

# HW10.AI 音樂和聲優化器

在你的 AI 音樂實驗室中，你正在開發一個更精確的和聲模型。這個模型不僅有「weight」 $\omega$ ，還有一個「bias」 $b$ 。

模型公式為： $y = \omega \cdot x + b$ 。你的任務是找出最佳的  $\omega$  和  $b$ ，使其能夠最小化 AI 生成音樂的「損失函數」(Loss Function)。

注意：此題目的公式及自適應學習率邏輯依題目規定撰寫，實際上此方法僅為近似演算法，非正統線性回歸梯度推導。

這題請一定要使用 C++ 寫作業，否則不計分。

## ◆ 任務規則 1(輸入)

程式首先會讀取一行，包含六個初始參數：

$K$  (迭代總次數)、 $LR$  (初始學習率)、 $\omega$  (初始 weight)、 $b$  (初始 bias)、  
 $L$  (懲罰強度)、 $C$  (裁剪閾值)

接著，您的程式必須讀取多筆訓練資料，直到讀取至 -1 -1:

(每行包含兩個整數或小數： $x$  和  $y$ )

在讀取資料的同時，你必須計算並儲存六個關鍵的統計總和

1. 資料總筆數  $N$
2. 所有  $x$  的總和 ( $\sum x$ )
3. 所有  $y$  的總和 ( $\sum y$ )
4. 所有  $x * y$  的總和 ( $\sum xy$ )
5. 所有  $x * x$  的總和 ( $\sum x^2$ )
6. 所有  $y * y$  的總和 ( $\sum y^2$ )

## ◆ 任務規則 2(模型訓練)：

接著，程式進入一個  $K$  次的 for 迴圈 (即「迭代」)。

您必須宣告變數，用來追蹤在所有迭代過程中產生過的最佳  $\omega$ 、最佳  $b$  以及最低的「總損失」。

在每一次迭代中，您都必須依序執行以下所有步驟：

- A. 總損失 (Total\_Loss) :

$$\text{Total\_Loss} = \text{MSE} + \text{L2\_Penalty}$$

- B. 均方差 (MSE) :

$$\begin{aligned}\text{MSE} = & \frac{1}{N} \cdot \left( \left( \sum y^2 \right) - 2\omega \left( \sum xy \right) - 2b \left( \sum y \right) + \omega^2 \left( \sum x^2 \right) \right. \\ & \left. + 2\omega b \left( \sum x \right) + N \cdot b^2 \right)\end{aligned}$$

(註：公式中的  $\sum x$  即為您在任務規則 1 計算的，依此類推)

### C. L2 懲罰 (L2\_Penalty) :

$$L2\_Penalty = L \cdot (\omega^2 + b^2)$$

### • D. 梯度 (Gradients) :

- w 的梯度 (grad\_w) :

$$\text{grad}_w = \frac{1}{N} \cdot (-2 \left( \sum xy \right) + 2\omega \left( \sum x^2 \right) + 2b \left( \sum x \right)) + 2 \cdot L \cdot \omega$$

- b 的梯度 (grad\_b) :

$$\text{grad}_b = \frac{1}{N} \cdot (-2 \left( \sum y \right) + 2\omega \left( \sum x \right) + 2 \cdot b \cdot N) + 2 \cdot L \cdot b$$

### • E. 追蹤最佳解 :

- 比較本次計算出的「總損失」是否低於您目前追蹤到的「最低總損失」。
- 如果是，您就必須更新「最佳  $\omega$ 」、「最佳  $b$ 」和「最低總損失」為當前的數值。

### • F. 自適應學習率 (Adaptive LR) :

- 比較本次的「總損失」與上一次的「總損失」。
- 如果「總損失」惡化了（即本次  $>$  上次），將「學習率」減半 ( $LR = LR * 0.5$ )。
- 如果「總損失」不變或改善了（即本次  $\leq$  上次），稍微加速學習 ( $LR = LR * 1.05$ )。
- 第一次迭代不作調整

### • G. 梯度裁剪 (Gradient Clipping) :

- 為了防止訓練過程「爆炸」，您必須檢查  $\text{grad}_w$  和  $\text{grad}_b$ 。
- 如果  $\text{grad}_w$  大於「裁剪閾值 ( $C$ )」，則強制  $\text{grad}_w = C$ 。
- 如果  $\text{grad}_w$  小於  $C$  的負值 ( $-C$ )，則強制  $\text{grad}_w = -C$ 。  
(對  $\text{grad}_b$  做同樣操作)

### • H. 更新權重 :

最後，使用  $\omega = \omega - LR * \text{grad}_\omega$  和  $b = b - LR * \text{grad}_b$  來更新  $\omega$  和  $b$ ，以供下一次迭代使用。

- 您的程式必須能處理「沒有輸入任何資料」( $N=0$ )的情況，此時不應執行迭代，並以初始值作為最佳解，Loss 應為 0。

## Input

- 第 1 行： $K \ LR \ \omega \ b \ L \ C$  (六個參數)
- 第 2 至  $N+1$  行： $x \ y$  (多筆訓練資料)

- 結束行：`-1 -1`

( $K$  為整數( $1 \leq K \leq 500$ )， $LR$  為小數( $0.0 \leq LR \leq 0.1$ )， $\omega$  為小數( $-100.0 \leq \omega \leq 100.0$ )， $b$  為小數( $-100.0 \leq b \leq 100.0$ )， $L$  為小數( $0.0 \leq L \leq 10.0$ )， $C$  為小數( $1.0 \leq C \leq 10000.0$ )， $x$  和  $y$  為小數( $-1000.0 \leq x, y \leq 1000.0$ )， $0 \leq N \leq 1000$ )

## Output

- 僅 3 行：
  - 程式在所有模擬結束後，只印出 3 個數字，每個數字單獨一行  
(答案所有浮點數請跟第九次作業一樣使用 `setprecision` 四捨五入到小數點第 2 位)
- [best\_loss.XX] (最低總損失)  
 [best\_w.XX] (最佳  $\omega$ )  
 [best\_b.XX] (最佳  $b$ )

Sample Input 1	Sample Output 1
200 0.01 0.0 0.0 0.0 1000.0 1 2 2 4 3 6 -1 -1	0.00 2.00 0.00

Sample Input 2	Sample Output 2
100 0.01 0.5 0.2 0.1 1000.0 -1 -1	0.00 0.50 0.20

## 繳交格式

1. 上傳內容須為 **.cpp** 檔
  - 檔名為：班級 XX-學號  
(XX: 第幾份作業, i.e. A09-114502000)
2. 請針對程式內的重要功能進行「適當註解」，簡單說明該段程式碼的作用，註解的部分會納入評分考量，請勿隨意註解
3. 記得在 Online Judge 系統上測試，若在正常繳交期限前未有測試紀錄，將扣該次作業 10 分
4. 若檔名錯誤，扣該次作業 5 分

## 繳交期限

- 2025/11/28 23:59 前將檔案上傳到 ee-class 上

## 遲交期限

- 2025/11/30 23:59 前將檔案上傳到 ee-class 上
- 該次作業分數打八折
- 遲交期限過後不再開放繳交作業，該次以 0 分計算