

即時系統期末報告

組長：沈張德

組員：張軍斌、謝晉豪、曾韻如

壹、前言

本次即時系統之專題報告，乃建立在「配合病人作息之多種藥物服用流程安排」此一基礎改念上，期望能找出較符合醫療實務運用的排程演算法，並將該方法依照已建立之虛擬模型做模擬，驗證其實用上之效益。

貳、設計概念

一、簡述

這次專案的評分標準總共有 3 項：

1. Hit rate
2. Annoyance rate
3. Error rate

第一項主要是評斷預測使用者使用手機的準確度；第二項是越多鬧鐘(包含貪睡鬧鐘)則越不好；第三項則是判斷有沒有在正確的時間內服用各項藥物。

我們認為對於一位病患而言，在正確的時間服用藥物是非常重要的。而且預測手機的使用時間具有不確定性，相較之下，藥物的服用時段是預先定義好的。對於提醒吃藥這個排程問題而言，我們認為可以從設計滿足第三項 Error rate 為零來著手，再透過預測使用手機的資料來調整第一、二項指標。如此一來也可以簡化問題。

以下將分兩部份列出及說明排程方法的 pseudo code。第一部份為滿足 Error rate 為零的鬧鐘時間、第二部分則為貪睡鬧鐘的設計方法。

二、Pseudo code

鬧鐘設計方法：

1. Find median of each medicine
2. If median(medicine) \neq integer
Chop median(medicine) to integer
3. Grouping: check average duration
$$\text{AVERAGE_DURATION} = \frac{\sum_{i=1}^n (\text{upper time limit of medicine } i) - (\text{lower time limit of medicine } i)}{n}$$

Chop the AVERAGE_DURATION to integer.
4. For i=1 to (number of medicine)
If duration of medicine(i) < AVERAGE_DURATION
Outlier_medicine(i) = medicine(i)

5. Group the remaining medicine into pairs so that

$$\min \sqrt{\left(\text{median}(MED_i) - \text{median}(MED_j)\right)^2 + \left(\text{median}(MED_k) - \text{median}(MED_l)\right)^2 + \dots}$$

6. Group the outliers

Takeout two medicines i, j from the outlier group, $i \neq j$,

If $[\text{abs}(\text{median}(\text{outlier_med}(i)) - \text{median}(\text{outlier_med}(j))) < \text{min}(\text{duration}(\text{outlier_med}(i)), \text{duration}(\text{outlier_med}(j)))]$

Pair outlier_med i and j together.

7. Choose one median in each pairs as a cycling time so that

$$\min L.C.M. \left(\text{cycTime}(\text{pair}_1), \text{cycTime}(\text{pair}_2), \dots, \text{cycTime}(\text{pair}_n), \text{cycTime}(\text{outlierPair}_1), \dots, \text{cycTime}(\text{outlierPair}_m) \right)$$

8. Start cycling.

三、 Pseudo code 解釋

這個藥物排程的演算法主要是將相近性質(中位數、週期)的藥物兩兩配成一組，每組從兩個中位數中以達成最小公倍數為目標選擇一個中位數作為循環週期。

這次總共有 6 種藥物，分別為：

藥物 0：4~8 小時

藥物 1：2~4 小時

藥物 2：5~9 小時

藥物 3：5~7 小時

藥物 4：3~6 小時

藥物 5：3~7 小時

步驟 1：

各藥物的中位數為：

藥物 0：6

藥物 1：3

藥物 2：7

藥物 3：6

藥物 4：4.5

藥物 5：5

步驟 2：

藥物 4 的中位數為 4.5，無條件捨去為 4。

步驟 3：

$AVERAGE_DURATION = 3.1$

無條件捨去後， $AVERAGE_DURATION = 3$

步驟 4：

藥物 1 和藥物 3 的週期為 2 小時，小於 3 小時，因此將這兩種藥物設定為 outlier，不列入步驟 5 之分群集合。

步驟 5：

將有最相近的中位數之藥物兩兩配對。符合步驟 5 中之式子的組合為：藥物 0 和藥物 2 一組，藥物 4 和藥物 5 一組。

步驟 6：

將 outlier 兩兩配對為一組。根據步驟 6 的條件，藥物 1 和藥物 3 之中位數的”距離” 沒有小於此兩種藥物時段之最小值

$6 - 3 = 3 > (2, 2) = 2$ 。因此此兩個藥物不會配對在一起。因為 outlier 藥物只有兩個，因此步驟 6 的配對就此結束，藥物 1 和藥物 3 將各自以其中位數作為週期循環。

步驟 7：

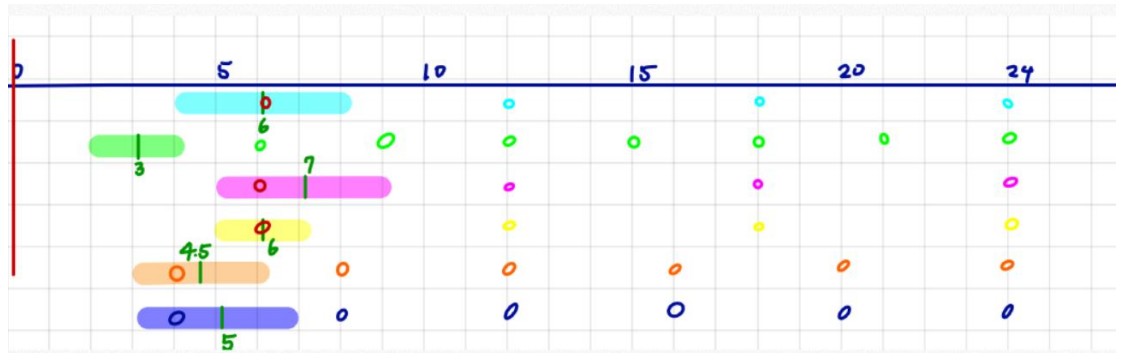
選擇各組之一中位數作為循環週期。以目前的組合而言，藥物 0、2 選擇 6 小時，藥物 4、5 選擇 4 小時，將會有各種組合中的最小公倍數。

步驟 8：

0：00 第一次吃藥後，各藥物的鬧鐘將會以步驟 7 之結果設定週期，如下表所示。

藥物	正常鬧鐘週期(小時)
0	6
1	3
2	6
3	6
4	4
5	4

四、 依照 Pseudo code 排出的初步鬧鐘



五、 設定貪睡鬧鐘

貪睡鬧鐘主要根據預測之手機使用資料作為設定依據，可調之參數為多久響一次貪睡鬧鐘，若使用者一直沒有吃藥，則貪睡鬧鐘在正常鬧鐘一小時後就不會再設定，因為若超過 1 小時，則有可能導致下次的正常提醒鬧鐘落在吃藥時間之前(太早響)。這次將貪睡鬧鐘設為每 15 分鐘響一次，因此次吃藥時間最多後面會有 4 個貪睡鬧鐘。

貪睡鬧鐘之設計方法如下：

1. 根據不同的天數，在鬧鐘響起的時候，會參考模型預測的結果。
2. IF 提醒吃藥時有用手機(包含正常鬧鐘和貪睡鬧鐘)，用 uniform distribution 產生 0~1 之間之亂數，若小於 0.2，則產生 15 分鐘後之貪睡鬧鐘。
3. IF 提醒吃藥時沒有用手機(包含正常鬧鐘和貪睡鬧鐘)，用 uniform distribution 產生 0~1 之間之亂數，若小於 0.9，則產生 15 分鐘後之貪睡鬧鐘。
4. 最多產生 4 次貪睡鬧鐘。

參、 模型說明

一、 概述

我們使用時間預測 Model: “Full-Fledged WaveNet”，利用卷積神經網絡進行高維時間序列預測。

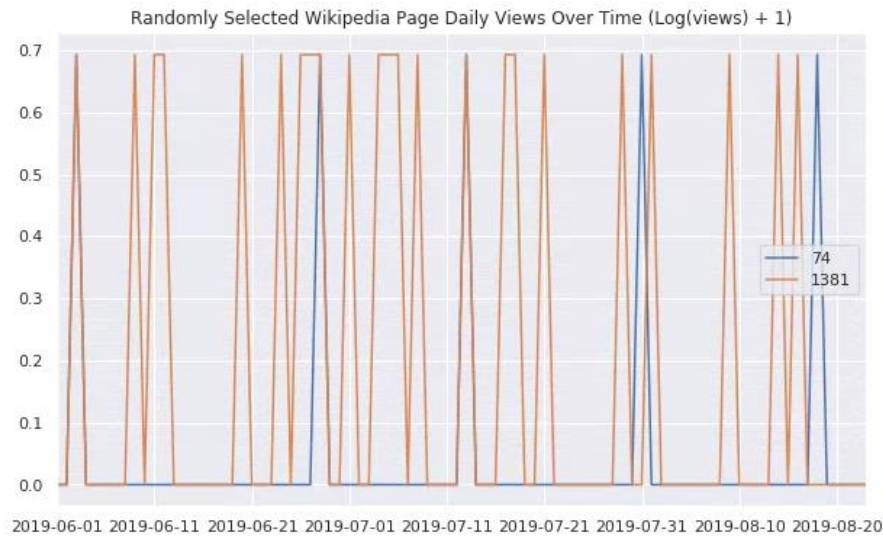
二、 數據處理

將數據整理成以分鐘為單位，每分鐘內只要有一秒處於使用手機時間內，則標志位設為 1，否則為 0。數據格式如下，第一列為 0-1440 分鐘，第一行為 0-83 天。

num	2019-06-01	2019-06-02	2019-06-03	2019-06-04	2019-06-05	2019-06-06	2019-06-07	2019-06-08	2019-06-09	2019-06-10	2019-06-11	2019-06-12	2019-06-13	2019-06-14	2019-06-15	2019-06-16	2019-06-17	2019-06-18	2019-06-19	2019-06-20	2019-06-21	2019-06-22	2019-06-23	2019-06-24	2019-06-25
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
4	4	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1

5 rows x 85 columns

下圖為隨機輸出的某兩行數據圖，橫坐標為 0-83 天，縱坐標為每一天這一分鐘處於使用手機時間段內的分佈。



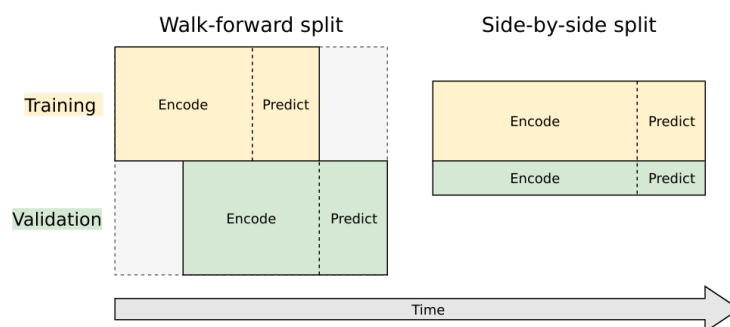
三、 Train 和 Validation 數據集劃分

不能只是將創建的數據流放入 keras 中。相反，必須設置一些數據轉換步驟來提取可以傳遞給 keras 的 numpy 數組。但在此之前，必須知道如何適當地將時間序列劃分為編碼和預測區間，以便進行訓練和驗證。注意，對於簡單的捲積模型不會使用 encoder-decoder 架構，但會保持“編碼”和“解碼”（預測）術語保持一致——在這種情況下，編碼時間間隔代表整個系列的歷史，將使用網絡學習的特性，但不輸出任何預測。

將使用一種風格的前向驗證，其中驗證集與訓練集的時間範圍相同，但是在時間上向前移動（在本例中是提前 60 天）。這樣，就可以模擬模型如何處理將來出現的不可見數據。

需要創建 4 個子段的數據：

1. Train encoding period
2. Train decoding period (train targets, 28 days)
3. Validation encoding period
4. Validation decoding period (validation targets, 28 days)



將通過為每個部分找到合適的開始和結束日期來做到這一點。從加載的數據的末尾開始，將反向工作以獲得驗證和訓練預測間隔。然後，將從一開始就向前工作，以獲得訓練和驗證編碼間隔。輸出結果如下：

```

Train encoding: 2019-06-01 00:00:00 - 2019-06-28 00:00:00
Train prediction: 2019-06-29 00:00:00 - 2019-07-26 00:00:00

Val encoding: 2019-06-01 00:00:00 - 2019-07-26 00:00:00
Val prediction: 2019-07-27 00:00:00 - 2019-08-23 00:00:00

Encoding interval: 28 days 00:00:00
Prediction interval: 28 days, 0:00:00
pred_28_encoding: 2019-06-01 00:00:00 - 2019-08-23 00:00:00
pred_28_Prediction: 2019-08-24 00:00:00 - 2019-09-21 00:00:00

```

四、 數據格式

現在有了時間段日期，將定義以 keras 友好格式提取數據所需的函數。以下是步驟。

將時間序列放入數組中，保存 date_to_index 映射作為引用數組的實用工具
創建函數從所有序列中提取指定的時間間隔
創建轉換所有系列的函數。

在這裡，通過取 loglp 和使用編碼器的序列均值來消除每個序列的意義，然後將其重塑為 keras 所期望的(n_series、n_timesteps、n_features)張量格式，從而使比例變得平滑。

注意，如果想要生成真實的預測而不是對數尺度的預測，可以很容易地在預測時應用反向轉換。

數據輸出格式如下：

```

2019-06-01    0
2019-06-02    1
2019-06-03    2
2019-06-04    3
2019-06-05    4
..
2019-08-19    79
2019-08-20    80
2019-08-21    81
2019-08-22    82
2019-08-23    83
Length: 84, dtype: int64
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]

```

五、 Full-Fledged WaveNet

這種卷積架構是 WaveNet 模型的一個成熟版本，是作為音頻的生成模型設計的。 wavenet 模型可以從音頻之外抽象出來，應用於任何時間序列預測問題，提供了一個很好的結構來捕獲長期依賴關係，而不需要過多的學習權重。

wavenet 模型的核心構建塊是擴展的因果卷積層。這種類型的捲積可以很好地處理時間流，並允許輸出的接受域隨著層數的增加呈指數增長。下面的波形網圖很好地展示了這種結構。

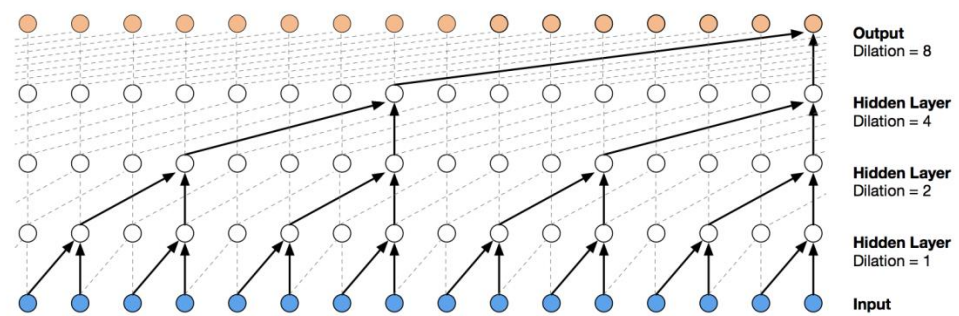


Figure 3: Visualization of a stack of *dilated* causal convolutional layers.

該模型還利用了其他一些關鍵技術：**gated activations**、**residual connections** 和 **skip connections**。將介紹並解釋這些技術，然後展示如何在 keras 中實現成熟的 WaveNet 體系結構。WaveNet 紙圖下面詳細說明逐區模型的組件組合在一起，變成一個堆棧的操作，將使用它作為一個方便的去參考（注意圖和實現之間有輕微的差異，如原始 WaveNet softmax 分類而不是回歸輸出）。

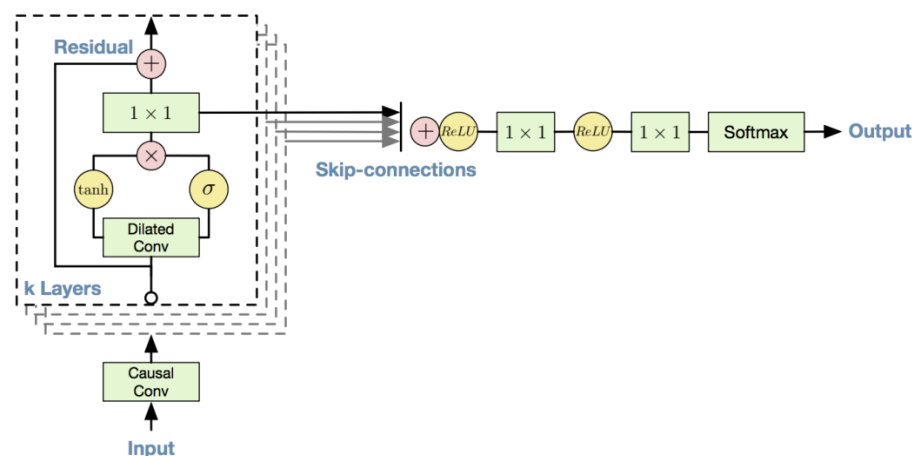


Figure 4: Overview of the residual block and the entire architecture.

現在與我們所有的組件，我們將使用：

- 16 擴張因果卷積模塊
 - ◆ 預處理和後處理(分佈式)完全連接層(與濾波器的捲積寬度 1):16 個輸出單位
 - ◆ 32 過濾器的寬度/塊
 - ◆ 成倍增加的膨脹率重置(1、2、4、8... ,128, 1, 2, ...,128)
 - ◆ 門控激活
 - ◆ 剩餘連接和跳過連接
- 2(時間分佈)全連接層，將跳過輸出和映射到最終輸出

將從輸出序列中提取最後的 28 個步驟作為訓練的預期輸出。將在訓練中再次使用教師強迫。與前面的類似，將有一個單獨的函數，它運行一個推理循環來生成對未知數據的預測，迭代地將以前的預測填充到歷史序列中。模型參數如下：

Model: "model_2"							
Layer (type)	Output Shape	Param #	Connected to				
input_3 (InputLayer)	[(None, None, 1)]	0					
conv1d_132 (Conv1D)	(None, None, 16)	32	input_3[0][0]				
conv1d_133 (Conv1D)	(None, None, 32)	1056	conv1d_132[0][0]				
conv1d_134 (Conv1D)	(None, None, 32)	1056	conv1d_132[0][0]				
activation_68 (Activation)	(None, None, 32)	0	conv1d_133[0][0]				
activation_69 (Activation)	(None, None, 32)	0	conv1d_134[0][0]				
multiply_32 (Multiply)	(None, None, 32)	0	activation_68[0][0] activation_69[0][0]				
conv1d_135 (Conv1D)	(None, None, 16)	528	multiply_32[0][0]				
add_34 (Add)	(None, None, 16)	0	conv1d_132[0][0] conv1d_135[0][0]				
conv1d_136 (Conv1D)	(None, None, 16)	272	add_34[0][0]				
conv1d_137 (Conv1D)	(None, None, 32)	1056	conv1d_136[0][0]				
conv1d_138 (Conv1D)	(None, None, 32)	1056	conv1d_136[0][0]				
activation_70 (Activation)	(None, None, 32)	0	conv1d_137[0][0]				
activation_71 (Activation)	(None, None, 32)	0	conv1d_138[0][0]				
multiply_33 (Multiply)	(None, None, 32)	0	activation_70[0][0] activation_71[0][0]				
conv1d_139 (Conv1D)	(None, None, 16)	528	multiply_33[0][0]				
add_35 (Add)	(None, None, 16)	0	conv1d_136[0][0] conv1d_139[0][0]				
				activation_98 (Activation) (None, None, 32) 0 conv1d_198[0][0]			
				activation_99 (Activation) (None, None, 32) 0 conv1d_194[0][0]			
				multiply_47 (Multiply) (None, None, 32) 0 activation_98[0][0] activation_99[0][0]			
				conv1d_195 (Conv1D) (None, None, 16) 528 multiply_47[0][0]			
				add_50 (Add) (None, None, 16) 0 conv1d_135[0][0] conv1d_195[0][0] conv1d_143[0][0] conv1d_147[0][0] conv1d_151[0][0] conv1d_155[0][0] conv1d_159[0][0] conv1d_163[0][0] conv1d_167[0][0] conv1d_171[0][0] conv1d_175[0][0] conv1d_179[0][0] conv1d_183[0][0] conv1d_187[0][0] conv1d_191[0][0] conv1d_195[0][0]			
				activation_100 (Activation) (None, None, 16) 0 add_50[0][0]			
				conv1d_196 (Conv1D) (None, None, 128) 2176 activation_100[0][0]			
				activation_101 (Activation) (None, None, 128) 0 conv1d_196[0][0]			
				dropout_2 (Dropout) (None, None, 128) 0 activation_101[0][0]			
				conv1d_197 (Conv1D) (None, None, 1) 129 dropout_2[0][0]			
				lambda_2 (Lambda) (None, None, 1) 0 conv1d_197[0][0]			
				Total params: 48,657			
				Trainable params: 48,657			
				Non-trainable params: 0			

定義了培訓體系結構後，就可以培訓模型了！將利用前面定義的 transformer 實用函數，並使用平均絕對錯誤損失進行培訓。需要訓練的總參數是使用簡單的 WaveNet 模型的兩倍多，可以解釋較慢的訓練時間（以及使用更多的訓練數據）。

六、訓練參數

訓練樣本數、驗證樣本數、超參數、學習率和 epoch 等的設置。

```
first_n_samples = 1440
batch_size = 8
epochs = 50
```

Train on 864 samples, validate on 576 samples

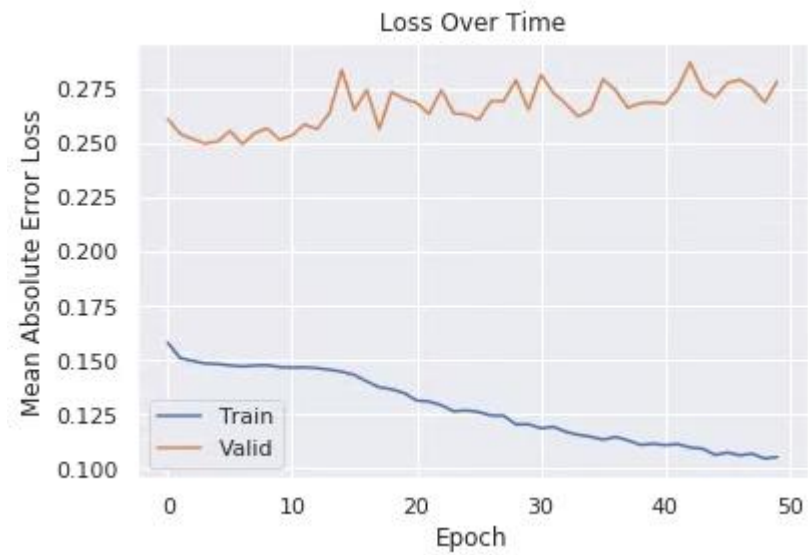
數據編碼前和編碼後：

[illegible]

[illegible]

七、訓練結果和預測

查看訓練/驗證損失的收斂曲線。



預測概率大於 0.2 的為 1，其他值為 0。

```
[ ] import math
import csv

file_name = '/content/drive/Shared drives/Me/TimeSeries_Seq2Seq/pre_28_11111.csv'
tmp2 = open(file_name, 'w', newline='', encoding='utf-8') # s表示在最后一行后面追加

csv_write = csv.writer(tmp2)
#pop = ['num','1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16','17','18','19','20','21','22','23','24','25','26','27','28']
#csv_write.writerow(pop)

for i in range(0,1440):
    pop = []
    pred_series = predict_and_plot1(encoder_input_data, sample_ind=i, enc_tail_len=100)
    pop.append(i)
    #print(pred_series)
    for k in range(0,28):
        #print(pred_series[0][k][0])
        if pred_series[0][k][0] >= 0.2:
            pop.append(1)
        else:
            pop.append(0)
            #pop.append(pred_series[0][k][0])
    csv_write.writerow(pop)
tmp2.close()
```

預測結果，列為天（1-84），行為分鐘（0-1144）：

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

开始

插入

页面布局

公式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

特色应用

帮助

文件

格式

数据

审阅

视图

安全

开发工具

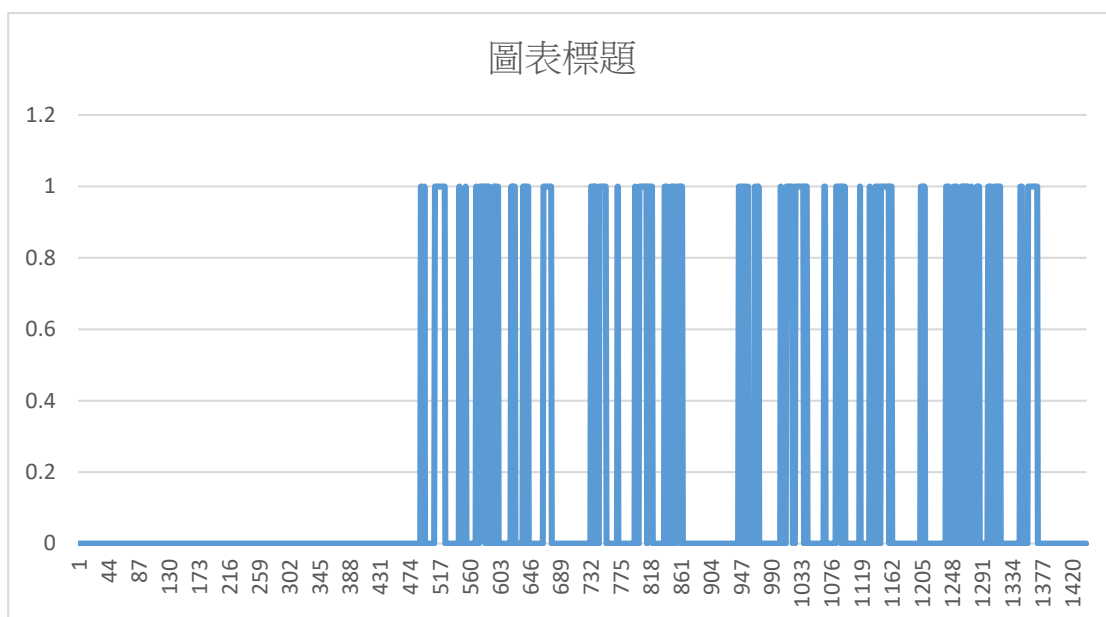
特色应用

帮助

文件

格式

取第 9 天數據展示。



肆、 結果與討論

此次藥物鬧鐘排程的結果如下

Hit rate	Annoyance rate	Error rate
0.7121	24.7857	0

由上表可以看到，Error rate 如我們所設計為 0，達到了使用者都有在正確時間內吃藥之目標。模型預測的結果也不錯，使 Hit rate 也有達到 0.7 以上。在 Annoyance rate 方面則較不理想，推測原因為「沒有使用手機時有 90% 機率產生下一個貪睡鬧鐘」，90% 的機率太高了。降低產生鬧鐘的機率或許可以減少 Annoyance rate。

由於本次專題是以 error rate = 0 為先決條件，剩下的兩個評斷標準若要增加其中一個條件之效能，勢必將一定程度地降低另一個條件的表現。基本上要讓 hit rate 有所增加，除了機率性提高貪睡鬧鐘的次數，就是把預測的模型做修改。另外，「訓練結果與預測」中，訓練/驗證損失的收斂曲線圖上的驗證損失曲線是緩速上升的趨勢，這表示該模型還有改善的空間。

以結果而言，我們認為該預測模型與排程演算法是目前的最佳效能，若能另外找到更好的模型或是較優的排程邏輯思考，將有助於提升預測結果的效益。