

國立成功大學
工業與資訊管理學系
類神經網路期末報告

使用人工神經網絡預測股市指數

Stock market index prediction using artificial neural network

指導老師：王泰裕教授

- 學生：
1. 工資管所 R36104138 王怡琿
 2. 工資管所 R36101237 廖竟帆
 3. 資管所 R76101023 崔緯平
 4. 資管所 R76101120 黃翌

中華民國一一一年一月

摘要

納斯達克指數（NASDAQ）是美國代表性的股市之一，全名為”National Association of Securities Dealers Automated Quotations “，組成大部分是高科技產業或 IT 產業的公司，以新興企業為多，是由在美國那斯達克交易所上市的公司市值加權指數所構成。跟道瓊指數及 S&P500 並列全美最重要 3 大指數，為投資人查看股市的重要資訊指標。

在這項研究中，研究了人工神經網絡在預測每日納斯達克股票匯率方面的能力。並評估了幾個由反向傳播算法訓練的前饋式人工神經網絡。本研究中使用的方法將短期的歷史股價和一週中的某一天作為輸入。使用納斯達克於 2015 年 1 月 28 日至 2015 年 6 月 18 日的每日股票匯率來開發穩健型模型。選擇前 70 天作為訓練的數據集，後 29 天用於測試模型預測能力。開發並驗證了兩種輸入數據集的 NASDAQ 指數預測網路。

關鍵字: NASDAQ、ANN、預測

目錄

第一章 緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究目的.....	1
第二章 文獻回顧.....	2
2.1 Levenberg–Marquardt 萊文貝格–馬夸特 (LM)	2
2.2 One Step Secant 一步割線法(OSS)	3
2.3 Gradient Descent with Adaptive Learning Rate 具有自適應學習率的梯度下降(GDA).....	3
2.4 Long Short-Term Memory 長短期記憶(LSTM).....	4
2.5 支援向量機 (SVM).....	5
第三章 研究方法.....	6
3.1 系統架構.....	6
3.2 類神經網路架構.....	6
3.3 資料集介紹.....	7
第四章 實驗與數據.....	8
4.1 實驗流程與架構.....	8
4.2 實驗方法.....	9
4.2.1 SVM.....	9
4.2.2 LSTM.....	10
4.3 結果比較.....	10
第五章 結論.....	12
參考文獻.....	13

表目錄

表 3.1 資料集概覽.....	7
表 4.1 具有不同訓練和傳遞函數的 BPNN 的預測能力.....	8
表 4.2 在不同結構的 BPNN 下以四天作一次預測的 R^2 值.....	8
表 4.3 在不同結構的 BPNN 下以九天作一次預測的 R^2 值.....	9

圖目錄

圖 2.1 Momentum 和 Nesterov 加速梯度.....	4
圖 2.2 LSTM 架構圖.....	5
圖 2.3 超平面分類器之示意圖.....	5
圖 3.1 系統架構圖.....	6
圖 3.2 MLP 網路架構圖.....	6
圖 4.1 SVM 結果.....	10
圖 4.2 LSTM 所有資料結果.....	11
圖 4.3 LSTM 測試資料結果.....	11

第一章 緒論

1.1 研究背景與動機

過去十年來 4 個最有代表性的股災，分別是 2011 年 8 月股災、2015 年 8 月股災，以及 2018 年 10 月股災，還有 2020 年 3 月的股災，許多投資客因此失利而在經濟上跌落谷底。而最嚴重的股市波動要屬 COVID-19 病毒肆虐這個災難了，為了阻止病毒傳播而實施的大量措施，卻導致經濟停擺，規模之大是近一個世紀前所未有。自 3 月中來，已經有數千萬美國人申領失業救濟金。

面對如此嚴峻情況，許多人對股市的成長走勢感到困惑。從 3 月 23 日至 4 月 29 日，S&P500 指數大漲超過 30%。部分市場參與者將股市與經濟的明顯脫節，歸因於市場不合理的樂觀情緒，投資者低估了冠狀病毒危機的嚴重性，以及對經濟和股市的長遠影響。然而，若只看市場反彈，卻沒有看之前的暴跌，看法未免以偏概全。只有看看股市的整體影響，並精確的預測出股市走向才能巧妙地從股市中生存。

在研究某些現象時，由於這些現象的複雜性，開發一個數學模型來模擬輸入和輸出參數之間的非線性關係是一項艱難的任務。人工神經網絡 (ANN)、模糊推理系統 (FIS) 和自適應神經模糊推理系統 (ANFIS) 等人工智能系統已被應用於對科學和工程中的各種具有挑戰性的問題進行建模。而 ANN 在破產預測方面表現出比判別分析和邏輯回歸等傳統統計方法更好的性能。信用評等過程的調查表明，由於金融和其他輸入的變量之間存在著複雜的關係，因此 ANN 比統計方法具有更好的預測能力。破產預測、信用風險評估和證券市場應用是 ANN 被廣泛應用的其他經濟領域。

1.2 研究目的

歷史上股災不斷在發生，人類也不斷在重複著歷史，但以人的本性很容易產生投機的羊群效應，容易跟隨眾人追高行情，導致股價與實際企業價值偏離太遠，最後造成股市崩盤。我們無法預測股災何時來臨，但我們能做的就是避開風險，並學會如何預測股市走向和波動。本研究的目的是為了調查 ANN 在預測每日那斯達克股票匯率方面的能力。利用不同的 training function 和 transfer function 彼此的配合，從中找出最佳的預測能力解。

第二章 文獻回顧

Guresen、Kayakutlu 和 Daim(2011 年)研究了多層感知器 (MLP)、動態 ANN 和混合 ANN 模型在預測市場價值方面的能力。Chen、Leung 和 Daouk (2003)使用概率神經網絡(PNN)來預測台灣股票指數回報的方向。在他們的研究中，PNN 在股票指數方面比廣義矩卡爾曼濾波器和隨機遊走預測模型具有更高的性能。Kuo、Chen 和 Hwang (2001)通過結合基於基因演算法的模糊神經網絡(GFNN)開發了決策支援系統和股票市場 ANN。使用台灣股票市場的數據對所提出的系統進行了評估。Qiu、Liu 和 Wang(2012)基於模糊時間序列和 C-模糊決策樹開發了一種新的預測模型來預測上海綜合指數的股票指數。Atsalakis 和 Valavanis(2009)開發了一種自適應神經模糊推理控制器來預測第二天的股價趨勢。他們研究了 ANFIS 在預測股票指數方面的潛在能力。

2.1 Levenberg–Marquardt 萊文貝格—馬夸特 (LM)

為提供數非線性最小化（局部最小）的數值解。此演算法能藉由執行時修改參數達到結合高斯-牛頓算法以及梯度下降法的優點，並對兩者之不足作改善（比如高斯-牛頓算法之反矩陣不存在或是初始值離局部極小值太遠）。

問題描述：

假設 f 是一個從 $R^m \rightarrow R^n$ 的非線性映射，也就是說 $P \in R^m$ ，那麼：
$$f(P) = X$$

目的就是希望任意給定一個 X 以及合理的初始值 p_0 ，能找到一個 p^+ ，使得 $\epsilon^T \epsilon$ 盡量小（局部極小），其中 $\epsilon = f(p^+) - X$ 。

解法：

像大多數最小化的方法一樣，這是一個迭代的方法。首先根據泰勒展開式把 $f(p + \delta_p)$ 寫為下面的近似，有兩個好處：第一是線性、第二是只需要一階微分。

$$f(p + \delta_p) \approx f(p) + J\delta_p$$

其中 J 是 f 的雅可比矩陣。對於每次的迭代：假設這次 iteration 的點是 p_k ，要找到一個 $\delta_{p,k}$ 讓

$|X - f(p + \delta_{p,k})| \approx |X - f(p) - J\delta_{p,k}| = |\epsilon_k - J\delta_{p,k}|$ 最小。根據投影公式知道當下面式子被滿足的時候能有最小誤差：

$$(J^T J)\delta_p = J^T \epsilon_k$$

將這個公式略加修改得到：

$$[\mu I + (J^T J)]\delta_p = J^T \epsilon_k$$

就是萊文貝格—馬夸特方法。如此一來 μ 大的時候這種算法會接近最速下降法，小的時候會接近高斯-牛頓方法。為了確保每次 ϵ 長度的減少：先採用一個小的 μ ，如果 ϵ 長度變大就增加 μ 。

這個演算法當以下某些條件達到時結束迭代：

1. 如果發現 ϵ 長度變化小於特定的給定值就結束。

2. 發現 δ_p 變化小於特定的給定值就結束。
3. 到達了迭代的上限設定就結束。

2.2 One Step Secant 一步割線法(OSS)

此方法試圖彌合共軛梯度算法和準牛頓（割線）算法之間的差距。該算法不儲存完整的 Hessian 矩陣；它假設在每次迭代中，前一個 Hessian 是單位矩陣，此具有額外的優勢，即可以在不計算逆矩陣的情況下計算新的搜索方向。

只要權重、淨輸入和傳遞函數具有導數函數，trainoss 就能訓練任何網絡。

反向傳播用於計算性能 perf 相對於權重和偏差變量 X 的導數。每個變量根據以下內容進行調整：

$$X = X + a * dX$$

其中 dX 是搜索方向。參數 a 被選擇來沿著搜索方向做最小化表現。線搜索函數 searchFcn 用於定位最小值點。第一個搜索方向表現梯度是負值。在隨後的迭代中，根據以下公式，根據新梯度和先前的步驟和梯度計算搜索方向：

$$dX = -gX + Ac * X_{\text{step}} + Bc * dgX$$

其中 gX 是梯度， X_{step} 是前一次迭代權重的變化， dgX 是上次迭代的梯度變化。

當出現以下任何一種情況時，訓練就會停止：

1. 達到最大時期數（重複）。
2. 已超過最大時間。
3. 表現被最小化到目標。
4. 表現梯度低於 min_{grad} 。
5. 自上次下降（使用驗證時）以來，驗證表現增加了超過 max_{fail} 次。

2.3 Gradient Descent with Adaptive Learning Rate 具有自適應學習率的梯度下降(GDA)

作為對傳統梯度下降算法的改進，可以利用自適應梯度下降優化算法或自適應學習率方法。以下為幾種算法：

Momentum 可以看作是 SGD 的演變。

$$v_t = \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta)$$

$$\theta = \theta - v_t$$

雖然 SGD 在梯度的一個方向上具有陡峭曲線的數據存在問題，但 Momentum 通過在乘以 γ （通常約為 0.9(1)）之前添加時間步的更新向量來規避這一問題。打個比方，你可以想像一個球沿著梯度滾下，收集動量（因此得名），同時仍然受到風阻的影響（ $0 < \gamma < 1$ ）。

Nesterov 加速梯度可以看作是對 Momentum 的進一步增強。

$$v_t = \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta - \gamma v_{t-1})$$

$$\theta = \theta - v_t$$

該算法以 γv_{t-1} 的形式添加了對下一步的猜測。在圖 2.1 中可以找到 Momentum 和 Nesterov 加速梯度的前兩個步驟的比較。附加項導致考慮上一步的誤差，與動量相比加速了進度。

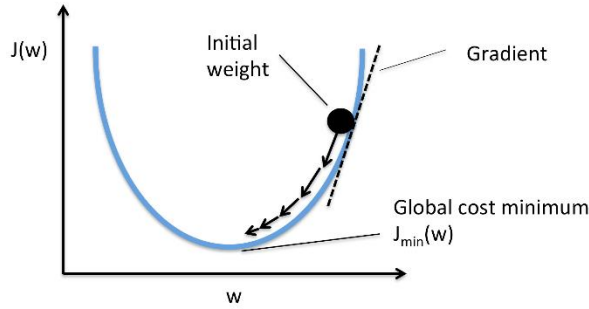


圖 2.1 Momentum 和 Nesterov 加速梯度

與 Nesterov 加速梯度相反，Adagrad 在其運行時調整其學習率 η ，並在每個時間步長 t 期間分別更新其參數 θ_i 。它必須如此，因為 η 自己適應每個 θ_i 。

2.4 Long Short-Term Memory 長短期記憶(LSTM)

LSTM 是一種含有 LSTM 區塊 (blocks) 或其他的一種類神經網路，文獻或其他資料中 LSTM 區塊可能被描述成智慧型網路單元，因為它可以記憶不定時間長度的數值，區塊中有一個 gate 能夠決定 input 是否重要到能被記住及能不能被輸出 output。

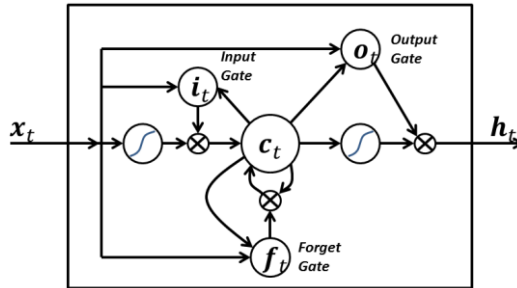


圖 2.2 LSTM 架構圖

方程式：

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

$$c_t = f_t * c_{t-1} + i_t * \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$

$$h_t = o_t * \sigma_h(c_t)$$

2.5 支援向量機 (SVM)

支援向量機 (SVM) 是一個用於處理分類問題的工具。考慮一組訓練資料 (x_i, y_i) ，其中 x_i 為隨機向量、 y_i 為目標變數、 $y_i \in \{-1, 1\}$ 及 $i=1, \dots, l$ 為訓練樣本數，利用超平面分類器 (Hyperplane classifier) 對資料進行分類，如等式(1)：

$$W^T x_i + b = 0 \quad (1)$$

其中 w 與 b 為超平面分類器之參數。

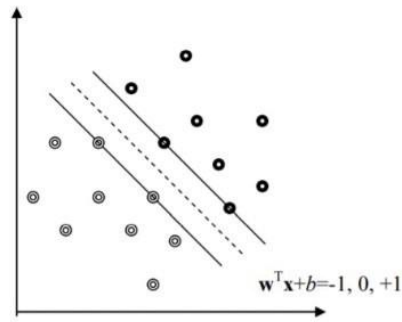


圖 2.3 超平面分類器之示意圖

在線性分類的情況下，可滿足等式(2)：

$$\begin{cases} w^T x_i + b \geq 1 & \text{if } y_i = 1 \\ w^T x_i + b \leq -1 & \text{if } y_i = -1 \end{cases} \equiv y_i(w^T x_i + b) \geq 1 \quad (2)$$

第三章 研究方法

3.1 系統架構

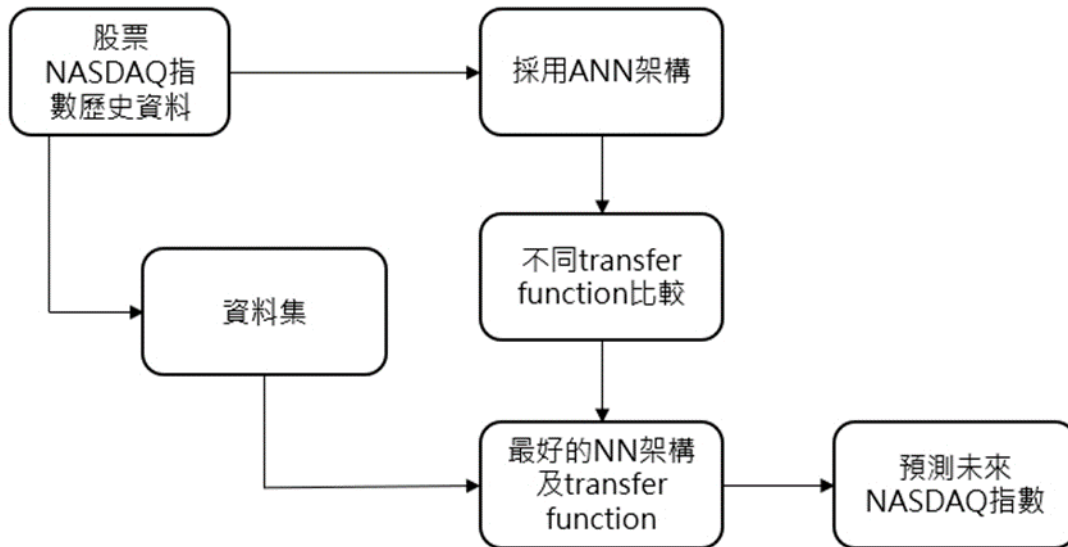


圖 3.1 系統架構圖

本論文研究所提出的系統架構圖如圖 3.1 所示，在研究中他使用了 ANN 網路架構，並且比較各個 transfer function，最後決定最佳並以此來預測為來的 NASDAQ 指數。

3.2 類神經網路架構

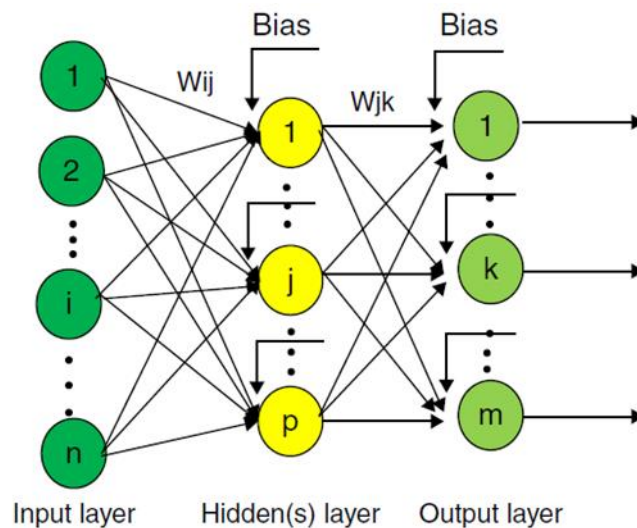


圖 3.2 MLP 網路架構圖

圖 3.2 為 MLP 網路架構圖，並且可以把 neuron P 的輸出寫成下列的數學示：

$$u_p = \sum_{i=1}^n w_{p_i} x_i \quad (3)$$

$$y_p = \varphi(u_p + b_p) \quad (4)$$

在等式(3)中 $x_1 \dots x_n$ 為輸入值， $w_{p_1} \dots w_{p_n}$ 為其權重， u_p 為輸入值的總和，在等式(4)中 y_p 是神經元的輸出， b_p 是 bias、 φ 為 activation function，並且在之後有許多的學習方法(learning techniques)，例如有 scaled conjugate gradient(SCG)、Levenberg-Marquardt (LM)、one step secant(OSS)、gradient descent with adaptive learning rate (GDA)、gradient descent with momentum (GDM)，並且在之後的實驗比較各種方法以此來找出最好的方法。

3.3 資料集介紹

資料集包含：

1. NASDAQ 指數 2015 年 1 月 28 號到 2015 年 6 月 18 號之股票匯率資料
2. 選取前 70 天的資料做為訓練資料集，後 29 天作為測試模型預測能力之資料集。

表 3.1 資料集概覽

Date	Open	High	Low	Close	Adj Close	Volume
2015/1/28	4740.69	4742.06	4637.48	4637.99	4637.99	2118680000
2015/1/29	4635.73	4688.41	4601.76	4683.41	4683.41	2110710000
2015/1/30	4671.21	4703.81	4631.1	4635.24	4635.24	2264230000
2015/2/2	4650.6	4676.69	4580.46	4676.69	4676.69	2006450000
2015/2/3	4693.25	4727.74	4670.82	4727.74	4727.74	2153520000
2015/2/4	4699.81	4744.33	4697.79	4716.7	4716.7	2212960000
2015/2/5	4729.65	4767.38	4722.8	4765.1	4765.1	2036590000
2015/2/6	4768.84	4787.18	4731.22	4744.4	4744.4	2034060000
2015/2/9	4723.73	4749.47	4719.61	4726.01	4726.01	1654680000
2015/2/10	4754.63	4793.27	4737.12	4787.64	4787.64	1773730000
2015/2/11	4788.37	4810.36	4780.13	4801.18	4801.18	1793630000
2015/2/12	4828.08	4857.61	4823.57	4857.61	4857.61	2075470000
2015/2/13	4869.73	4893.84	4860.12	4893.84	4893.84	1942240000
2015/2/17	4889.99	4901.89	4880.64	4899.27	4899.27	1751490000
2015/2/18	4890.84	4907.5	4885.6	4906.36	4906.36	1684800000
2015/2/19	4901.51	4929.53	4900.63	4924.7	4924.7	1592890000

第四章 實驗與數據

4.1 實驗流程與架構

本篇論文針對兩個輸入數據集（前四天和前九天）開發和驗證了幾個用於納斯達克指數預測的網絡，並根據兩種類型的數據集的能力預測選擇優化的網絡結構。

表 4.1 具有不同訓練和傳遞函數的 BPNN 的預測能力

No.	Training function	Transfer function	R ²			
			Train	Test	Validation	Total
1	LM	TANSIG	0.9925	0.9869	0.8864	0.974
2	LM	PURELIN	0.9457	0.9675	0.9027	0.9395
3	LM	LOGSIG	0.9989	0.9698	0.7339	0.9475
4	OSS	LOGSIG	0.9166	0.9133	0.8669	0.9069
5	OSS	PURELIN	0.7016	0.8824	0.8230	0.7675
6	OSS	TANSIG	0.9386	0.8917	0.9408	0.9267
7	GDA	LOGSIG	0.9016	0.8308	0.8497	0.8649

如上表 4.1 所示，根據驗證數據集的 R^2 值，應用 OSS 訓練方法和 TANGSIG 傳遞函數產生了一個最優化的訓練網絡。因此，在本研究的接下來的實驗中，分別選擇 OSS 和 TANSIG 作為訓練方法和傳遞函數。

表 4.2 在不同結構的 BPNN 下以四天作一次預測的 R^2 值

No.	Structure	R ²			
		Train	Test	Validation	Total
1	2	0.8177	0.9616	0.9493	0.8692
2	5	0.9250	0.9188	0.9605	0.9264
3	5-5	0.9229	0.9724	0.8631	0.9212
4	5-10	0.2185	-0.0165	-0.2065	0.0838
5	10-10	0.9534	0.9602	0.6811	0.9344
6	10-20	0.9059	0.9758	0.9108	0.9263
7	40-40	0.9003	0.9639	0.9616	0.9264
8	50-100	0.9576	0.9324	0.9393	0.9483
9	100-200	0.9390	0.9466	0.9533	0.9393
10	200-300	0.9267	0.9642	0.8822	0.9276
11	20-40-20	0.9386	0.8917	0.9408	0.9267
12	20-50-20	0.9403	0.9417	0.9077	0.9374
13	50-100-50	0.6837	0.8108	0.7785	0.7326
14	20-40-40-20	0.8990	0.8445	0.8093	0.8739
15	10-20-20-10	0.8977	0.9602	0.9015	0.9109
16	10-20-20-20-10	0.9304	0.9341	0.9456	0.9329

表 4.2 中給出了以四天作為一次預測值的 MLP 配置。作者將納斯達克指數 99 天的數據隨機分為訓練集（60%）、驗證集（20%）和測試集（20%）。在初步研究的基礎上，訓練方法和傳遞函數分別為 OSS 和 TANGSIG。通過對 hidden layer 數和每個 hidden layer 中的神經元數應用不同的值來優化神經網絡的架構。

生成、訓練和測試了 16 個具有不同架構的網路。計算訓練集、驗證集和總數據的 R^2 值，但僅考慮驗證的 R^2 值來選擇優化的網路架構。

如表 4.2 所示，當 hidden layer 數為 2 且 hidden layer 中的神經元數為 2 時， R^2 具有期望值（最大值）40。值得注意的是，神經元數量的任何變化都會影響模型的熟練程度。例如表 4.2 所示，雖然具有 5-5 個神經元的網路具有可接受的 validation (0.8631)，但具有 5-10 個神經元的網路具有較差的預測能力。

與四天為單位做預測類似，已經生成並測試了不同訓練算法的 R^2 值和具有 20-40-20 個神經元 hidden layer 的 MLP 傳遞函數。應用 OSS 訓練方法和 LOGSIG 傳遞函數根據 validation 數據集的值 (0.9622) 得到優化的訓練網路。

表 4.3 在不同結構的 BPNN 下以九天作一次預測的 R^2 值

No.	Structure	R^2			
		Train	Test	Validation	Total
1	5	0.8195	0.8480	0.8425	0.8274
2	10	0.8025	0.7113	0.8608	0.7882
3	5-5	0.8437	0.7762	0.8518	0.8280
4	5-10	0.8127	0.8554	0.8262	0.8180
5	20-20	0.8344	0.9116	0.7524	0.8389
6	20-30	0.8859	0.8690	0.8756	0.8707
7	50-100	0.8335	0.9028	0.8300	0.8292
8	200-300	0.8473	0.7899	0.8938	0.8506
9	300-400	0.8476	0.8685	0.7988	0.8435
10	20-50-20	0.8648	0.7937	0.8809	0.8372
11	20-40-20	0.9318	0.8827	0.9622	0.9262
12	50-100-50	0.8697	0.8367	0.8388	0.8552
13	20-40-40-20	0.7761	0.9205	0.8601	0.8188
14	20-50-50-20	0.8359	0.8462	0.8977	0.8443
15	10-20-20-20-10	0.8578	0.8348	0.6704	0.8107

如上表 4.3 中給出了以九天為一次做預測值的 MLP 的幾種配置。訓練方法和傳遞函數分別是 OSS 和 LOGSIG。我們發現具有三個 hidden layer 和 20-40-20 個 hidden layer 神經元的網路是最優化網路。根據表二和三，利用前四個工作日和前九個工作日作為輸入參數的預測能力沒有明顯差異。

4.2 實驗方法

4.2.1 SVM

本組利用 sklearn 之 SVM 套件，且由於資料是連續值，因此選用 SVR 進行實驗，並根據本篇論文將前 70 天的資料集樣本個數作為訓練樣本，其餘 29 天的資料集樣本作為測試樣本。

固定參數部分，皆採 sklearn 套件之預設參數，kernel 採用 RBF，C 設定為 1000，gamma 為 0.1。

此外，本組在 SVM 部分採用原資料之 date 與 price 進行訓練。

以下是本組 SVM 作法：


```
svr_rbf = SVR(kernel= 'rbf', C= 1e3, gamma= 0.1)
svr_rbf.fit(dates, prices)
```

4.2.2 LSTM

由於資料是具有時間性的，因此選用 LSTM 來做實驗，本組使用 keras 之 LSTM 套件，並根據本篇論文來將前 70 天資料個數來做訓練集，其餘 29 天來做測試資料集。

在模型參數的部分，我們設定 time step 為 4，設定此值是為了跟論文設定 4 天為一個時間預測單位，並且此 LSTM 採用 2 層隱藏層，第一層 150 個神經元，第二層 60 神經元，其 epochs=1000，並且 batch_size =5，最後用 mean-squared 來做梯度下降，如下圖所示：

```
#LSTM model
model = Sequential()
#Adding the first LSTM layer and some Dropout regularisation
model.add(LSTM(units = 150, return_sequences = True, input_shape = (X_train.shape[1], 1)))
model.add(Dropout(0.2))
# Adding a second LSTM layer and some Dropout regularisation
model.add(LSTM(units = 60))
model.add(Dropout(0.2))
# Adding the output layer
model.add(Dense(units = 1))
# Compiling the LSTM
model.compile(optimizer = 'adam', loss = 'mean_squared_error')

# Fitting the LSTM to the Training set
model.fit(X_train, y_train, epochs = 1000, batch_size = 5)
```

4.3 結果比較

圖 4.1 為使用 SVM 之預測結果與原資料進行比較，其結果之 R^2 值為 0.9776。

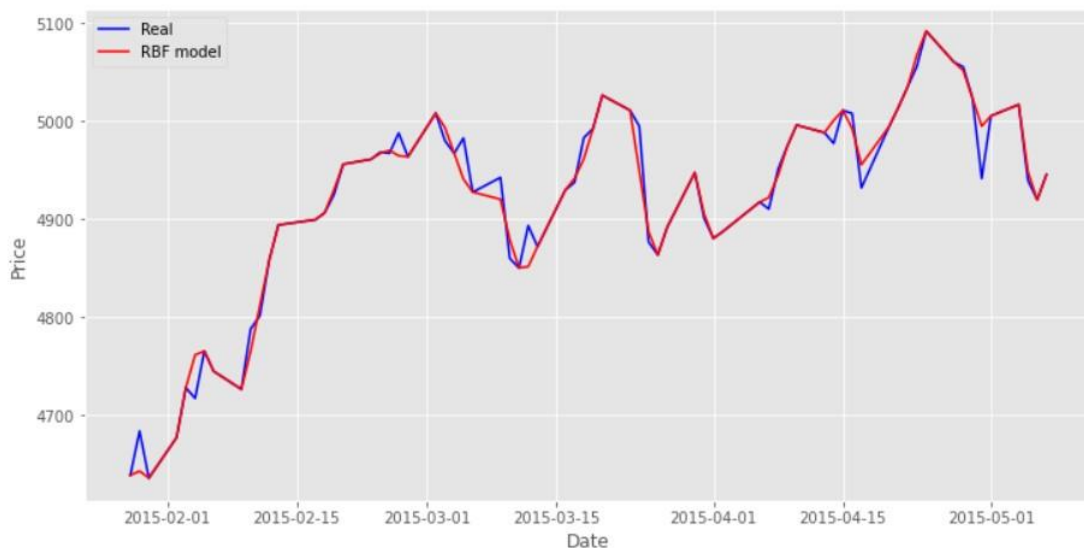


圖 4.1 SVM 結果

圖 4.2 為使用 LSTM 之預測結果與原資料進行比較，並且此為全部的股市資料，其結果如下：

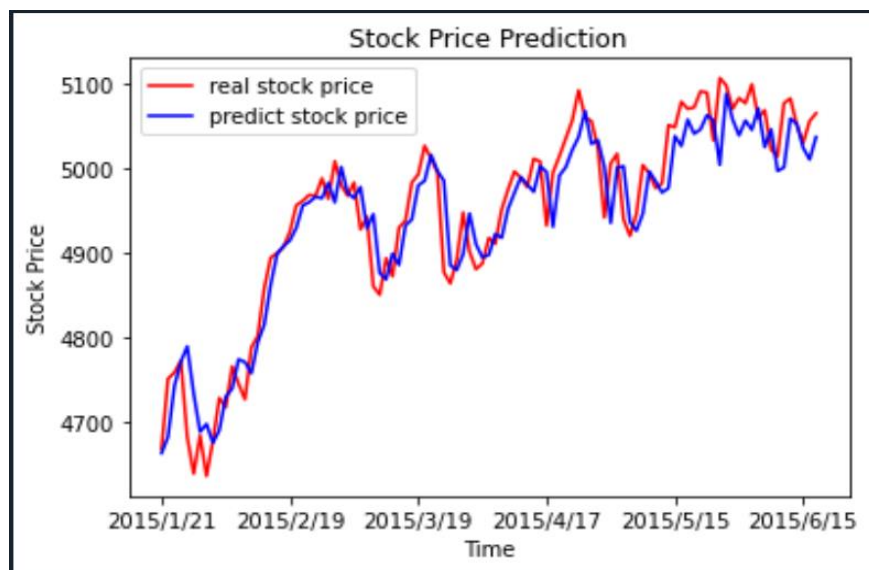


圖 4.2 LSTM 所有資料結果

圖 4.3 為使用 LSTM 之預測結果與原資料進行比較，此資料為測試資料集，其 R-score 為-0.208 其結果如下：

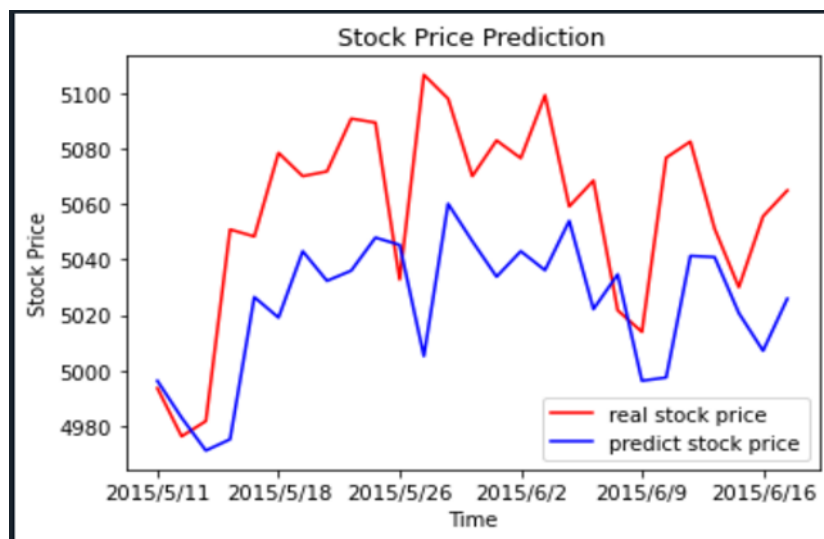


圖 4.3 LSTM 測試資料結果

第五章 結論

在這篇論文中，他所選取的資料是 NASDAQ 指數 2015 年 1 月 28 號到 2015 年 6 月 18 號之指數資料，總共 99 筆資料，希望運用類神經網路的方法來預測指數，原先論文所使用的方法為 MLP 且選擇 OSS 和 TANSIG 作為訓練方法和傳遞函數，其效果良好。隨後在本組所使用兩種方法 SVM 與 LSTM 中，SVM 所達到的效果是良好的，但可能有過度擬合的問題存在；而在 LSTM 的模型中，預測的效果則不如預期，在圖 4.2 及圖 4.3 中我們可以看到其測試結果較不如 SVM，由此我們可以發現在這種少量資料的預測當中，若沒有過度擬和，則 SVM 以及簡單的 MLP(論文方法)其效果可能比 LSTM 良好。

參考文獻

1. Moghaddam, A. H., Moghaddam, M. H., & Esfandyari, M. (2016). Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science*, 21(41), 89-93.