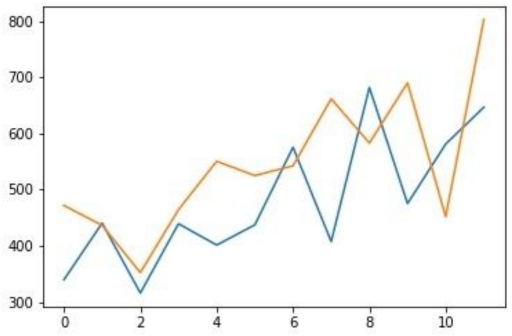
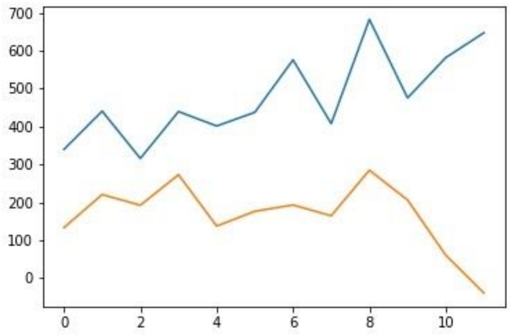
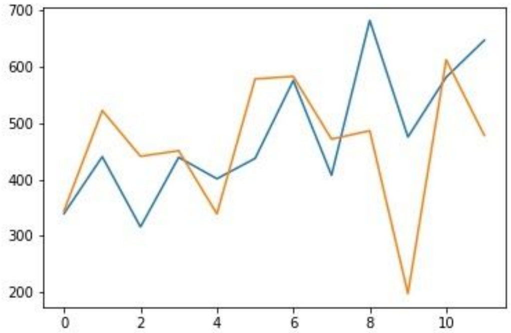
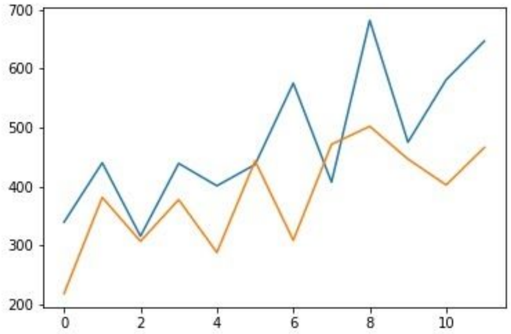
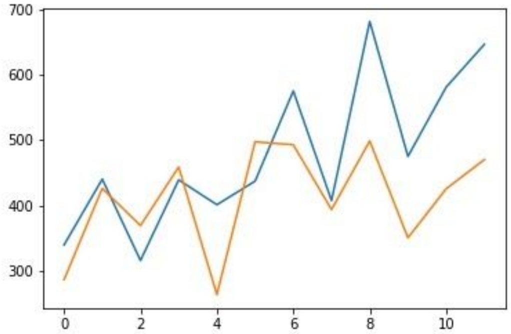
課堂活動 LSTM (2020/0602)

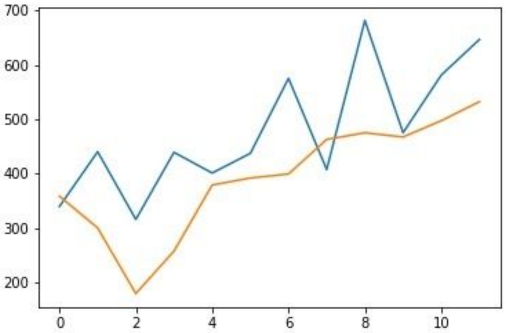
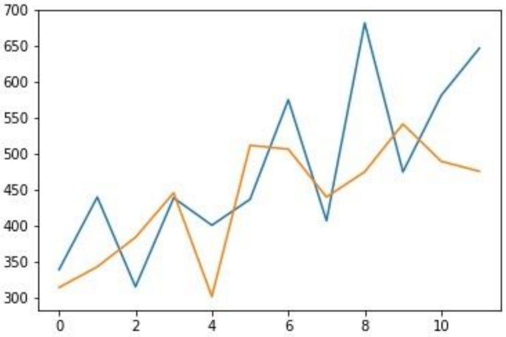
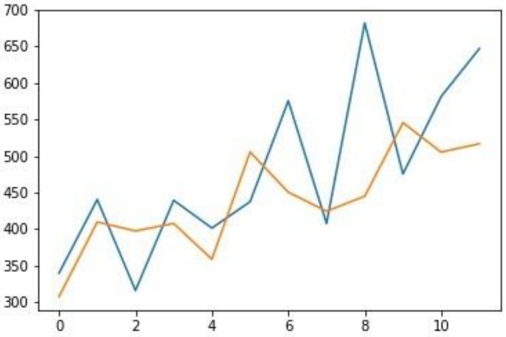
曾宏鈞 06160485
徐友笙 05360365
蔡毓丞 06370136
盧君彥 05360153

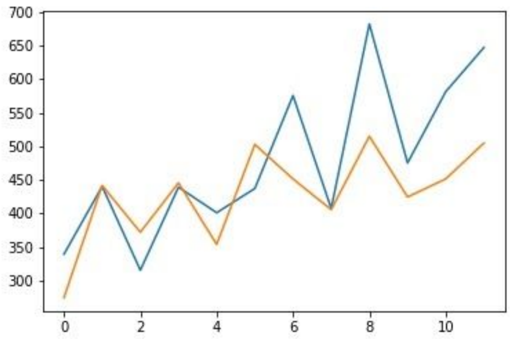
1. 請修改Example1程式，用LSTM架構, 進行月銷售額預測。

batch_size:1、epoch:3000

Neurons:4 RMSE: 125.33	
Neurons:8 RMSE: 76.61	

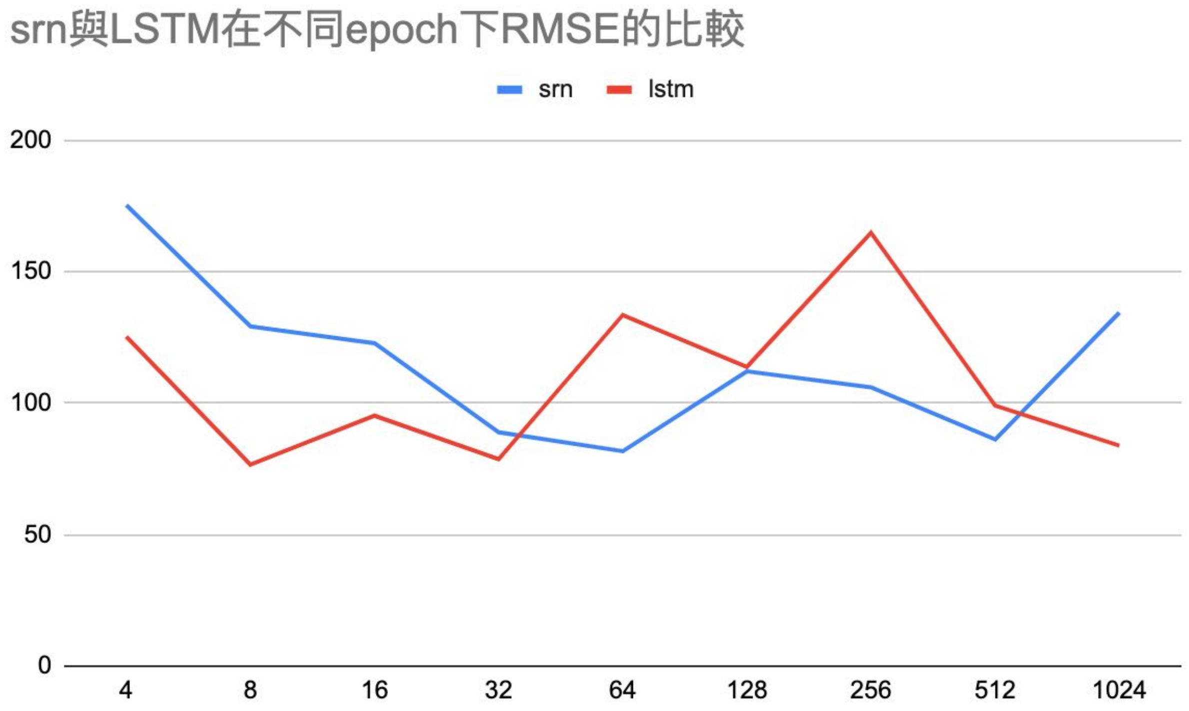
Neurons:16 RMSE: 95.25	
Neurons:32 RMSE: 78.7	
Neurons:64 RMSE: 133.55	

Neurons:128 RMSE: 113.74	
Neurons:256 RMSE: 164.84	
Neurons:512 RMSE: 99.04	

Neurons:1024 RMSE: 83.82	

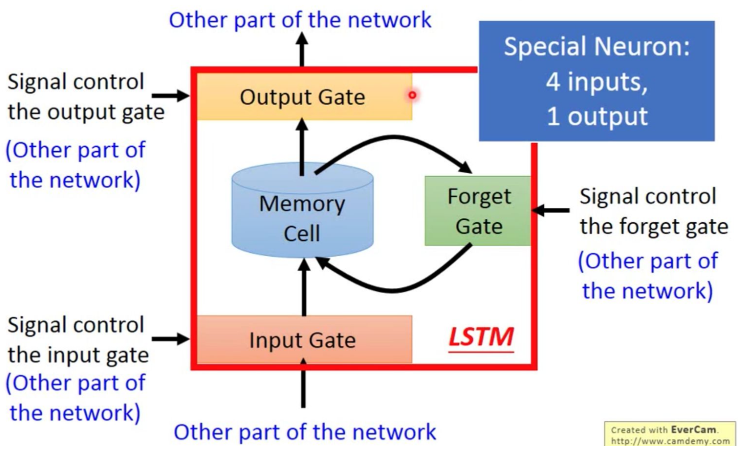
(縱軸)藍色為實際值，橘色為預測值、(橫軸)時間(月)

我們比較單一RNN網路(srn)及長短期記憶單元的RMSE可發現，LSTM在epoch數為4-32及1024的表現比較好。



縱軸為RMSE，橫軸為epoch數

心得

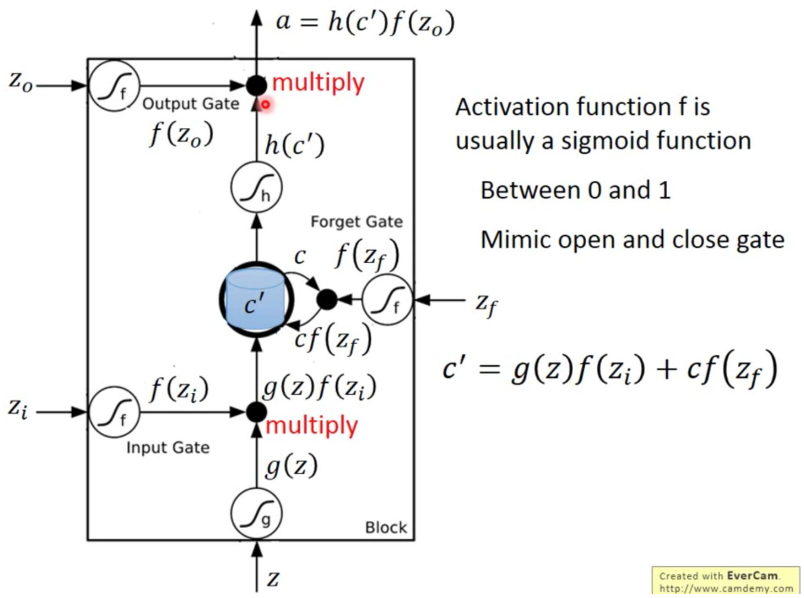


Input: 4個藍色箭頭

- 想被存進 **memory cell** 的值
- 控制 **input gate** 的訊號
- 控制 **forget gate** 的訊號
- 控制 **output gate** 的訊號

Output：other part of the network

LSTM，由三個不同的Gate組成，分別是input、Output、及forget，每一個Gate負責控制是否將其往後傳遞做計算，中間有一個Memory cell負責將值儲存起來，以利下一次的計算。

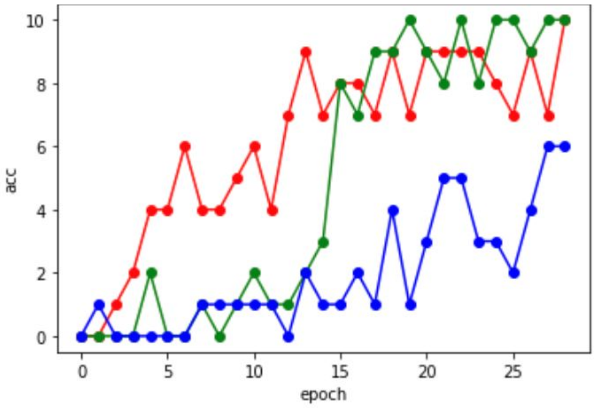


將 Z 與 Zi，經過activation func(sigmoid)，及來自forget gate的input，決定是否存進memory cell。

因為有forget gate去控制是否要記憶進去，因此與RNN相比可以記住比較長時間點的資訊，也是long short-term memory的由來，長的 短期記憶單元 。

參考
https://www.youtube.com/watch?v=xCGidAeyS4M

2.Example4 用LSTM架構, 進行加法訓練

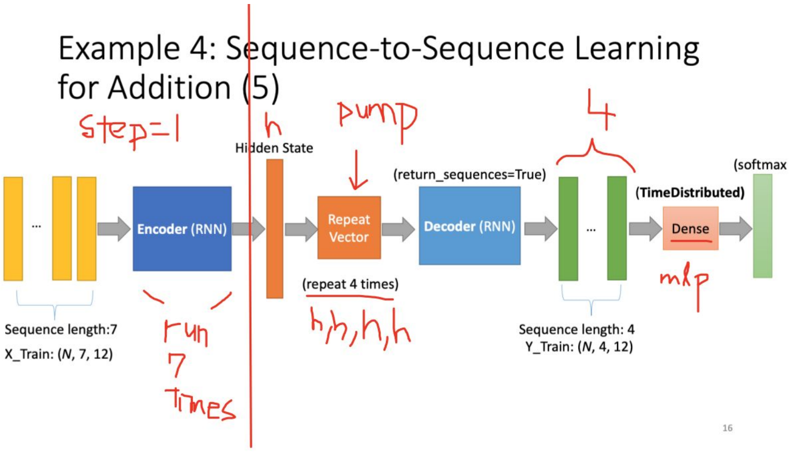


紅色：SimplerRNN
綠色：LSTM
藍色：GRU

縱軸acc為隨機抽10個數字對的次數，橫軸為第幾個epoch時

觀察實驗結果可發現，LSTM在15個epoch以後準確率最高、且穩定維持在10(全對)，使用SimpleRNN的準確率雖然在第三個表現最好，但15個epoch以後準確率卻卡在9，而沒有記憶單元的GRU在此實驗中效果最差。

心得


Example 4: Sequence-to-Sequence Learning for Addition (5)

在課堂中老師有說明到基本上此次學習數字加法是由以RNN為基礎的加法學習，圖為整個學習的步驟。

可看出我們使用encoder與decoder的RNN來處理數字的編碼及解碼，來使RNN可以順利地學習。

Example 4: Sequence-to-Sequence Learning for Addition (1)

- What is sequence-to-sequence learning?

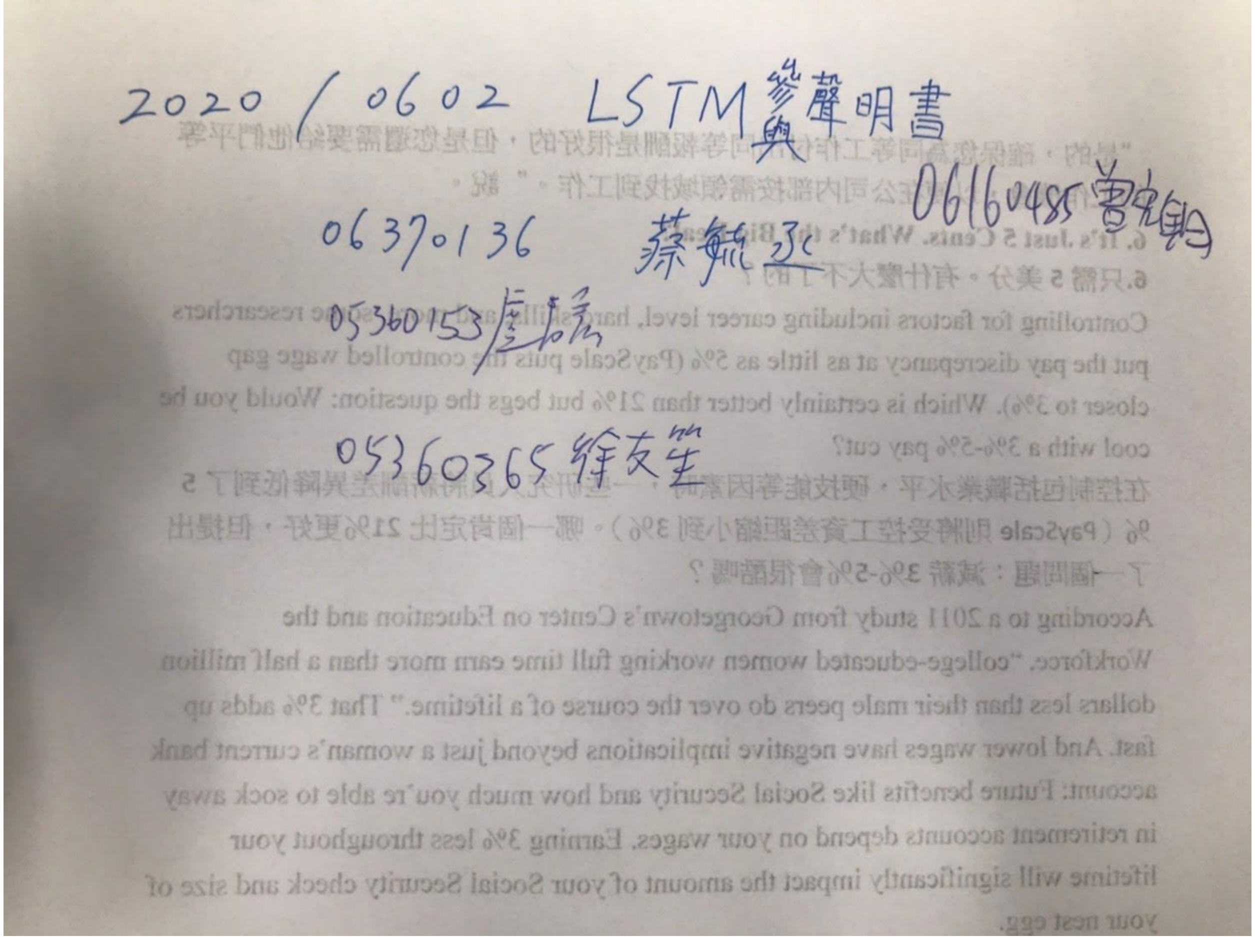
```
Input: ‘3+361_‘
Output: ‘364_‘
```

- In this example, the input string is of length 7, and **reversed**
  - two 3-length numbers and one + sign
- The output is of length 4
- Need a **coding table** for ‘0.123456789+ ‘ (of length 12)

```
Input (reversed): ['  163+3', '  88+71', '    7+2', '356+153', ' 951+01']
Expected (not reversed): ['364 ', '105 ', '9   ', '1004', '169 ']
```

```
ctable.encode('_163+3', 7)
➔
array([
      [1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], #code for blank
      [1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], #code for blank
      [0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.], #code for 1
      [0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0.], #code for 6
      [0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0.], #code for 3
      [0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], #code for +
      [0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0.] #code for 3
])
```

因為需要訓練加法時，需要將資料去做詞嵌入，我們使用one-hot encoding將數字轉成詞向量，再經由RNN去學習input、output及加法後成果的關係。



2020 / 0602 LSTM 參與聲明書

06160485 曾宛昀

06370136 蔡毓弘

05360153 廖晧

05360365 徐友笙