取路徑錯誤將不予給分。

有使用 AI 查詢之處務必寫入 readme.txt。例如：查詢 ChatGPT numpy random 怎麼使用、矩陣乘法原本寫  $A*B$ ，但 ChatGPT 告訴我要使用  $A@B$  等。

#### 3.2 程式碼要求

- 為確保實驗可重現性，每份作業請設定亂數種子，如 `np.random.seed(seed=42)`
- 請詳細撰寫程式碼，並在每一行加上註解
- 根據各題提供的提示與要求，完成對應的程式內容
- 命名要具有可讀性，避免使用過於簡單或不具描述性的變數或函數名稱，例如 `a`、`los` 等
- 僅可使用 python 內建函式庫、numpy、matplotlib
- Python 版本請使用 3.12.11

請同學 不要使用 AI 工具生成作業。學習過程中，比起單純追求高分，更重要的是理解與掌握程式設計概念。有疑問時，可利用 AI 工具輔助、線上資源或詢問助教來協助學習。

### 3.3 評分方式

1. 正確性(工具應用、題目要求等) (45%)
2. 邏輯性(解法思路等) (45%)
3. 變數名稱、排版、註解內容 (10%)

若程式碼無法執行，或執行結果與繳交檔案不符(例如第一小題 json 檔案產生不一致)，則將不予以計分。

若發現抄襲行為，抄襲者與被抄襲者皆以 0 分計算。

直接使用 AI 工具產出作業者，該份作業以 0 分計算。

遲交一日，作業分數上限扣 10 分，直到 0 分。

對於作業有疑慮之處，助教將提出疑問，請同學解釋。