Preprocess & Feature Engineering - RNN

我嘗試過很多preprocessing & feature的組合，方式分類不外乎以下幾種：

1. 全部 feature V.S 部分 feature

   若僅取部分feature其選法不外乎計算feature跟label的相關係數，僅取前 n 高的feature拿去訓練。後來我們有參考Driven Data上的Benchmark (ref [1])，它僅取4個Feature 就可以有很不錯的結果，以下是4個feature的名稱。

   - `reanalysis_specific_humidity_g_per_kg`
   - `reanalysis_dew_point_temp_k`
   - `station_avg_temp_c`
   - `station_min_temp_c`

2. 加入平方 nor not?

   針對既有的feature在標準化之前取平方，結合原本1次方的feature一起去訓練。

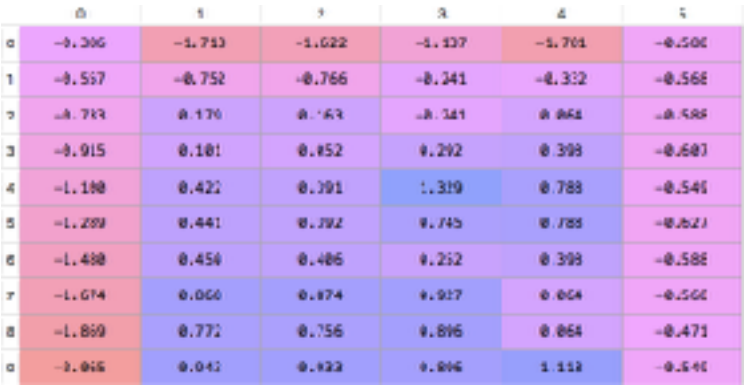
3. 加入 Label 當作 feature ？

   需不需要加入前幾週的label當作feature？(這在experience & discussion 那段有詳細說明)

4. 週數Cosine 化?

   第52週跟第1週的天氣狀態應該是相似的，但對於machine來說他們僅只是不同的數字，把週數丟入cosine (or sine) function可以讓第52週跟第1週的值相似，並維持週數的週期性。

我最後採用的組合是 4個 benchmark的feature + 前幾週的Label + 週數Cosine化，它可以給我最好的預測結果。(右圖是一組RNN的feature)



Cosine 週數 **4 features from benchmark** **前幾週的 Label**

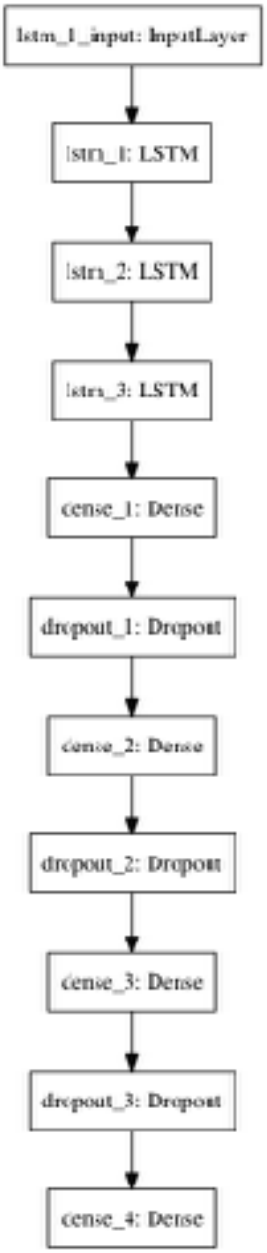
Model Description

動機

DNN上傳之後的結果MAE僅介於26~28之間，這讓我們不得不想其他方法來改善訓練成效，我們不禁想，登革熱案件的高低難道只會跟當週的 Feature 相關嗎？如果答案是否定的，那把前面幾週的資料也一起放進去預測是否會比較好? 在有這個疑問之後，我們就上網去找登革熱的Domain Knowledge(ref [2],[3])，理想的溫溼度條件的當週會促成病媒蚊的大量繁殖，但從卵到成蟲再到大量肆虐又會經過 5 ~ 10週的潛伏期。在這樣的Domain Knowledge之下，我們決定用RNN做嘗試。

架構

我們嘗試了很多LSTM的架構，左下是固定的參數跟調變過的參數列表，右邊則是在三層 Dense Layer 跟三層 RNN Layer 下的架構圖。



固定參數

參數名稱	數值
Batch Size	32
Optimization	Adam
RNN Activation	tanh
Dense Activation	Relu
Output Activation	Sigmoid

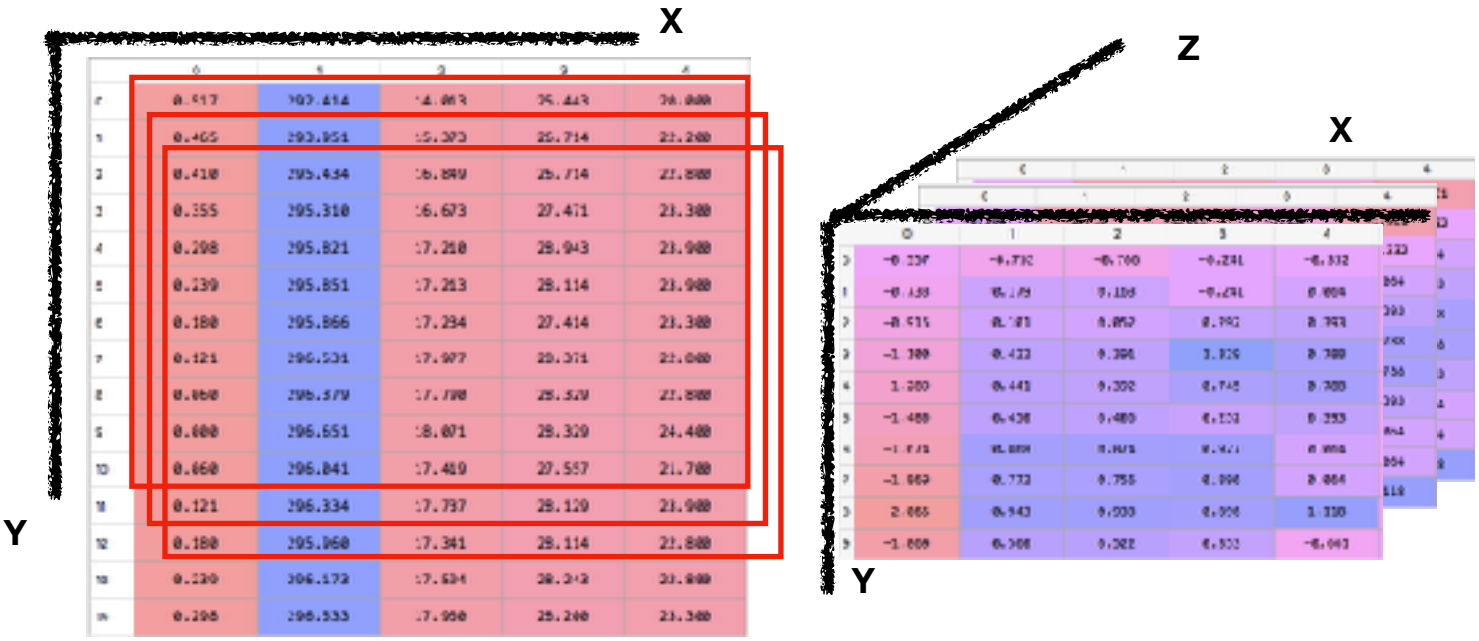
調變參數

參數名稱	調變過的數值
Dense Layer數	1,2,3
RNN Layer 數	1,2,3
Dropout	0.4,0.5,0.6,0.7
連續預測的週數	**5, 8, 9, 10, 11, 12**

Data Processing for RNN

為了讓資料放入RNN的架構，我們必須讓原本二維的 training Set 變成三維，如下圖所示，左邊是原始的 Training Feature，右邊是修改過要放入RNN的 Training Feature。假設要取連續10週的資料，那我們就會以左圖紅色框框為一個Set，按第1-10週、2-11週....依序

取值，並對每個Set做標準化。最後，再把這些Set在Z軸依序並排成一個三維陣列，放入RNN。



Experiment & Discussion

- **RNN 跟 DNN 的比較**

無論是DNN 還是 RNN，我們都有設定Early Stopping 跟 ModelCheckPoint的機制。相比之下，DNN Loss最小的Model通常都出現在前10個Epoch，而RNN Loss最小的Model平均落在第 50~60 個 Model左右，Training 途中 best model 的更新次數明顯比 DNN 多，自然也比 DNN 更晚了才Early Stopping，在這樣的結果下，DNN在SJ城市的 Validation Loss 大概落在 20 ~ 23，而RNN的 Loss 可落在15 ~ 17。

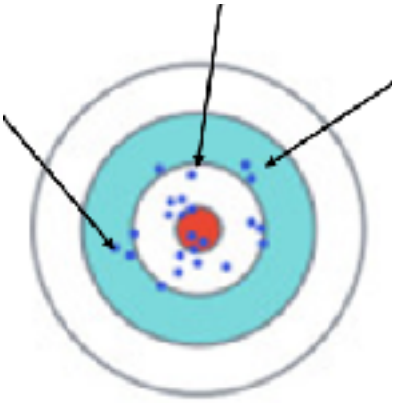
原本預期RNN的訓練成效會比DNN好很多，然而事與願違，上傳之後的MAE居然才落在25.5 ~ 27之間，RNN成效是比DNN好沒有錯，但改善還不夠讓我們破 Strong Baseline。

- **RNN 的改良：以 Label 作為Feature**

如果說前幾週的 Feature 會影響當週的登革熱Case數，那前面幾週的Case應該也會跟當週的Case相關吧！根據這樣的推測，我把Label也加到 Training Set 裡面，也就是說，我用前n週的 feature跟 label 一起去預測第n+1週的 label。如右圖所示，最右列的就是加入的 Feature。

此訓練的成效都比原始RNN的好，best model 的更新次數更多，SJ城市的 Validation Loss可降到 9 ~ 10左右，IQ城市的 Loss < 1，上傳的MAE落在 24 ~ 25左右。

- **RNN 的改量 : 不同Model 的 Ensemble**



我調過不同的參數 (參見”調變參數”列表)，基本上連續預測的週數只要介於5 ~ 12之間，最好model的 loss 基本上都不會差太多。即便有幾組參數 (ex: dense layer數 = 1, LSTM = 2)，的 validation loss能比其他組合低，但上傳之後都無顯著的差異。在這樣的狀況下，我直接嘗試用不同參數訓練出來的Model去做Ensemble，其中有一組成功突破Strong Baseline，loss 達到22.77，它的參數跟取的feature如下。

	Feature 有沒有加上平方項	# of Dense	# of RNN	連續預測週數
Model 1	no	1	1	10
Model 2	yes	2	2	10
Model 3	yes	2	2	12
Model 4	no	2	3	10

Reference

[1] http://blog.drivendata.org/2016/12/23/dengue-benchmark/

[2] http://61.57.41.133/uploads/files/b549879d-9614-4927-a751-0c9ddfbee8c1.pdf (衛生福利部疾病管制署)

[3] http://www.epa.gov.tw/fp.asp?fpage=cp&xItem=33354&ctNode=31507&mp=epa