

RNN

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，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

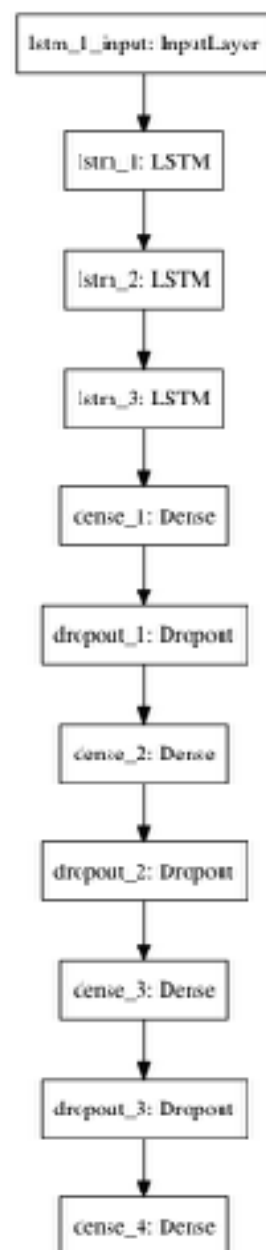
我們嘗試了很多LSTM的架構，左下是固定的參數跟調變過的參數列表，右邊則是在三層 Dense Layer 跟三層 RNN Layer 下的架構圖。

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

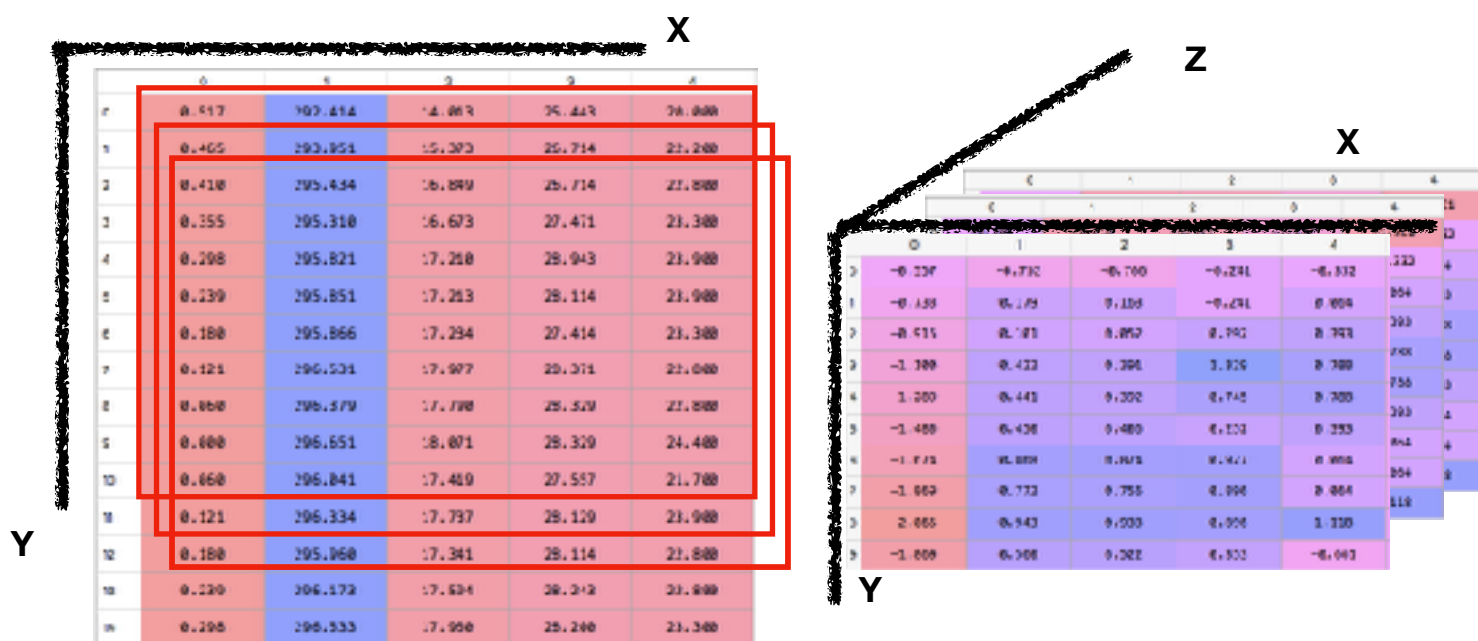
參數名稱	調變過的數值
Dense Layer數	1,2,3
RNN Layer 數	1,2,3
Dropout	0.4,0.5,0.6,0.7
連續預測的週數	5, 8, 9, 10, 11, 12



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

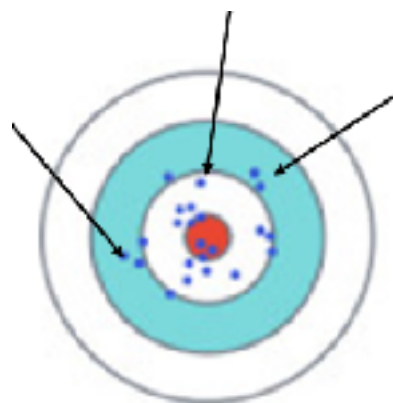
如果說前幾週的 Feature 會影響當週的登革熱Case數，那前面幾週的Case應該也會跟當週的Case相關吧！根據這樣的推測，我把Label 也加到 Training Set 裡面，也就是說，我用前n週的 feature跟 label 一起去預測第n+1週的 label。如右圖所示，最右列的就是加入的 Feature。

	0	1	2	3	4	5
0	-0.305	-1.780	-1.022	-1.137	-1.770	-0.500
1	-0.337	-0.332	-0.305	-0.241	-0.332	-0.366
2	-0.144	0.119	0.184	-0.081	0.084	-0.180
3	-0.935	0.191	0.862	0.892	0.262	-0.697
4	-1.100	0.422	0.391	1.325	-0.700	-0.549
5	-1.209	0.441	0.592	0.745	0.765	-0.507
6	-1.250	0.258	0.286	0.942	0.707	-0.588
7	-1.674	0.868	0.274	0.921	0.844	-0.568
8	-1.809	0.772	0.736	0.596	0.864	-0.471
9	-2.085	0.581	0.851	0.598	1.118	-0.549

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 的改良：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