國立中正大學資訊工程學系

期末專題報告

以LSTM預測股價

學生：謝柏威 405410088

黃甚華 405410076

指導教授：林維暘 老師

中　華　民　國　108　年　6　月

目錄

摘要

本研究透過類神經網路中遞迴神經網路模型，以台積電2330作為研究標的，研究所採用的資料為2013到2017期間的股票公開資料，共計5年1224筆。我們預測方式為移動視窗法，以前n天的股價資料來預測第(n+1)天的股價。此外，除了台積電的股價資訊，我們還加入了那斯達克綜合指數(NASDAQ)和道瓊工業平均指數(DJI)，我們採用的是這兩支股票的收盤價格。

我們驗證結果的部分有分為兩種方法，分別是針對股票價格回歸分析，還有針對漲跌的分類方式。

目前研究顯示，回歸分析的部分，我們的誤差都在2%以下(計算方法為MSE)，而以漲跌做分類的部分，我們仍需多加研究。

一、簡介

1.1 研究背景與動機

股票在整個金融市場中，佔了一個非常重要的角色，在股票市場中，光是6/4號一天台灣股市的總成交金額就高達台幣九百七十億，相較於融資融券、公司債券、選擇權、期貨，是較為普羅大眾所熟知的。

以往我們看到股市的時候，可能會認為股票市場是一個隨機漫步的模型，但經過去的研究[3]，我們可以發現其實股票市場不是無跡可循，在市場中許多的變因都是彼此相關而且彼此引響的。

近年來因為人工智慧的發展，隨著演算法的優化，以及電腦運算速度的提升，越來越多的股票操作使用人工智慧來建立模型的方式，來預測在金融市場中股價波動的趨勢，研發出自動交易的模式。以往透過人工的方式，較容易受到情緒的影響而使判斷失準，而能夠處理的資訊量也比較少一些；而透過電腦，可以同時看完非常大量的資料，行為模式也會較為一致。

我們所選用的方式為方式為長短期記憶(Long Short Term Memory Network, LSTM)類神經網路，除了用有大規模並行處理的能力，也擁有極強的自學、自適應和容錯的能力，並擁有良好的多輸入、多輸出的非線性系統，使得它可以被使用的金融領域之中。

1.2 工作分配：

謝柏威：程式撰寫(80%)，資料分析(20%)

黃甚華：程式撰寫(20%)，資料分析(80%)

二、文獻探討

Aaron Elliot, Cheng Hua Hsu(2017)[4]：選用標準普爾500指數作為時序input資料，此篇研究總共實驗了四種模型來做比較，分別是baseline model，Linear model，Generalized Linear Model，Recurrent Neural Network。其中經過實驗RNN(LSTM)的效果最好。

Sang Il Lee and Seong Joon Yoo(2019)[5]: 輸入不同國家的股票資訊，美國股市採用DJI、S&P、NASDAQ，韓國股市則採用KOSPI，將不同時間的股票資訊整合進同一個DNN模型當中。結果發現多層不同的模型架構整合的DNN成效會比單層的來的好。

Najeb M. (2014)[6]：研究期間為2007.1月至2013.3月，使用12項技術指標為輸入變數，並採用artificial neural network來預測利比亞股票市場股票價格，以及使用倒傳遞類神經網路所得之MAE、RMSE、MAPE和R2，能夠更準確的證明其準確度。研究結果顯示，預測出利比亞每日股票價格準確率達91%，以及在MAE、RMSE、MAPE和R2>=0.99，均呈現顯著的效果。

吳月明(2006)[7]：研究期間為2002.7月至2005.6月，樣本以台灣50成分股為標的，資料樣本以季為單位，輸入變數為各公司之財務比率，輸出變數為季股價報酬率，分別利用倒傳遞類神經網路與灰預測模型來做比較。研究結果顯示，倒傳遞類神經網路預測能力較灰預測能力佳。

吳秉奇(1999)[8]：研究期間為1998.7月至1999.4月，以台灣股價指數期貨為標的，共輸入14項變數，並分為三項實驗標的來做預測分析，分別為五日總漲幅、隔日震幅、以及隔二日震幅。研究結果顯示，預測期貨未來走勢準確率最高達71.43%，最低則28.57%，但一半以上準確率達50%。

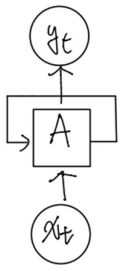
Nghia Nguyen, Minh-Ngoc Tran, David Gunawan, R. Kohn[9]:將兩種時常被金融市場所使用的模型合成一個SV-LSTM模型，透過SV模型可以補捉不同股中動態的關係，比起單純的LSTM模型通常可以有更好的效果。

三、研究模型

3.1 遞迴神經網路(RNN)

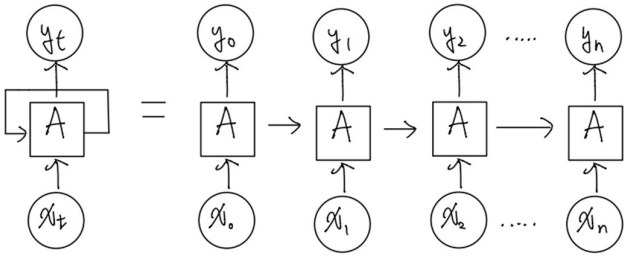
人類不會隨時都重新思考。當閱讀這篇文章時，你會根據上下文的理解來了解這篇文章。你不需要忘記前面的片段重新思考，你的思考是具有記憶性的。傳統的神經網絡無法做到這一點，這是一個明顯的缺點。

遞迴神經網絡解決了這個問題。它們是帶有循環的網絡，藉由訊息的重複傳遞，允許信息持續存在(記憶性)。

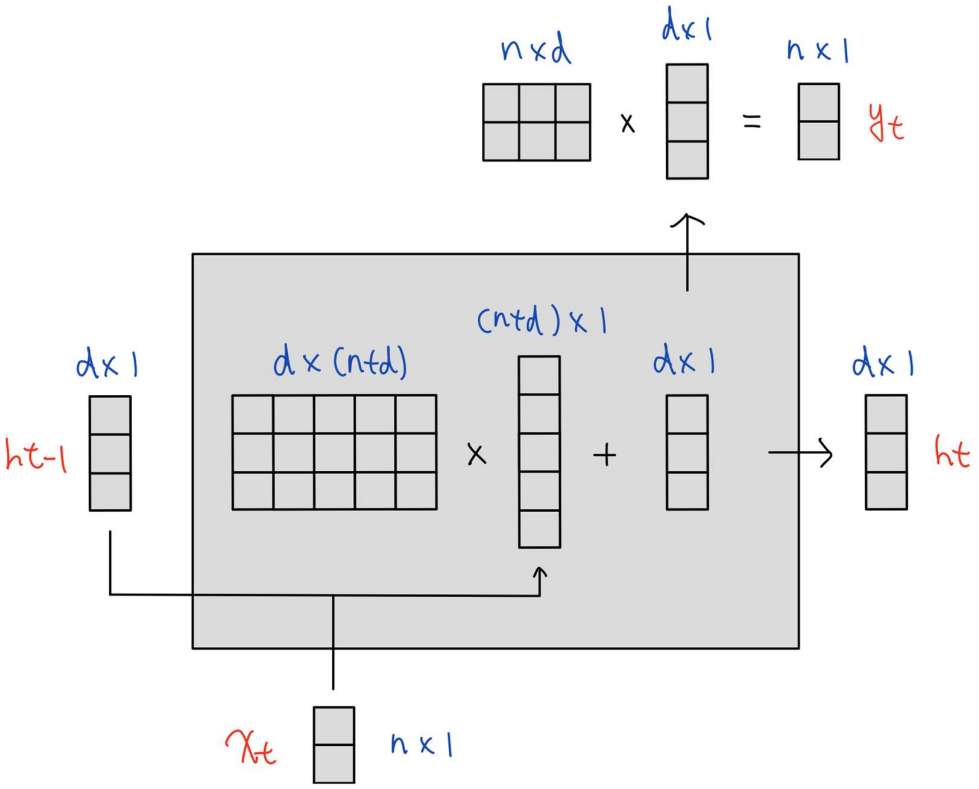


在上圖中是RNN(Recurrent Neural Network)的一個基本單位，x_t為input，y_t則是output。循環的路徑允許訊息從網絡的一個步驟傳遞到下一個步驟。

這種循環讓RNN看起來跟一般的神經網路不太一樣，但是如果再仔細想想，其實它們跟普通的神經網絡並沒有什麼不同。可以將循環神經網絡視為同一單位的多個副本，每個單位都將消息傳遞給後繼者。



在過去幾年中，RNN也常常被用於很多不同的面相：語音識別，語言建模，翻譯，圖像字幕，等等。以下為他的運原理與過程。



這個cell會有2個input，一個來自上一個cell的輸出h_t(假設維度為 dx1)，和一個來自資料的輸入x_t(假設維度維 nx1)，兩個會合成一個(n+d)x1的矩陣。因此這裡的W的維度為dx(n+d)，B的維度為(dx1)，以下公式為這部分的運算。a為activation function。

$h_t = a(W_1 x [h_{t-1}, x_t] + B)$

得到的h_t會有兩個輸出，一個會傳到下一個cell，另一個會經過以下運算輸出。其中W_2的維度為nxd，y_t 的維度為nx1。

$y_t = a(W_2 x h_t)$

若沒有特別說明，我們實驗中採用的參數為n=4, d=50。

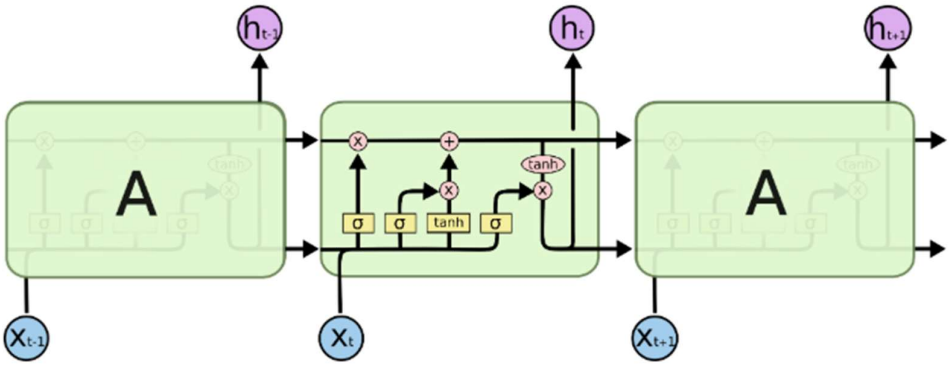
這些研究會如此有成效還有個原因，關鍵在於使用“LSTM”，這是一種特殊的遞歸神經網絡，對於許多問題而言，它比原本的RNN好得多。以下介紹LSTM。

3.2 長短期記憶模型(LSTM)

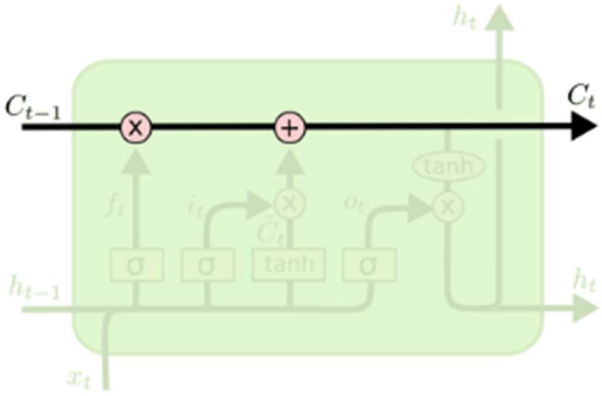
LSTM(long short term memory)[1]是一種特殊的RNN，可以解決長期依賴問題。它是由Hochreiter和Schnidhuber最先提出的，後來很多人用它解決了很多問題，現在被廣泛的應用。

LSTM是被設計出來解決長期依賴問題的。記住時間有用信息是它的基本功能。

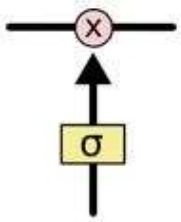
LSTM和RNN一樣，都是這種鍊式的結構，只是重複單元的內部結構不一樣，它不是單獨的NN層，而是4個NN，這4個相互影響。



LSTM的重點是cell state，下面水平這條線從架構的最上面走，cell state就是傳送帶，整個系統就像一條長直鏈，只有一些線性關係，信息往下傳而不會改變。



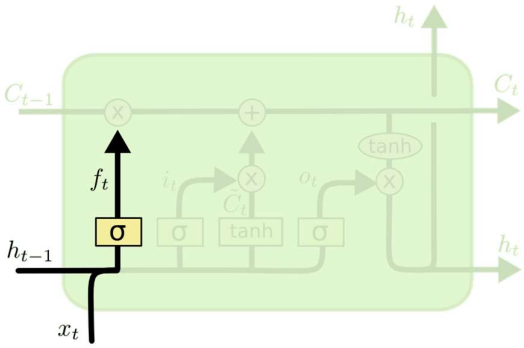
LSTM能刪去或者增加信息，依靠的則是閥門結構。閥門是信息選擇性通過的一種手段，閥門由sigmoid和一個點乘單元組成。



Sigmoid層的輸出介於0和1之間，輸出為0表示信息完全不通過，輸出為1表示信息全部通過。一個LSTM有3個這樣的門來控制和保護cell state。

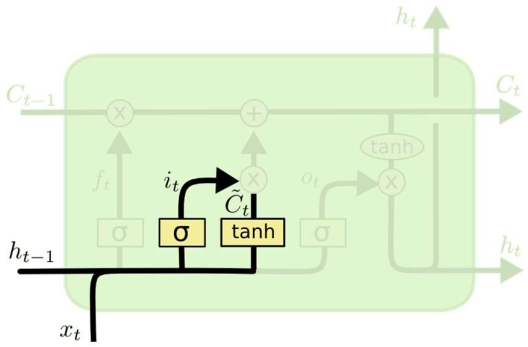
LSTM的第一步是決定哪些信息丟掉，這個動作是由包含sigmoid層的“forget gate layer”來做的。和輸入門中，輸出一個介於0和1的值，作用於（前一個cell state），1表示完全保留，0表示完全丟掉。

回到基於前面的詞預測最後一個詞的語言模型中，如果這個cell state包含有前一個話題的特徵，那麼正確的代詞會被用到，如果從一個新的話題開始，那麼舊話題的特徵將被遺忘掉。



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

下一步決定哪些新的信息將被加入cell state。這由兩部分構成，一是由sigmoid層構成的輸入門，它用來決定哪些值要更新。另一個是tanh層構成的新候選值的向量生成器，可能會被輸入到cell state裡面。

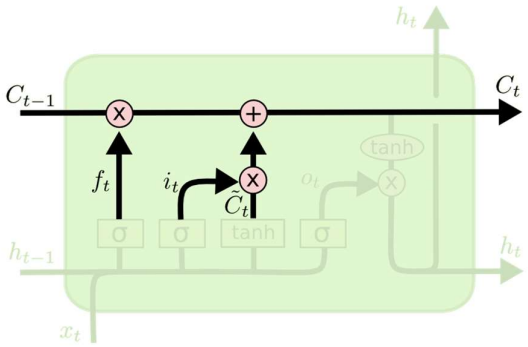


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

接下來，前一個cell state（輸入到當前cell state）中，前面已經決定了要做什麼，這裡只需要一次完成就好。

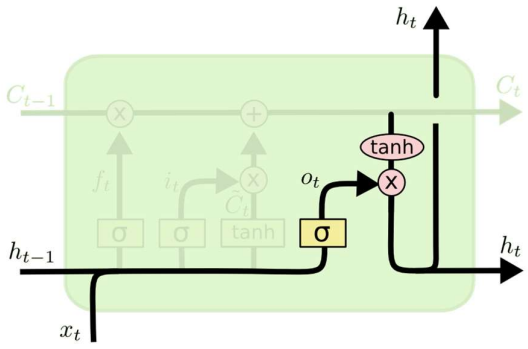
我們用舊的cell state乘上forget gate的輸出，組成當前cell state。

(c_t的維度和h_t相同)



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

最後決定要輸出什麼。這個輸出基於cell state，但會有一個過濾過程。首先，一個sigmoid層決定哪些cell state輸出，同時將cell state輸入到tanh層，乘sigmoid門的輸出，得到最後的輸出。



$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

四、研究方法

本文研究方法為應用類神經網路的訓練來預測隔日的股票未來走勢。本章會分成四節來說明本研究所採用之研究方法。第一節為資料來源、資料描述、研究對象與研究時間，第二節為說明我們所使用的研究方法架構，第三節為兩種不同方向的研究方法。

4.1 資料的來源以及樣本說明

我們的資料來源來自於Yahoo!奇摩股市， 2013年至2017年之日資料，一共1224筆，收集對象包含台積電(2330.TW)之收盤價、交易量，另外我們還收集了那斯達克綜合指數(NASDAQ)和道瓊工業平均指數(DJI)之收盤價和交易量，進而預測未來股票未來的走勢。

4.2 研究方法與架構

1. 移動視窗法[2]

本文的研究對象為台積電(2330.TW)，樣本期間為2013年1月至2017年12月，共4年。訓練時，第一筆的訓練資料，以LSTM訓練前60天來預測第61天的股票收盤價，下一筆測試資料則是以LSTM訓練第2天到第61天來預測第62天，一次移動1天，共1164次的訓練次數。測試時，則是取2018年的其中連續20筆來做預測，同樣以前60天來預測61天。

2. 研究模型

目前我們所選擇的是LSTM的模型架構，給予的訓練資料須包含輸入變數與輸出變數，主要由輸入層(input layer) 、 輸出層(output layer)、隱藏層(hidden layer)，輸入層的神經元用於將外部的資料輸入，輸出層的神經元用於將結果輸出，隱藏層負責其中資料的交互作用，其中的遺忘閥(forget gate)又最為重要，它可以讓較久遠的資料同樣對結果造成影響。

輸入的變數有台積電的收盤價、交易量，因為台積電為偏向工業的產業，所以我們加入了美國工業的指數性股票，那斯達克綜合指數(NASDAQ)和道瓊工業平均指數(DJI)的收盤價。

4.3 研究方法

第一種所用的方法是用同樣收盤價來預測未來的收盤價。

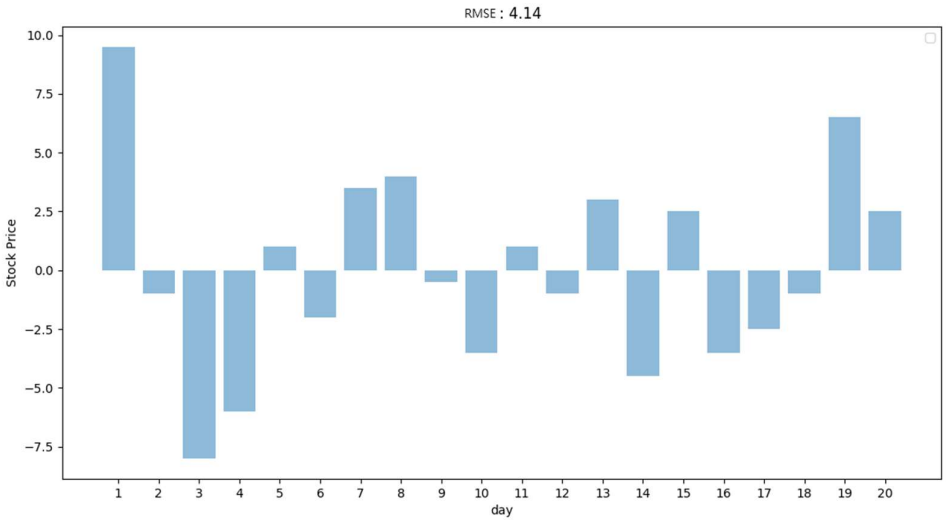
第二種則是除了以上幾種的輸入之外，我們另外加入的一個資料是漲跌的資訊，而輸出的部份我們也是取漲跌的部分。

我們用來判斷模型的好壞的依據是Root Mean Square Error(RMSE)，這個error function能夠讓我們看出模型的擬和程度。將每個預測出的x值與其相對應的ground truth取差值，並取方均根，及為我們要的error值。

$$\bar{s} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}}$$

4.4 研究分析與結果

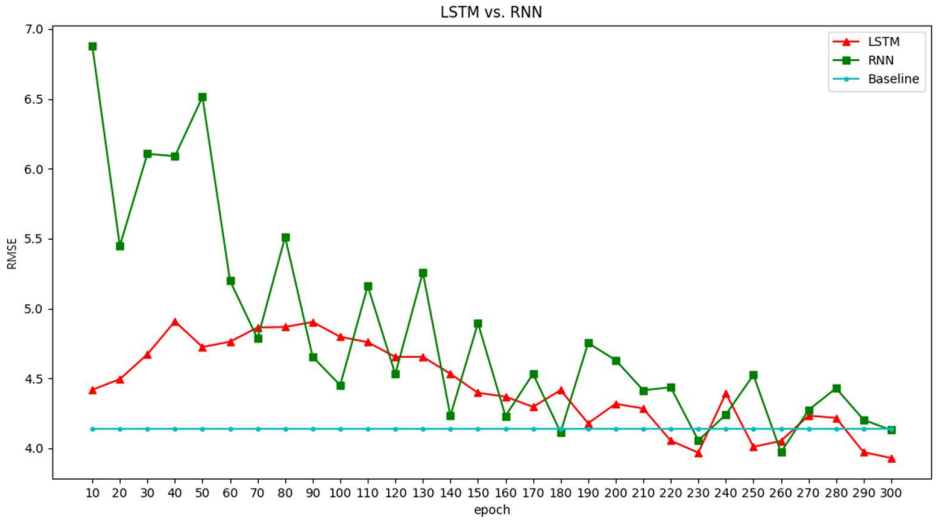
本研究採用類神經網路建立LSTM模型，採用2013到2017的股票價格，並使用移動式窗法進行收盤價格的預測。本章共分為X個部分做討論，觀察不同的測驗環境對於結果預測的影響。我們除了平常大家熟知的RMM和LSTM之外，我們還加入了BASELINE當作參考，這個的模型是預測第n天時，就直接取第n-1天實際的值當作預測的結果，因此不管訓練的epoch數，都會保持不變。在下方的結果除了RNN和LSTM的比較外，無特別標示的皆為LSTM。下圖為baseline的error呈現。

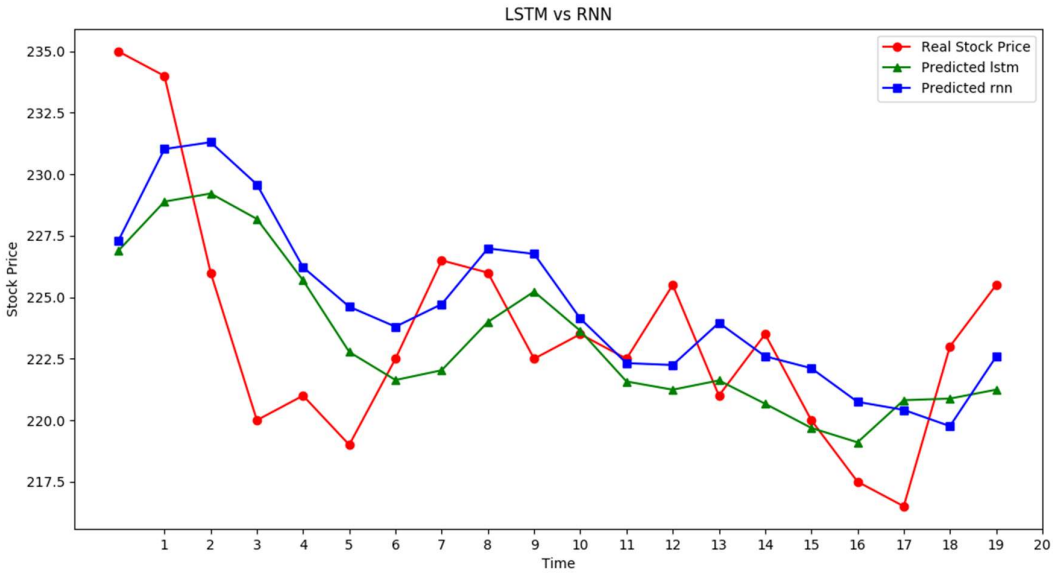


1. RNN 與 LSTM比較

首先我們先固定所有參數，並分別測試了LSTM與RNN的模型，觀察兩種不同的模型對於我們結果的影響。

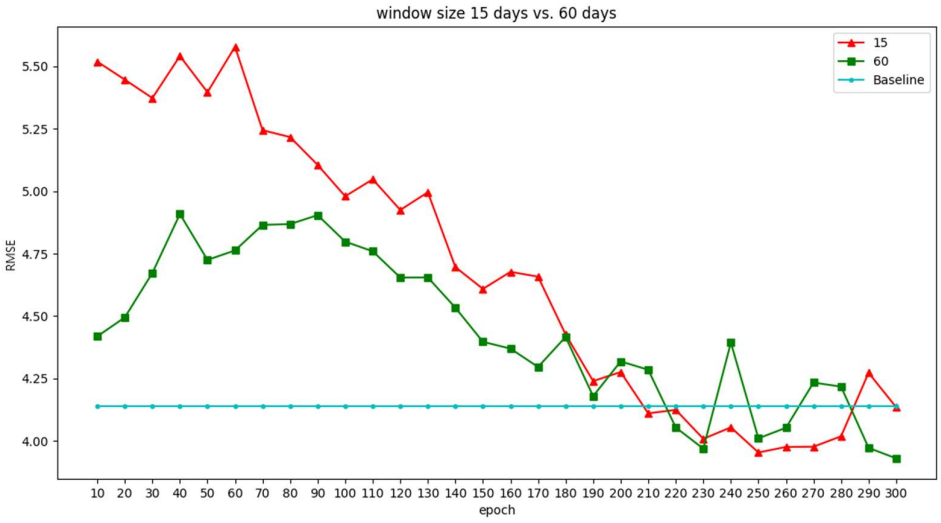
可以由下圖看出，在趨勢線的走勢上，LSTM會與ground truth較為接近，但是在Mean Square Error那張圖上，則較難看出差異。因此目前這部分在我們的實驗上影響不大。

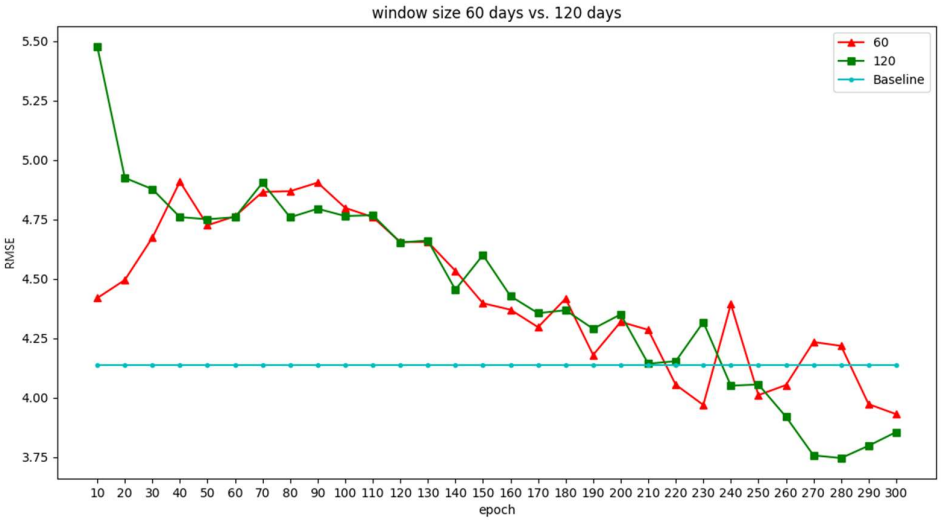




2. 移動視窗的天數的調整對結果的影響

在我們原先的設定下，移動視窗的天數是60天，但是我們好奇究竟模型能夠接受多麼長的一段的資訊，因此我們嘗試拉長與縮短預測時間，以下是我們的結果。如圖所見，超過60天並不太有太多影響，但是天數太短確實會影響準確率，因此我們經過調整後的天數改為15天。

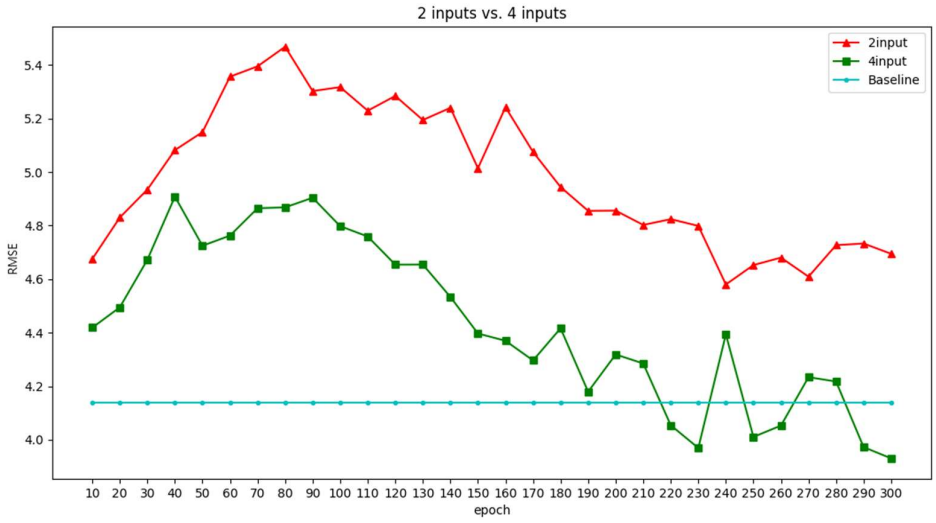




3. 改變資料輸入的維度(輸入多樣性)

如同前面所提到的，我們的輸入資料分別是2330股票的收盤和交易量、nasdaq的收盤價格、和道瓊的收盤價格，但我們原先只有採用股票的資料，因此以下是我們實驗過程中加入nasdaq與道瓊指數對結果的影響。

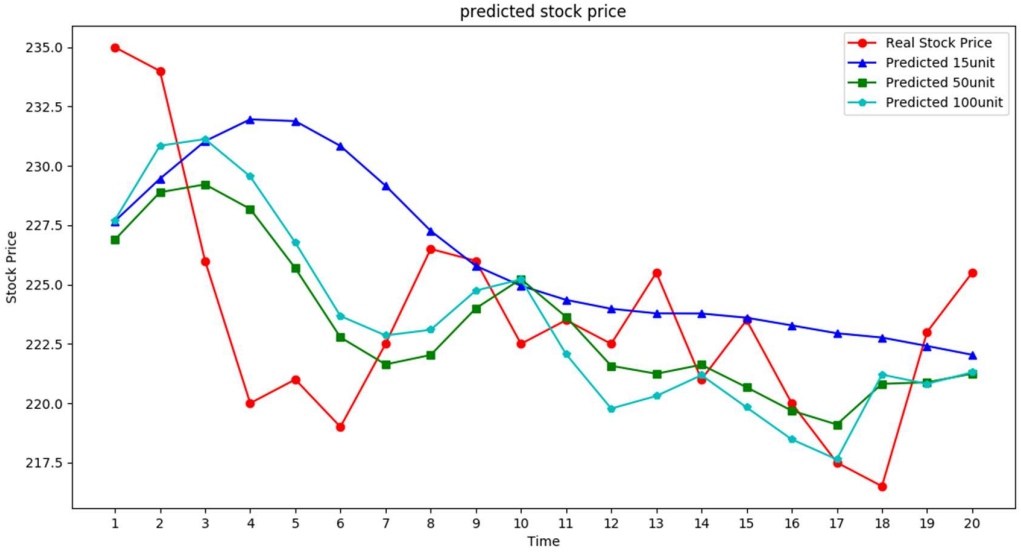
如下圖可以發現，有4input的模型最終有較小的error值，因此我們可以判斷這兩筆資訊對於模型是有幫助的。

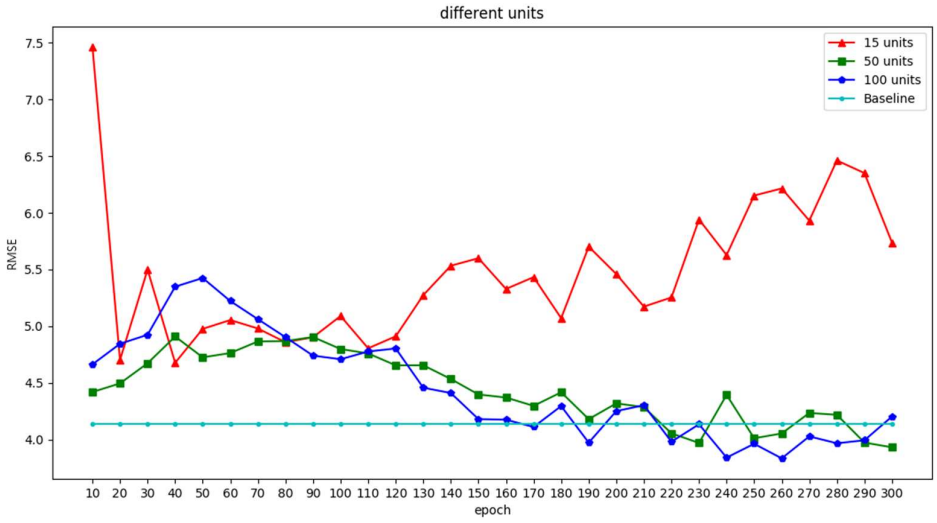


4. 改變hidden state(unit)的維度

我們改變了在各個cell中傳遞的向量(hidden state)的維度，我們想要嘗試是否只要讓傳遞的向量越大，結果就會越好。

如下圖所示，unit較小時，在預測時的表現較差，但當unit增加到了一定的大小之後，兩者的表現是差不多的。





5. regression vs. classification

以上的實驗結果皆為regression的實驗結果，以下為我們的classification結果。

在classification的部分，我們輸入的資料和上面介紹的regression的輸入是相同的(股票的收盤價、交易量、dji的收盤價、Nasdaq的收盤價)。但在輸出的部分由原本的4output變成2output。2output分別為預測上漲，與下跌的機率，而我們的實作方法是在模型的最後一層加入softmax，可以將數據轉換為機率形式。

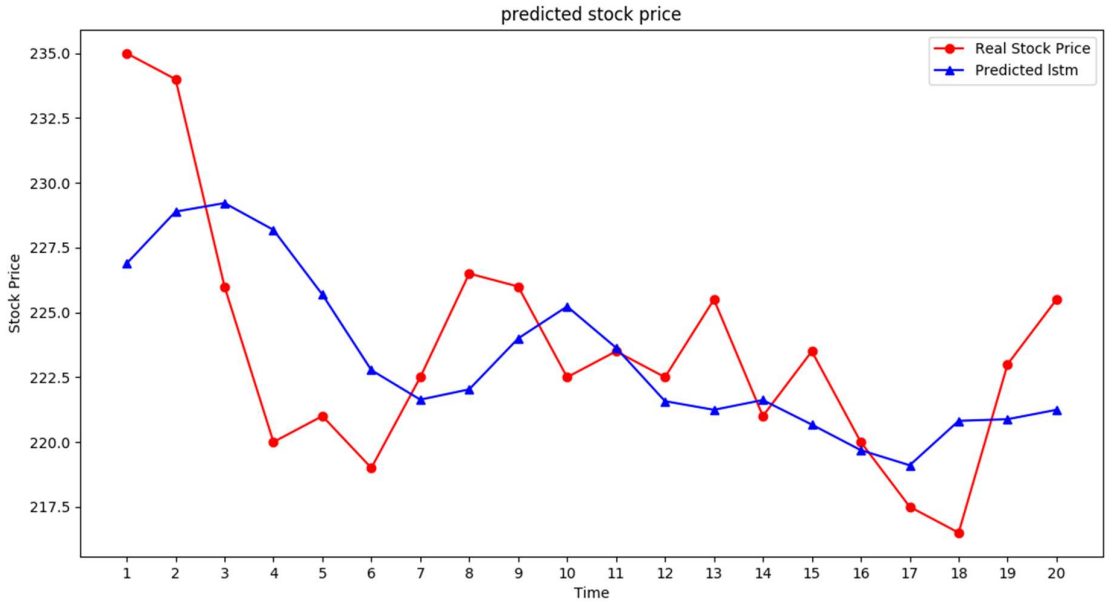
但是如果觀察下圖，可以發現模型預測的每一天都是一樣的(在這裡預測的都是下跌)，因此準確率是45.00%，且不論是RNN或是LSTM皆為類似結果。

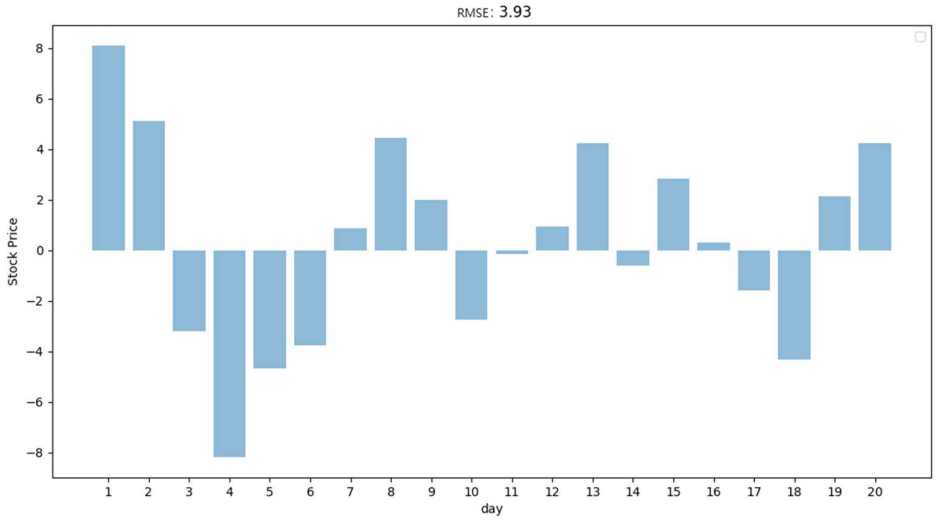
```
[[0.38650236 0.6134976 ]
 [0.38592234 0.6140777 ]
 [0.38519505 0.614805  ]
 [0.38438845 0.6156115 ]
 [0.38356698 0.616433  ]
 [0.38298357 0.61701643]
 [0.38278908 0.617211  ]
 [0.38291577 0.6170842 ]
 [0.3833021  0.61669797]
 [0.38381067 0.6161893 ]
 [0.384285   0.61571497]
 [0.3849314  0.6150686 ]
 [0.38569197 0.6143081 ]
 [0.38663918 0.61336076]
 [0.38763937 0.6123606 ]
 [0.38858977 0.6114102 ]
 [0.38971362 0.61028636]
 [0.39104813 0.60895187]
 [0.39255708 0.6074429 ]
 [0.3941355  0.6058646 ]]
[1 0 0 0 1 0 1 1 0 0 1 0 1 0 1 0 0 0 1 1]
accuracy: 45.00%
```

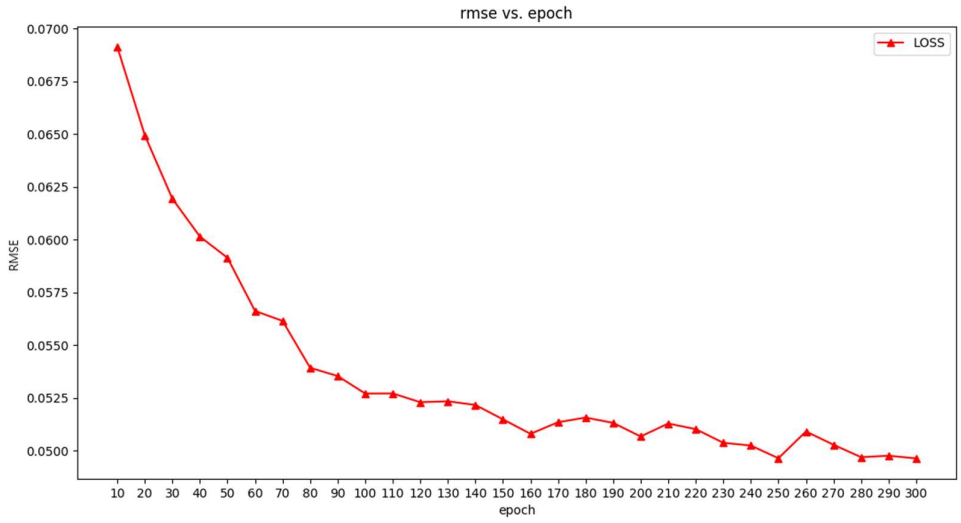
五、結論

5.1

下圖是我們目前第一種方法所得出最好的模型(300epooch , 4 input , 60window size, 50unit)







5.2

我們以2013到2017的股票作為資料，以下是我們的結論。

1. 若是增加跟模型有相關性的資料(close, volumn, nasdaq, dji)作為輸入資料，對於預測的準確率具有幫助。

2. RNN與LSTM間的效果，以目前的實驗結果來說，並沒有差非常多。

3. 移動視窗的天數太長對於模型來說沒有意義，但是也不能太短到讓模型無從學習起。

4. 目前訓練分類漲跌的分類方式仍需改進。

六、參考文獻

1. S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 1997.
2. A Study on the Profitability of Neural Network Stock Prediction:The Case of TAiwan 50I ndex Constituents.2017
3. Fama, E. F. The behavior of stock-market prices. The Journal of Business, 1965
4. Aaron Elliot, Cheng Hua Hsu, Time Series Prediction: Predicti ng Stock Price, 2017
5. Sang Il Lee and Seong Joon Yoo. Multimodal Deep Learning for Finance: Integrating and Forecasting International Stock Markets.Computer Engineering.2019
6. 郭英哲，運用倒傳遞類神經網路技術於台灣指數期貨預測之研究，南台科技大學 資訊管理研究所 碩士論文，2004
7. 吳月明，股票報酬率預測模式績效之研究，朝陽科技大學 財務金融系 碩士論 文， 2006
8. 吳秉奇，類神經網路在台股指數期貨的預測應用，國立中央大學 資訊管理學系 碩士論文，1999
9. Nghia Nguyen, Minh-Ngoc Tran, David Gunawan,R. Kohn†,A long short-term memory st ochastic volatility model. Machine Learning 2019