

1. 請簡明扼要地闡述你如何抽取模型的輸入特徵 (feature)

答：

載入 train.csv 後，將裏頭的數字 data 每 18 個欄位後抓進一個 list，因此這個 list 的 18 個 rows 分別代表某個 feature。之後再將每連續 9 個小時的所有抓進另一個 train\_x 的 list，第十個小時則放進 train\_y 的 list。因此每個月連續 480 小時的 data 中共取了 471 比 18 種 feature 的資料。

2. 請作圖比較不同訓練資料量對於 PM2.5 預測準確率的影響

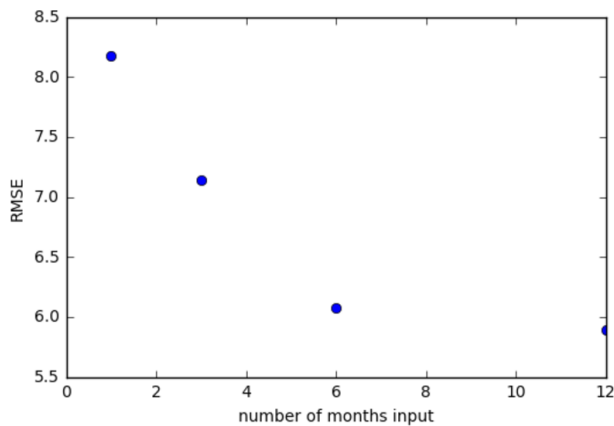
答：

將訓練量減半為前六個月後：Kaggle 上的 RMSE = 6.08049

將訓練量再次減半剩下三個月：Kaggle 上的 RMSE = 7.13567

若訓練量只有一個月：Kaggle 上的 RMSE = 8.17345

以下為作圖結果



```
import numpy as np
import matplotlib.pyplot as plt
x = np.array([1, 3, 6, 12])
y = np.array([8.17345, 7.13567, 6.08049, 5.8886])
plt.plot(x,y,'o')
plt.xlabel("number of months input")
plt.ylabel("RMSE")
plt.show()
```

3. 請比較不同複雜度的模型對於 PM2.5 預測準確率的影響

答：

一開始使用抽取所有的 feature 來當作參數，在 iteration 60000 次的訓練後，可在 kaggle 上得到 5.886 的 RMSE。若將抓進來的所有 feature 進行二次方，經由 60000 次的訓練後 kaggle 上的 RMSE 也只有 5.895，兩者並沒有顯著差異。

#### 4. 請討論正規化(regularization)對於 PM2.5 預測準確率的影響

答：

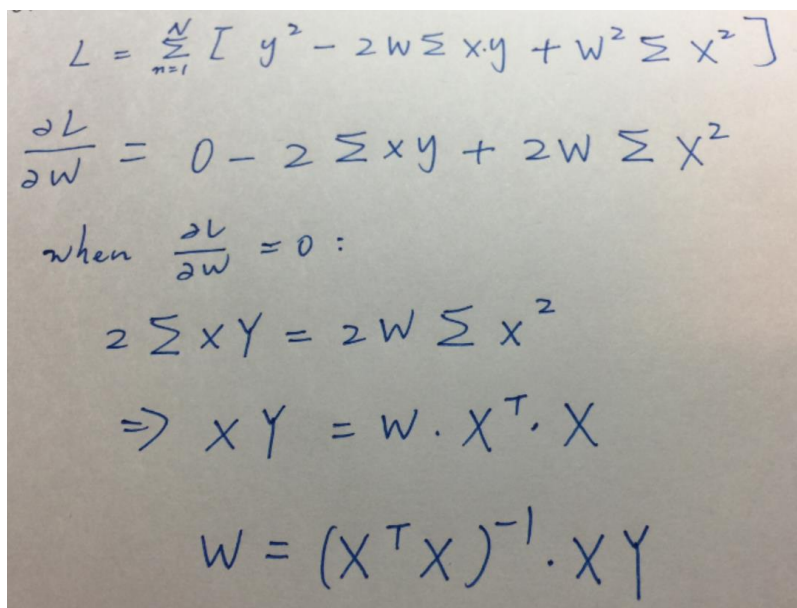
為了防止我們的loss function 過度overfitting，我們在loss function上再添入了一項與 weight有關的項當作penalty(如下)，防止weight太高而造成的overfitting。

$$\sum (w \cdot x_i^T - y_i)^2 + \frac{1}{2} \lambda \|w\|^2$$

然而嘗試 3 次正規化 lambda 分別為 1、10、100 在 kaggle 上的 public score 大約都落在 5.893 左右，與沒有進行 regularization 的情況相差不大。但有進行 regularization 的那組 data 在 private score 剛好超過 strong baseline，推測較小的 overfitting 結果可能就會在這部分顯現出些微的差異。

5. 在線性回歸問題中，假設有 N 筆訓練資料，每筆訓練資料的特徵 (feature) 為一向量  $x^n$ ，其標註(label)為一存量  $y^n$ ，模型參數為一向量  $w$  (此處忽略偏權值  $b$ )，則線性回歸的損失函數(loss function)為  $\sum_{n=1}^N (y^n - w \cdot x^n)^2$ 。若將所有訓練資料的特徵值以矩陣  $X = [x^1 x^2 \dots x^N]$  表示，所有訓練資料的標註以向量  $y = [y^1 y^2 \dots y^N]^T$  表示，請以  $X$  和  $y$  表示可以最小化損失函數的向量  $w$ 。

答：推導過程如下



Handwritten derivation of the linear regression formula:

$$L = \sum_{n=1}^N [y^2 - 2w \cdot x \cdot y + w^2 \cdot x^2]$$
$$\frac{\partial L}{\partial w} = 0 - 2 \sum x y + 2w \sum x^2$$

when  $\frac{\partial L}{\partial w} = 0$ :

$$2 \sum x y = 2w \sum x^2$$
$$\Rightarrow X Y = w \cdot X^T \cdot X$$
$$w = (X^T X)^{-1} \cdot X Y$$

$$W = (X^T X)^{-1} X^T y$$

助教，辛苦了!!!