基於 LSTM 每月入境人口預測

吳定軒

109318095

roy87816@yahoo.com.tw

施昀楷

109318044

ian860916@gmail.com

台北科技大學電機工程系碩一 412 研究室

第十二組

***Abstract*—使用 LSTM 倚靠前六個月的入境人數來預測下個月的入境人數，在疫情嚴峻的情況下，可藉由預測的數值作為參考，判斷未來醫療資源是否足以充足。**

***Keywords*—LSTM、Batch Normalization、Early Stopping**

I. INTRODUCTION (*HEADING 1*)

現今各國的疫情不斷地升溫，台灣、菲律賓、印度…等國家，國內迎來了第二波疫情的爆發，在這樣嚴峻的條件下，醫療資源已十分緊張，更何況這僅僅是國內的疫情，就造成了醫療資源的匱乏，加上每個月從海外歸來的國人、航空人員、遊客…等等，在資源匱乏的情況下，如果不事先預測下個月有多少入境人數，以便分配資療資源的話，疫情將會再度的失控。

II. PROBLEM STATEMENT

時至今日台灣本土疫情持續升溫，每天確診人數更是居高不下，台灣的醫療資源早已不夠來應付這樣的疫情環境，報導上可以看到許多民眾在急診室外吊著點滴，在家遲遲等不到救護車及防疫旅館，本土的疫情已經這麼嚴重了，每個月還有許多的入境民眾，台灣的醫療資源遲早有一天會面臨崩潰，所以在醫療資源還沒崩潰之前，由於本土疫情無法預測，所以我們不如先做好入境人數的預測，倚靠前六個月的入境人數來預測下個月的入境人數，並且藉由此分配醫療資源(防疫旅館數目)。

III. DATA PROCESSING

A. *訓練及驗證資料*

為了準確的預測，需要一個可信度高且資料量齊全的資料來源，所以我們使用了交通部來台旅客人數(按照居住地區分)[1]，其中有分為總計、亞洲地區、香港、澳門、中國大陸…等等，如圖 1 所示。

	總計	亞洲地區	香港、澳門	中國大陸	日本	南韓	馬來西亞	新加坡	印尼
90年	2,831,035	2,224,356	435,164	211,050	976,750	85,744	57,615	98,771	89,921
90年 1月	212,103	165,137	27,638	18,217	72,705	6,143	4,737	5,825	6,640
90年 2月	246,509	200,471	27,755	15,890	102,187	7,233	5,680	7,196	7,774
90年 3月	266,176	211,277	38,222	18,480	102,385	7,719	4,257	8,106	7,263
90年 4月	247,999	190,818	40,011	17,628	79,962	6,920	4,452	7,986	8,146
90年 5月	240,696	188,003	30,193	17,132	84,574	6,741	5,293	11,383	7,957
90年 6月	253,597	193,325	40,803	17,638	79,456	8,860	4,843	8,540	9,007
90年 7月	232,608	181,532	32,562	17,984	84,373	7,811	3,793	7,207	7,654

圖 1. 交通部來台旅客人數統計

我們採用每個月來台的總計人數進行預測，並且使用前六個月來預測後一個月，即 Many to One，其原因在於旅遊業通常都區分為上下半季，且由於旅遊業與政策有關，像是疫情的關係，許多航班都會取消，理所當然的入境人數就會下降，所以不宜將 timestep 調整過大，導致誤差太大。

再來我們使用 Sklearn 套件中的 MinMaxScaler 函式對輸入數據進行處理，MinMaxScaler 會將每個輸入樣本 X 減去所有 X 中最小的數值，再除以所有 X 中最大的數值減去 X 中最小的數值，如式 1。

$$MinMaxScaler = \frac{X - X_{min}}{X_{max} - X_{min}} \epsilon\ [0,1] \qquad (1)$$

而訓練資料的部分我們是採用民國九十年一月至民國一百零五年十二月，並且將其拆分為訓練資料(80%)及驗證資料(20%)。

B. *測試資料*

測試資料我們採用民國一百零六年一月至民國一百一十年三月共五十一筆資料。

C. *Groundtruth*

我們將 Ground truth(第七個月開始的資料)進行標準化處理，如圖 2 所示。

90年 7月	232,608	181,532	32,562	17,984	84,373	7,811	3,793	7,207	7,654
90年 8月	240,704	194,662	41,229	20,279	85,968	7,630	3,890	7,028	7,618
90年 9月	216,500	176,904	38,500	15,749	76,455	6,796	5,074	7,320	7,184
90年 10月	213,603	166,388	36,309	16,452	67,125	6,611	3,949	7,245	7,633
90年 11月	217,828	168,588	36,796	17,644	68,589	6,524	5,015	9,183	6,013
90年 12月	242,712	187,251	45,146	17,957	72,971	6,756	6,632	11,752	7,032

圖 2.Ground Truth 示意圖

有趣的是我們在訓練資料、驗證資料及測試資料皆有將 Ground truth 進行 MinMaxScaler 處理，其原因在於我們起初先將訓練資料的輸入標準化之後，與訓練資料的 Ground truth 標準化進行 Loss 的對比，發現 Loss 非常的微小將近 0.01 左右，但是我們在進行測試資料時，發現我們將輸出結果進行尺度復原時，雖然整體的趨勢是相同的，但與實際的數值落差許多，我們推測原因為我們進行尺度復原時的尺度是原本訓練集 Ground truth 的參數，但並不一定適合測試集，所以經過我們的討論，我們決定對測試集的 Ground truth 也進行標準化，以利於我們對測試集的預測結果進行尺度復原。

Identify applicable funding agency here. If none, delete this text box.

IV. NEUEAL NETWORKS STRUCTURE AND TECHNICAL DETAILS

A. *神經網路架構*

整體模型如圖 3 所示，第一層 LSTM 為 100 個神經元，第二層 LSTM 為 10 個神經元，由於輸出為後一個月的入境人數，因此加入了神經元為 1 的全連接層，為了防止梯度消失或梯度爆炸，我們在每一層後面我們都加上了 BN 層，模型的總參數量為 45,695，訓練時使用 Adam 作為優化器，Loss 則以 MSE 進行計算，並使用了早停技術。

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_14 (LSTM)               (None, 6, 100)            40800
_________________________________________________________________
batch_normalization_19 (Batc (None, 6, 100)            400
_________________________________________________________________
lstm_15 (LSTM)               (None, 10)                4440
_________________________________________________________________
batch_normalization_20 (Batc (None, 10)                40
_________________________________________________________________
dense_7 (Dense)              (None, 1)                 11
_________________________________________________________________
batch_normalization_21 (Batc (None, 1)                 4
=================================================================
Total params: 45,695
Trainable params: 45,473
Non-trainable params: 222
```

圖 3.模型架構圖

B. *早停技術*

當我們訓練深度學習神經網絡的時候通常希望能獲得最好的性能，但過度擬合訓練資料會使得模型無法有效應用在未知資料，且當模型趨近穩定後，會發現即使繼續訓練，模型效果也不會繼續進步，只是白白浪費訓練時間罷了，為了因應上述兩種情形，我們試著在訓練時使用早停技術，在下個章節我們將會討論設置不同的早停週期對於模型的整體影響。

C. *Dropout*

Dropout 是一種正則化方法，透過隨機停用一些神經元，能夠有效地防止神經元之間的相互協調，對於一般的前饋神經網路，Dropout 可以有效避免過擬合，但對於 LSTM 這種重視先前記憶的網路，使用 Dropout 反而會破壞其效果[2]。起初我們也嘗試加入 Dropout 層，但效果並不是很好，Loss 不減反增，在下個章節會有更詳細的結果討論。

D. *Batch Normalization*

由於 Dropout 成效不佳，因此我們嘗試了另一個正則化方法－Batch Normalization。它可以防止梯度消失或梯度爆炸，也可稍微解決過擬合的情形，雖然在查詢資料時有看到學者認為 RNN 這種網路並不適合加入 BN 層[3]，但根據我們實驗的結果，使用 BN 層可提升我們模型的效能，在下個章節也會有更詳細的結果討論。

V. EXPERIMENTS

在附檔中共含十份檔案，分別為 source1.ipynb、data1.txt、data2.txt、data1.csv、data2.csv 以及 requirement.txt 等，表 1 為各個檔案之簡介。

表 1.各檔案簡介

檔名	內容
source1.ipynb	主程式
data1.txt/ data1.csv	訓練資料
data2.txt/ data2.csv	測試資料
requirement.txt	版本需求

A. *訓練/驗證資料的準確度及損失結果*

我們訓練資料的數量總共有 192 筆，從民國九十年一月至民國一百零五年十二月每個月的出入境人數， 其中我們又將其拆分為訓練資料(80%)及驗證資料(20%)，我們最終使用 Batch_Size 設為 12，Epoch 設為 1000、早停週期設為 200 的模型，由圖 4 可以看見不管是在訓練集或是驗證集上都表現出良好的成果，而從表 2 可以看見，訓練資料的 Loss 為 0.0016，驗證資料的 Loss 為 0.0018，皆能獲得極小誤差。

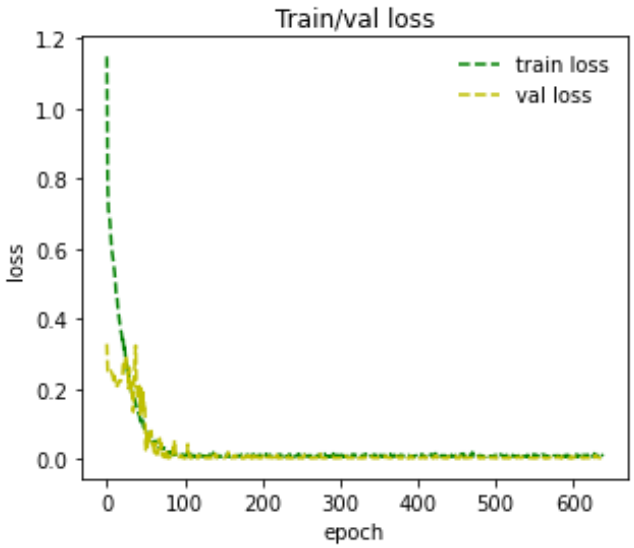

圖 4.Train/Val 之 Loss 圖

表 2.模型之 Train/Val/Test 之 Loss

Train Loss	Val Loss	Test Loss
0.0016	0.0018	0.0076

B. *測試資料*

測試資料總共為五十一筆，從民國一百零六年一月至民國一百一十年三月，從表 2 所知，我們的模型在測試資料的 Loss 也僅有 0.0076，並且從圖 5 可以看見我們模型不僅預測的趨勢與實際趨勢符合，且預測數值及實際數值有很高的相似性。

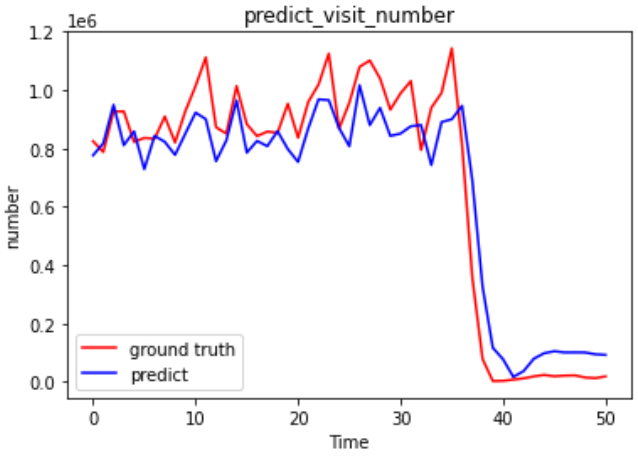

圖 5.預測結果圖

我們發現圖 5 上預測結果與真實結果落差較大的部分幾乎都落在十二月，舉個例子，如表 3 所示，一般的月份如民國 107 年 6 月在預測與實際上大約只差了 6%，但在民國 107 年 12 月的預測與實際落差竟然高達了 15%，我們認為這是因為 12 月時逢跨年，許多海外的朋友們都會來台跨年，由於這個原因，造成了我們模型在預測十二月這個月份上會有較大的誤差。

表 3.不同月份之預測人數及實際人數

年/月份	預測人數	實際人數
107/06	807,681	857,575
107/12	965,484	1,125,112

在測試的後半段，由於疫情的緣故，從民國 109 年 1 月至 110 年 3 月，入境人數急遽的下降，而我們的模型也有預測到這樣的情形發生。

C. 早停技術

我們比較了不同的早停週期｛50,100,200｝對於模型的執行時間、Epoch 次數、Train Loss、Val Loss 、Test Loss，如表 4 所示。

表 4.早停技術之比較

Method	執行時間	Epoch 次數	Train Loss	Val Loss	Test Loss
w/o 早停	35s	1000	0.0004	0.0013	0.019
50	6s	247	0.0024	0.00204	0.021
100	10s	174	0.0032	0.00205	0.013
200	20s	623	0.0016	0.0018	0.0076

從表 4 可以發現，早停週期越大執行時間越久，但在 Epoch 的次數上並沒有直接的關係，因為每一次的訓練起始的數值都不大相同，而如果早停週期調整的太大，其實就跟沒有使用早停週期差不多，後來經過我們的討論以時間為基準，Train Loss 及 Val Loss 表現不會太差下，我們選擇了早停週期為 200 的模型，而在最後的 Test Loss 表現上，也比其他來的要好。

D. Dropout

上個章節中有提到，我們使用了 Dropout 後的表現不增反減，如表 5 所示，比例分別設為｛0.2,0.5,0.8｝後獲得的 Loss 為｛0.0118,0.065,0.1562｝，我們從結果來推論，Dropout 似乎確實會破壞 LSTM 的記憶能力，從圖 6、圖 7、圖 8 來看，會發現有些模型雖然預測數據趨勢正確，但誤差偏大，而有些模型甚至連預測趨勢都無法做到，因此我們認為這個模型並不適合使用 Dropout。

表 5.Dropout 對模型之影響

	Train Loss
Model	0.0016
Model+Dropout(0.2)	0.0118
Model+Dropout(0.5)	0.065
Model+Dropout(0.8)	0.1562

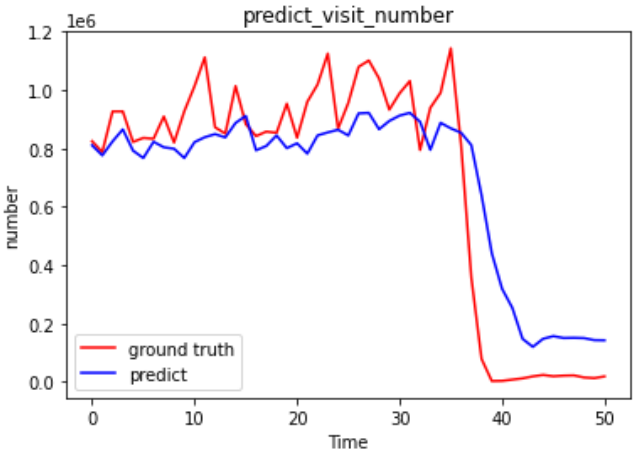

圖 6. 使用 Dropout(0.2)之結果圖

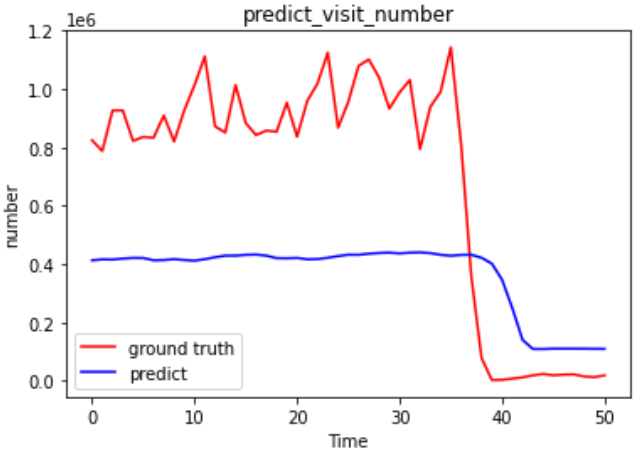

圖 7.使用 Dropout(0.5)之結果圖

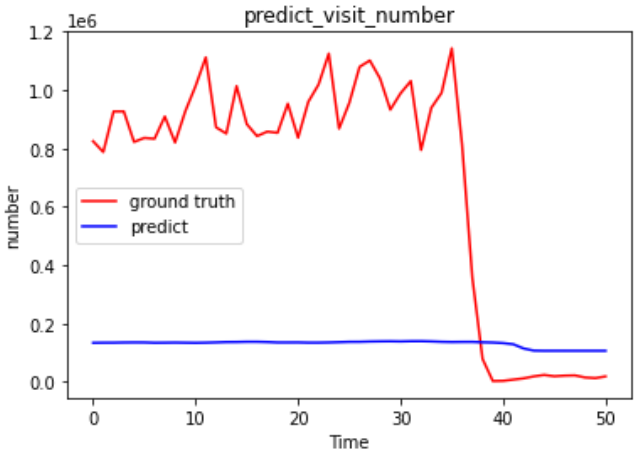

圖 8.使用 Dropout(0.8)之結果圖

並且從表 6 可以發現，不管是 Train Loss、Val Loss 或 Test Loss 都大幅提升，而這並不是我們所樂見的。

表 6.不同 Dropout 對 Train/Val/Test 之影響

Method	Train Loss	Val Loss	Test Loss
Model+Dropout(0.2)	0.0118	0.058	0.095
Model+Dropout(0.5)	0.065	0.14	0.255
Model+Dropout(0.8)	0.1562	0.166	0.382

E. Batch Normalization

我們同時分析了 BN 層對於整體模型的影響，試著在每一層後面都加上了 BN 層，表 7 比較了使用 BN 層及不使用 BN 層的 Train Loss、Val Loss、Test Loss，也能從圖 9 中看出不使用 BN 層後的結果對於後期入境人數大幅降低時的預測會失準，雖然學者普遍認為使用 BN 層會使得效能降低，但經過測試的結果，使用 BN 層之後確實是能提升效能，我們推測是因為後期變化太大，因此需要使用 BN 層，但是否為實際的原因我們並不清楚。

表 7.BN 對模型之影響

Method	Train Loss	Val Loss	Test Loss
w/o BN	0.0012	0.0025	0.023
w/ BN	0.0016	0.0018	0.0076

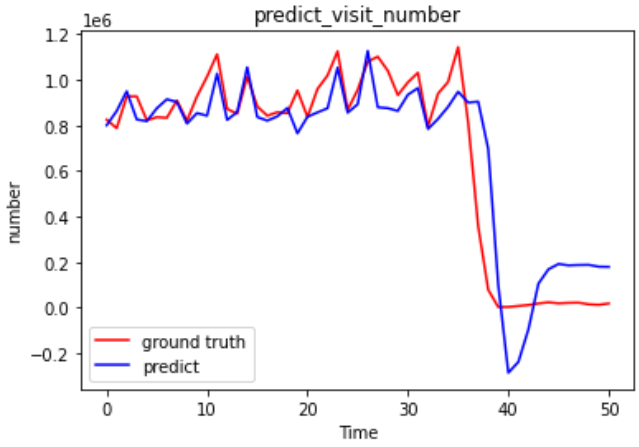

圖 9.不使用 BN 層之結果圖

F. Optimizer

我們試著使用不同的優化器，從表 8 發現其實在 Train loss、Val Loss 的表現上是差不多的，但 SGD 及 Adagrad 相較於其他兩個方法表現較差，所以我們就不多加考慮。

表 8.不同優化器之比較

Method	Train Loss	Val Loss	Test Loss
SGD	0.0035	0.0025	0.04
Adam	0.0016	0.0018	0.0076
Adagrad	0.0035	0.0026	0.455
RMSprop	0.0015	0.0017	0.02

接下來我們對於 Adam 及 RMSprop 進行三次訓練，來比較需花費多少 Epoch，如表 9 所示，早停週期皆為 200，我們可以發現 Adam 的 Epoch 都是小於 RMSprop 的，綜合評估各個優化器的運算時間以及效能表現，最終我們採用 Adam 作為優化器。

表 9.不同優化器所需 Epoch 的比較

Method	Train-1	Train-2	Train-3
Adam	350	657	588
RMSprop	490	1000	882

VI. CONCLUTION

在這次研究中，我們使用 LSTM 對入境人口進行預測，並且使用了各式各樣的小技巧去改進我們的模型，同時對於各個小技巧，我們對其獨立地進行實驗，看看這些技巧是否對模型有益亦或是有害於模型，最終我們獲得了極為優異的成果，未來若是能夠將此模型運用在政府機關，也許能成為政府對醫療資源進行補助的一種參考，為消滅疫情盡一份心力。

VII. REFERENCE

[1] https://stat.motc.gov.tw/mocdb/stmain.jsp?sys=100
[2] https://lonepatient.top/2018/09/24/a-review-of-dropout-as-applied-to-rnns.html
[3] https://www.zhihu.com/question/308310065