Paper Title\* **(**use style: ***paper title***)

Chi Feng Lina and Chunshien Lib

Information Management. National Central University

National Central University, Taoyuan, TAIWAN

aEmail: j8888888871@yahoo.tw

bEmail: [jamesli@mgt.ncu.edu.tw](mailto:jamesli@mgt.ncu.edu.tw) (corresponding author)

*Abstract*

時間序列的預測是一個很廣泛且重要的研究議題。目前最為廣泛被使用的分析模型為類神經模糊系統，因此我們提出了球式複數神經模糊系統 (Sphere Complex Neuro-fuzzy System, SCNFS)來對時間序列進行預測。透過一般的複數類神經模糊系統 (Complex, Neuro-fuzzy system, CNFS)，我們可得到一組複數型態的輸出，實數部分和虛數部分可針對不同目標做預測。本篇的模型用球式模糊集合(Sphere Complex Fuzzy Sets, SCFSs)取代原本CNFS中的複數模糊集 (Complex Fuzzy Sets, CFSs)，得到的歸屬程度仍然是複數型態的值，但是能擁有多組輸出，即可同時針對兩個以上目標預測。資料前處理部分採用多目標特徵挑選，篩選出對所有目標最有利的特徵，並以此作為模型輸入，以降低模型整體運算負擔。在模型方面，採用SCFSs以及知名的Takagi–Sugeno的線性函式，兩者透過箭靶類神經層(Aim Object Neural Layer)結合，使得兩層可有不一樣的神經元個數。機器學習部分，為了最佳化模型的預測結果，我們利用分治法，SCNSs使用粒子群演算法 (Particle Swarm Optimization, PSO)進行參數優化，T-S線性函式則使用 (Recursive Least Square Estimation, RLSE)進行參數優化。最後我們將透過三個實驗與不同的方法做效能比較。

Keywords— Sphere complex neurofuzzy system (SCNFS); complex fuzzy set (CFS); complex neurofuzzy system (CNFS); time–series forecasting.

# Introduction (rewite)

時間序列資料是利用時間先後作為排序依據的集合，如股市、匯率、能源耗損等等。時間序列間的間隔為一恆定值，故可視為離散時間數據，可以數理方法或是統計模型進行分析。隨著現今資訊科技的發展，不論是物聯網或是網路帶來的資料，這些資料量有著爆炸性的成長，導致資料量有了量級的轉變。而這些數據無法再以人類智慧處理，而是需要利用機器學習或是資料探勘等方式，找出隱藏於資料背後所隱藏的關聯性或是規則，以挖掘出其中的價值。

時間序列資料的預測被廣泛應用於多個領域，如，

人工智慧在現實中有很多的應用，像是財經方面的預測[1]，醫療急診方面[2]–[3]、入學預測[4]等。Patton[1] ……… 上述這些有一部分屬於時間序列的預測，時間序列的預測是一個很重要的議題，因為他在真實世界中的應用範圍非常廣。過去已經有很多學者提出不同的方法來針對時間序列的預測問題，像是ARIMA[5][6]、模糊理論、神經網路運算、神經模糊混合系統等。其中，最常被提出的就是類神經模糊系統(Neuro–fuzzy systems, NFSs)[7]–[10]。

NFS系統一直是被廣泛研究的模型，其中，類神經網路系統有所謂的IF–THEN規則，這些規則如同我們人類的經驗法則。通常可以將IF–THEN的規則結合模糊理論，使整體的架構更有彈性，我們稱之為神經模糊混合系統。如上述所提，神經模糊混合系統的特性使他對時間序列的預測有著不錯的效果。所以現在有關時間預測這方面的研究，大多採用類神經網路為模型架構。

本實驗是採用神經模糊混合系統的作法，將整體模型效仿IF–THEN規則，建造成多層神經元架構。為了使模型彈性增加，有別於傳統的IF–THEN規則方法，本實驗在前鑑部與後鑑部間，採用箭靶層 (Aim object neural layer)連接，使得前鑑部與後鑑部個數得以不相同。在模型實作方面，我們將模糊理論結合類神經網路系統，形成神經模糊混合系統，前鑑部使用高斯型態的球式複數神經模糊集，後鑑部則使用Takagi–Sugeno的線性函式[17]，前鑑部和後鑑部則透過箭靶層結合。透過此模型和機器學習我們預期對時間序列的預測可以更加精準。

關於模糊集，在1965年由Zadeh學者，最先提出模糊集的概念[11]，使資料可以透過某個函式，得到介於0到1之間的歸屬程度 (Membership degree)。2002年，Ramot et al. [12] 提出了複數模糊集 (CFSs)[12]的概念，歸屬程度可以透過函式得到一個複數型態的值，這使得歸屬程度可以呈現在一個半徑為1的複數單位圓盤中。這個概念使原本能表示的歸屬程度更加的豐富。一般來說我們可以透過複數型神經模糊集系統 (CNFS)[13][14]，得到一組複數型態的輸出，而實數部分和虛數部分可針對不同目標做預測，所以可針對兩個不同目標。目前兩個目標的預測已經有很多的研究產出[6][13][14]。而為了同時對更多的目標做預測，本篇論文將原本複數型神經模糊集系統進行改良，將原本使用的複數模糊集 (CFSs)改成球式複數模糊集 (SCFSs)，其中歸屬程度仍然是複數型態，但能呈現在3–D的立體空間中，能擁有更多組複數型態的輸出，意即可以一次對多個目標進行預測。

在本研究中，為了使資料可以有效的被應用，在資料前處理的部分，我們根據將原始資料的30個漲跌值作為特徵，並透過夏農資訊熵 (Shannon Entropy)[15]，去計算他們個別對目標的資訊貢獻量，除此之外，我們透過了多目標特徵選取的概念[16]，算出每個特徵對目標的有效資訊量，以此作為挑選訓練資料的依據。從資料中萃取出最有效的資料，除了降低模型的運算負擔，也能有效的提升預測的效能。最後，機器學習部分，我們使用知名的Particle Swarm Optimization (PSO)[18]和廣為人知的Recursive Least Square Estimator (RLSE)[19]進行參數優化，並將它們整合成一種方法我們稱之為PSO–RLSE方法[20]。我們將前鑑部和後鑑部參數藉由不同的演算法訓練，想透過divide–and–conquer原理，降低搜尋的維度，使模型更容易找到最佳解，提高整體的效能。Use a paragraph to specify the structure of paper.

# Methodology

## Sphere Complex Fuzzy Sets

過去的模糊集合概念，可以導出一對一的歸屬程度，而複數型態的模糊集合[12]，可以擁有更豐富的歸屬程度，但為了使應用更廣泛，我們希望可以透過一個觀念使得歸屬程度更加的豐富。

球式複數模糊集合是本論文所提出的原創概念，透過此概念可以將一筆資料轉換成多個複數型態的歸屬程度，以便之後模型可以一次預測多個目標。首先，將原高斯函數得到的歸屬程度置放於半徑為1的球式複數模糊集合內(Fig. 1)，即可得到一組空間向量:







其中，為原高斯函數(17)的歸屬程度;;。藉由的拆解，可得到至少四組的複數型態歸屬程度，包含了保維法得到的歸屬程度:



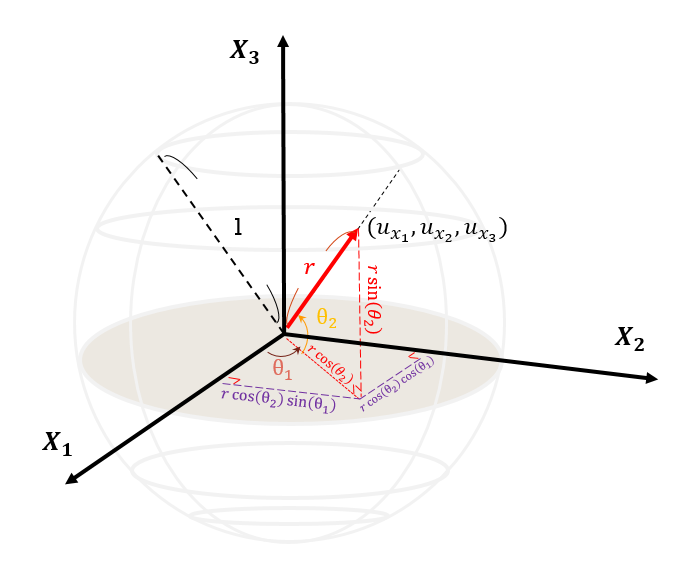
以及降維法得到的歸屬程度:







其中，。



1. 球式複數模糊集。為高斯函數的歸屬程度，可透過與平面的夾角及計算出在各維度上的投影。

## Multi–Target Feature Selection

特徵挑選不僅能刪去負面的資訊來源，更有助於減輕模型的運算負擔，故是資料前處理中重要的一環。而面對多個目標時的特徵挑選，更需要謹慎的處理，才能帶來正面的效果。本論文將同時預測多個目標，故使用夏農資訊熵[15]概念，並參考多目標特徵選取方法[16]，實作特徵挑選，最後從挑選後的特徵中取得訓練資料。

熵一詞最早是由德國物理學家Clausius於 1854 年提出[28]，是一種對物理系統之無秩序或亂度的量度。在1948年，學者Shannon則提出了資訊熵[15]的概念，熵定義為資訊內容其不確定性的量值，若資訊的隨機性越高，則資訊熵值會越高。對於某一個隨機變數*X*，資訊熵[15]的定義如下。

8

其中，H(X)是隨機變數X的資訊熵;是事件的發生機率;則被視為的資訊混亂度。

但若大於1，則部分會是負數，會影響到整體的期望值，所以我們對公式做了一些更改，更改後的公式如下:

9

10

其中，為很小的正值。

由於我們是特徵的選擇是針對目標，所以我們透過資訊熵的概念，計算每個特徵與目標之間的影響資訊量(Influence Information)[16]，公式如下:

11

其中，為特徵對目標的影響資訊量;為特徵為正數以及目標的互資訊[16];為特徵為負數以及目標的互資訊;互資訊的定義公式如下:

12

13

其中，為目標的資訊熵;為特徵為正數時所對應的目標的資訊熵;為特徵為負數時所對應的目標的資訊熵;而條件式資訊熵公式如下:

14

15

16

17

其中，為很小的正值，為特徵為正數時的機率密度，為特徵為正數時所對應的目標的機率密度，為特徵為負數時的機率密度，為特徵為負數時所對應的目標的機率密度。

透過上述影響資訊量[16]的公式，可以得到每個特徵變數對每個目標的影響資訊量，為了方便使用影響資訊量，我們利用特徵與特徵之間以及對於第個目標的影響資訊量整理出影響資訊矩陣(Influence Information Matrix, IIM)。如下:

18

其中，為第個特徵變數;為第個目標變數; n為特徵的總個數;為特徵變數對特徵變數的影響資訊量。

而後可依據這些影響資訊量做多目標的特徵選取，步驟如下:

Step 1 : 算出第個特徵對第個目標的資訊增益量(selection gain)標記為，其中，為第個特徵變數;為第個目標變數。資訊增益量公式如下:

19

其中，為對的影響資訊量;為第個已選特徵池(Selected Pool, SP);為對中已存在特徵的冗餘資訊量。冗餘資訊量公式如下:

20

其中，代表第個已選特徵池內的特徵個數;為對內的第個特徵變數的影響資訊量;為內的第個特徵變數對的影響資訊量。經過上述計算若大於0，則將特徵加入第個已選特徵池 中。

Step 2 :無論重疊與否，將所有已選特徵池中出現過的特徵變數記錄下來，儲存成。其中，為目標變數個數;，是中第個特徵變數。計算每個特徵出現在所有SP的次數，標記為。

Step 3 :透過即可計算覆蓋率，公式如下:

21

計算的平均，標記為。

Step 4 :累加每個SP裡，特徵的資訊增益量:

22

計算的平均標記為。

Step 5 :根據累加後的資訊增益量和覆蓋率，計算出特徵的有效貢獻量:

23

Step 6 :測試中所有的特徵變數，若，則將累加。

Step 7 :設定上下界，標記為和，透過上下界找出，表示最後選取的特徵數目。本研究中所有實驗皆設定為4，皆設定為2。若介於上下界間，則將設定成;若小於下界，則將設定成;若大於上界則將設定成。

Step 8 :將排序，並選取前個特徵變數加入最後特徵池(Final Pool, FP)中，當作多目標的特徵挑選結果。

## Structure Learning

結構學習是為了將訓練資料可以更有邏輯的應用到模型建造中。經過多目標特徵選取後可以得到一些被挑選出的特徵，我們可以將所有特徵變數以及目標組合成一個資料矩陣(Data Matrix, DM)，如下:



其中，為第一組輸入變數，即為挑選出的特徵變數，，為一個特徵的總筆數，為輸入維度的數量，及為第0層的神經元數量; 為第一個目標變數，，為一個目標的總筆數，為目標的總數量。

在本研究中，會將這些不同輸入維度的訓練資料，透過減數分群演算法[27]進行分群。並將分群後的群中心配合每個維度的標準差形成模糊集，各個維度的模糊集個數總和即為第1層神經元的數量，本研究採用高斯型態的模糊集合，高斯函數的公式如下:



其中，為輸入變數，和為群中心和標準差的參數。基於各個輸入維度的模糊集，我們可以得到個區塊，已第個區塊為例，如下:

*:*

26

其中，為第個輸入的語意變數;為第個輸入變數，;為第個區塊中第個輸入語意變數的模糊集。



1. 模糊集合輸入空間(2維)。兩個輸入維度，各分出3群則會形成共9塊區域，其中z軸為該區的資料密度。

為了模型的運算效率，以及降低模型運算負擔，我們將會篩選位於第2層中的神經元數量，即為神經元的數量。若以兩個輸入維度為例子可得到如圖(Fig. 2)的區塊，而後將透過資料密度的概念，將資料灑入區塊中，並計算資料密度量，用以決定神經元數量。步驟如下:

Step 1:從各個輸入維度與模糊集可以得到每個區塊的資料密度量，以第個區塊為例，資料密度量公式如下:

27

其中，為第個輸入維度的第筆資料;為第個前鑑部中第個輸入維度的模糊集。

Step 2:將每個區塊的資料密度量累加標記為，公式如下:

28

其中，為資料總筆數。計算平均值標記為，標準差標記為。

Step 3:查看每個區塊，若，則將累加。設定上下界，標記為和，透過上下界找出，表示最後選取的神經元數目。本研究中所有實驗皆設定為15，皆設定為4。若介於上下界間，則將設定成;若小於下界，則將設定成;若大於上界則將設定成。

Step 4:將排序，並保留前個前鑑部，當作之後模型的神經元。

第3層的神經元數目將由訓練資料集合進行減數分群[27]決定，決定神經元數量後，將透過Fuzzy C–Mean分群方法[]決定第3層神經元–箭靶神經元的相關數值，步驟如下:

Step 1:將輸入資料集合進行減數分群[27]，決定箭靶神經元個數為，即為第3層神經元的數目。

Step 2: 決定群數之後，透過Fuzzy C–Mean對目標集合進行分群，可以得到個群中心以及標準差。

Step 3: 由於本研究採用球式複數模糊集合，箭靶又會承接上一層的輸入，故每個箭靶都會有許多層，以接收輸入向量中的每個值。其中，第層的第群中心標記為}，第層的第群標準差標記為{, }。用得到的群中心以及標準差製作箭靶，箭靶的製作以及詳細公式將會在下一小節討論。每個箭靶神經元後面連接著一個T–S神經元，T–S神經元為T–S function構成，T–S function公式如下:

29

其中，{}是第個T–S神經元的參數，是第個輸入。

## Model Structure and I/O Relationship

本研究的模型為一個六層的類神經網路。訓練資料集合標記為，為資料總筆數，是的輸入向量，為輸入維度數量;為的目標向量，為複數型態目標的數量。透過模型可以得到輸出。

Layer 0:這層為輸入層，是原始資料透過多目標特徵選取後，將最後挑出的特徵當作訓練資料，我們將時間序列第個點的輸入向量標記為:



Layer 1:這層為球式複數模糊集合層，透過前面結構學習的分群，可在不同維度上建構數個模糊集，而每個不同維度的輸入都可經由模糊集得到歸屬程度。透過球式複數模糊集合可得到多組複數型態的歸屬程度，不同的歸屬程度可以給不同的模型輸出做應用，以達到多目標預測的效果，透過球式複數模糊集的公式(??)，從中可得到歸屬程度的向量:



Layer 2:經過前面的結構學習，我們可以篩選出個對模型較有用的神經元，神經元的輸入為上一層的歸屬程度，將每個輸入維度的歸屬程度相乘後，可得到每個神經元的啟動強度，由於本研究採用球式複數模糊集合，故每個神經元的輸入會是向量型態，輸出亦然如此:



其中，為第個神經元的各個輸入向量中第項的乘積，。

Layer 3: 箭靶是用以承接上一層的輸出，為向量的型態，而本層的輸出亦是向量的型態，如下:

33

其中，為第個神經元射在第個箭靶第層的值，。本研究中使用的是球式複數模糊集，因此輸入會是複數型態，故輸出也要在複數單位圓盤中，因此箭靶需進行轉換，以確保箭靶層輸出也是複數型態，如下:







其中，。為第個箭靶轉換後的中心，為第個箭靶轉換後的靶寬，公式如下:









**41**



其中，為的變數;為的變數;為第個目標的平均;為第個目標的標準差。

Layer 4:此層為T–S層，經過此層的運算可以得到個模型輸出，公式如下:





其中，{}是第個T–S神經元的參數。













Layer 5:此層為輸出層，將上一層得到的個模型輸出加總，即為我們的模型輸出:



## Parameter Learning

根據分治法(divide and conquer)的概念，我們將使用不同的機器學習演算法，對各層的參數優化，以便更容易找到最佳解。對於第一層模糊集的參數優化，我們使用知名的粒子群演算法(Particle Swarm Optimization, PSO)演算法[18]學習，其原理為模擬鳥群在尋找食物，每回合透過自身的最佳位置和全群最佳位置調節速度，特性為收斂快速，演算法公式如下:

52

53

其中，為第回合時第個粒子的位置，為第回合時第個粒子的速度，為第回合時第個粒子的最好位置，為第回合時全部粒子中最好的位置，為PSO的參數，、為介於0到1的隨機數。在本研究中，粒子的位置代表模糊集中的參數，其中包含了每個維度的分群中心、標準差以及相位頻率函數、。

在本論文中遞迴式最小平方演算法(Recursive Least Square Estimation, RLSE)[19]是用來更新T–S神經元參數，一般來說LSE問題可以被視為一個線性的問題，如下:



其中，y是目標;u是模型的輸出;{}是u已知的方程式;{, =1,2,…,m}是我們估計的未知參數，則是整個模型的誤差LSE的問題也可以被寫成矩陣的方式表達，如下:



其中:









是輸入的矩陣，是我們估計的未知參數矩陣，是目標矩陣，是誤差的向量。要最佳化，可透過RLSE的等式[19]運算:





其中，是遞迴次數，{}，是的第行，再開始RLSE演算法時，會設定為0，則設定為，為一極大整數，為單位矩陣。

PSO–RLSE混合演算法的流程如下:

Step 1:準備訓練資料及測試資料。

Step 2:以PSO粒子位置作為模糊集參數，將訓練資料帶入模型，並計算每個神經元啟動強度。

Step 3:用RLSE更新神經元的參數，RLSE算式中的和向量如下:







其中。因為多目標預測中，啟動強度為一向量，且為一矩陣，因此在原本的公式(??)中，利用單位矩陣取代原本的常數項1，改良後公式如下：



Step 4:更新完所有參數後，計算出模型的輸出。

Step 5:計算成本，更新PSO粒子自身的最佳位置和全群最佳位置。本研究成本函數(cost function)使用均方根誤差(Root Mean Square Error, RMSE)，定義如下:



其中，為模型第筆資料的誤差;為埃爾米特共軛(Hermitian transpose)，意即轉置矩陣後，並對矩陣元素做共軛運算。

Step 6:對所有PSO粒子重複Step 2~Step 5，直到PSO迭代結束。

# Experimentation

## Example 1—Quadruple Time Series of Daily National Association of Securities Dealers Automated Quotation Composite Index

在這個範例中，我們使用真實世界的時間序列數據來驗證模型的效能。使用的資料為National Association of Securities Dealers Automated Quotation(NASDAQ)每天的開盤和收盤價以及Standard and Poor’s (S&P 500)每天的開盤和收盤價。為了和其他論文比較效能，NASDAQ採用2007年1月3號至2010年12月20號的開盤和收盤價當作第一個複數型態的目標，總共998筆資料，而S&P 500的部分則以2010年12月20號往前取998筆開盤和收盤價當作第二組複數型態的目標。本篇論文所提出的模型可以一次有多個複數型態的輸出，故可以預測多個目標，我們這個範例中使用的第一個複數目標實數部分為NASDAQ的開盤價，複數部分則為NASDAQ的收盤價，第二個目標實數部分為S&P 500的開盤價，複數部分為S&P 500的收盤價。由於本實驗中有做多目標特徵選取[16]，故此範例的原始資料為1029筆，經過一次差分得到1028筆，並以每日漲跌從4組原始資料取出30個特徵，共有120個特徵，每個特徵會有998筆資料，經過篩選後的每個特徵前500筆資料為訓練資料，剩下的為測試資料。

範例一模型設定

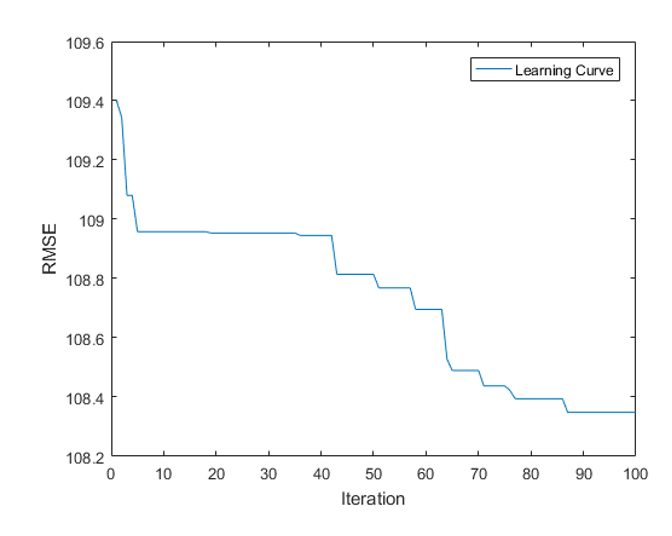
|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | {} |
| Number of fuzzy sets of each input | 2、3 |
| Number of targets (real–valued) | 4 |
| Number of outputs (complex–valued) | 2 |
| Number of neurons(before selection) | 9 |
| Type of neurons | Sphere complex fuzzy set |
| Number of neurons (after selection) | 4 |
| Number of neuron parameters | 20 |
| Number of aim object | 3 |
| Type of T–S neurons | Takagi–Sugeno |
| Number of T–S neurons | 3 |
| Number of T–S neuron parameters | 9 |

結構學習部分，則將每個被挑選的特徵，用減數分群演算法[27]分群。並透過第二章所介紹的神經元挑選方法，從原本的9個神經元篩選到剩下4個神經元。整體模型在結構學習後的參數，如表I所示。PSO–RLSE混合方法的機器學習參數設定，如表II所示。

機器學習參數設定

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 64 |
| Iterations | 100 |
|  | 0.8 2.0 2.0 |
|  | Random in [0,1] |
| Initial position | Given by the subtractive clustering algorithm |
| Initial velocity | 0 |
| **RLSE** | |
|  |  |
|  | 201 zero vector |
|  | **I** |
| **I** | 2020 identify matrix |

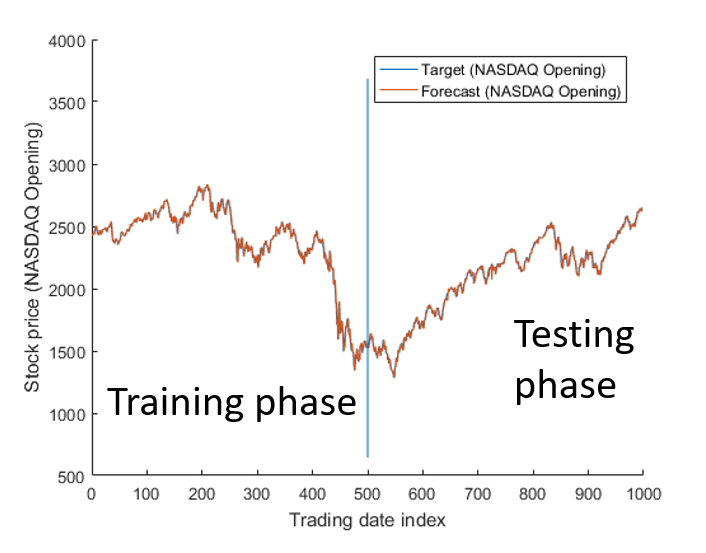
本次範例的結果將與其他文獻[6]所提的方法做比較。其中除了SVR之外的模型，皆可以同時預測兩個實數目標。所以我們將使用模型的第一組複數型態的輸出與其他文獻做比較，結果如表III所示。模型的機器學習曲線，如Fig. 3所示;目標與模型輸出的結果，如Fig. 4所示;誤差圖如Fig 5所示。



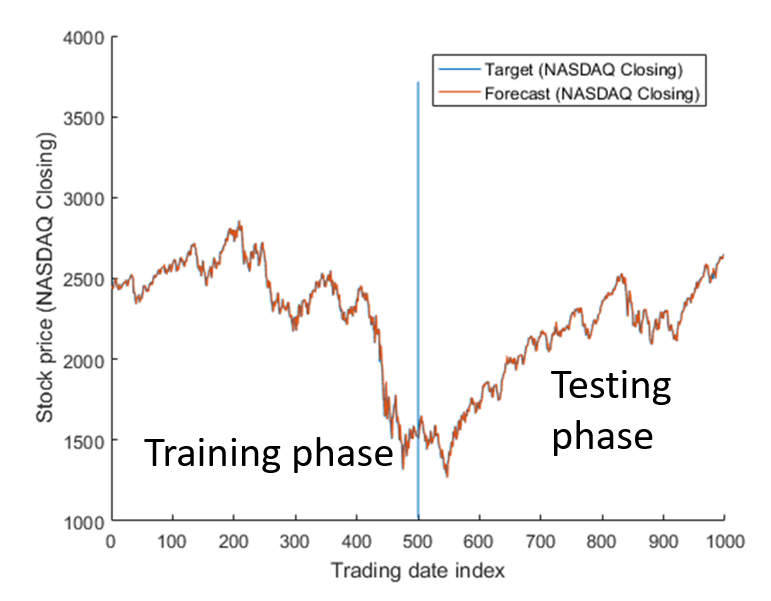
1. 範例一學習曲線。可從此曲線看出RMSE的變化量，x軸為機器學習迭帶，y軸為RMSE。

NASDAQ 雙目標預測效能比較表(RMSE)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **RMSE** | | | | |
| Training phase | |  | Testing phase | |
| Opening index | Closing index |  | Opening index | Closing index |
| SVR (two models, each with single output) [6] | 35.18 | 35.24 |  | 37.23 | 40.24 |
| ANFIS (two models, each with single output) [6] | 37.83 | 38.66 |  | 38.80 | 42.36 |
| ANFIS (one model with two outputs) [6] | 62.75 | 71.51 |  | 72.52 | 85.08 |
| RBF (two models, each with single output) [6] | 37.59 | 33.89 |  | 37.52 | 44.08 |
| RBF (one model with two outputs) [6] | 178.57 | 179.87 |  | 261.37 | 258.89 |
| CNFS(5)–ARIMA (one model with two outputs) [6] | 21.56 | 20.81 |  | 32.52 | 33.70 |
| SCNFS(proposed) | 38.58 | 38.59 |  | 27.79 | 27.83 |

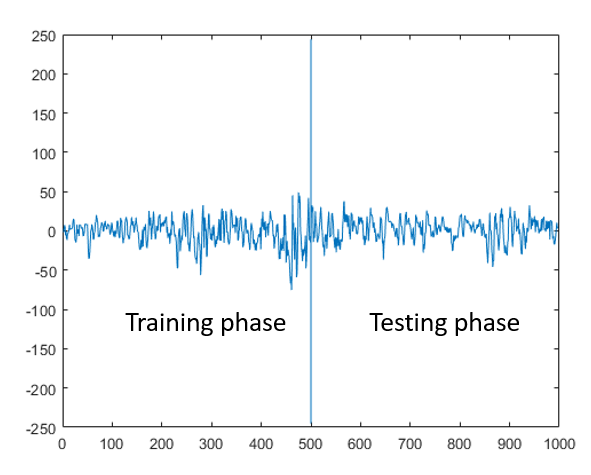


(a)



(b)

1. NASDAQ的實際數值和模型輸出(a)每日開盤價(b)每日收盤價。x軸為交易日，y軸為股票價格，可看出訓練和測試階段皆有不錯的效果。



1. 範例一誤差圖。可從圖中看出每個交易日，模型預測與實際數據的差異。

十次測試效能表(實驗一)

|  |  |  |
| --- | --- | --- |
|  | Performance(RMSE) | |
| Trials | Opening index | Closing index |
| 1 | 29.56 | 28.49 |
| 2 | 130.08 | 109.50 |
| 3 | 27.79 | 27.83 |
| 4 | 30.52 | 30.46 |
| 5 | 28.16 | 27.75 |
| 6 | 33.86 | 29.16 |
| 7 | 45.77 | 33.34 |
| 8 | 28.20 | 28.53 |
| 9 | 28.54 | 28.93 |
| 10 | **27.88** | **27.56** |

## Example 2—Quadruple Time Series of Daily Dow Jones Industrial Average Index

在這個範例中，我們也是對現實世界的時間序列數據來驗證模型的效能，與範例一不同的是，四個目標之間不是收盤價與開盤價的關係，意即目標的曲線相似度沒有像範例一目標曲線的相似度高。範例二的四個目標使用的資料為The Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX)、Dow Jones Industrial Average Index (DJI) 、National Association of Securities Dealers Automated Quotation (NASDAQ)、Standard & Poor's 500 (S&P500)，每年的收盤價。為了和其他論文比較效能，我們對2001年至2004年的收盤價做預測，將訓練資料做一次差分，並以每日漲跌從每組原始資料取出30個特徵，4個目標共有120個特徵。每年收盤價筆數如下，2001年資料為245筆，2002年資料為248筆，2003年資料為249筆，2004年資料為250筆，每年的四目標做一次預測，每年的前十個月的資料當作訓練資料，剩餘的當作測試資料，2001年測試資料為181筆，2002年為184筆，2003年為185筆，2004年為185筆。本篇論文所提出的模型可以一次有多個複數型態的輸出，故可以預測多個目標，我們在範例中使用的第一個複數目標實數部分為TAIEX當年的收盤價，複數部分則為DJI當年的收盤價，第二個目標實數部分為NASDAQ當年的收盤價，複數部分為S&P500當年的收盤價。

結構學習部分，則將每個被挑選的特徵，用減數分群演算法分群[27]。並透過第二章所介紹的神經元挑選方法，減少神經元數目。整體模型在結構學習後的參數，如表V至表VIII所示。PSO–RLSE混合方法的機器學習參數設定，如表IX所示。

範例二模型設定(2001年)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | { } |
| Number of fuzzy sets of each input | 3、4、3、4 |
| Number of targets (real–valued) | 4 |
| Number of outputs (complex–valued) | 2 |
| Number of neurons (before selection) | 144 |
| Type of neurons | Sphere complex fuzzy set |
| Number of neurons (after selection) | 15 |
| Number of neuron parameters | 56 |
| Number of aim object | 3 |
| Type of T–S neurons | Takagi–Sugeno |
| Number of T–S neurons | 3 |
| Number of T–S neuron parameters | 15 |

範例二模型設定(2002年)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | { } |
| Number of fuzzy sets of each input | 3、3、3、3 |
| Number of targets (real–valued) | 4 |
| Number of outputs (complex–valued) | 2 |
| Number of neurons (before selection) | 81 |
| Type of neurons | Sphere complex fuzzy set |
| Number of neurons (after selection) | 13 |
| Number of neuron parameters | 48 |
| Number of aim object | 5 |
| Type of T–S neurons | Takagi–Sugeno |
| Number of T–S neurons | 5 |
| Number of T–S neuron parameters | 25 |

範例二模型設定(2003年)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | { } |
| Number of fuzzy sets of each input | 3、3、3、3 |
| Number of targets (real–valued) | 4 |
| Number of outputs (complex–valued) | 2 |
| Number of neurons (before selection) | 81 |
| Type of neurons | Sphere complex fuzzy set |
| Number of neurons (after selection) | 11 |
| Number of neuron parameters | 48 |
| Number of aim object | 3 |
| Type of T–S neurons | Takagi–Sugeno |
| Number of T–S neurons | 3 |
| Number of T–S neuron parameters | 15 |

範例二模型設定(2004年)

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | { } |
| Number of fuzzy sets of each input | 5、4、4、4 |
| Number of targets (real–valued) | 4 |
| Number of outputs (complex–valued) | 2 |
| Number of neurons (before selection) | 320 |
| Type of neurons | Sphere complex fuzzy set |
| Number of neurons (after selection) | 15 |
| Number of neuron parameters | 68 |
| Number of aim object | 8 |
| Type of T–S neurons | Takagi–Sugeno |
| Number of T–S neurons | 8 |
| Number of T–S neuron parameters | 40 |

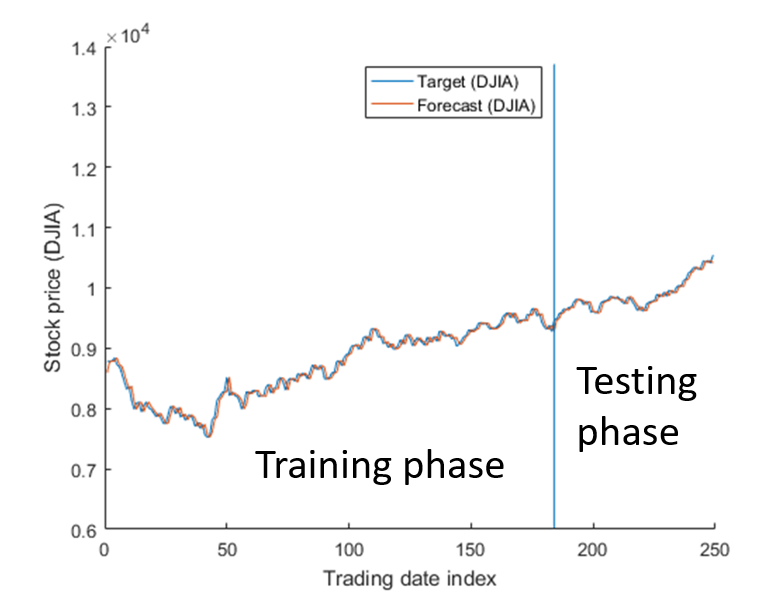
機器學習參數設定

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 64 |
| Iterations | 100 |
|  | 0.8 2.0 2.0 |
|  | Random in [0,1] |
| Initial position | Given by the subtractive clustering algorithm |
| Initial velocity | 0 |
| **RLSE** | |
|  |  |
|  | Given by Number of T–S neuron parameters |
|  | *D*1 zero vector |
|  | **I** |
| **I** | *DD* identify matrix |

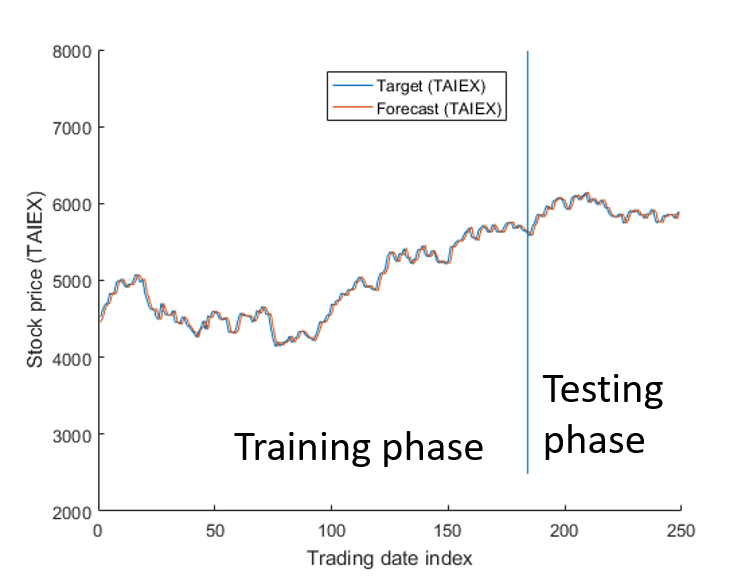
本次範例的結果將與其他文獻[6]所提的方法做比較。其中除了SVR之外的模型，皆可以同時預測兩個實數目標。所以我們將使用模型的第一組複數型態的輸出與其他論文做比較，結果如表XI和表XII所示。以2003年為例目標與模型輸出的結果，如Fig. 6所示;模型誤差圖如Fig. 7所示。

十次效能表(實驗二)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Performance(RMSE) | | | |
| Trials | 2001 | 2002 | 2003 | 2004 |
| 1 | 259.74 | 279.93 | **196.91** | 282.82 |
| 2 | 259.27 | 286.02 | 198.10 | 283.94 |
| 3 | 260.99 | 280.34 | 199.06 | 286.83 |
| 4 | 260.44 | 280.56 | 199.54 | 284.57 |
| 5 | 263.29 | 281.85 | 196.94 | 278.60 |
| 6 | 263.57 | 282.00 | 197.85 | 288.52 |
| 7 | 259.86 | 279.46 | 200.05 | 275.31 |
| 8 | 262.02 | 282.35 | 198.33 | 285.69 |
| 9 | 261.23 | 282.11 | 197.53 | 288.15 |
| 10 | 260.59 | 280.93 | 197.62 | 286.68 |

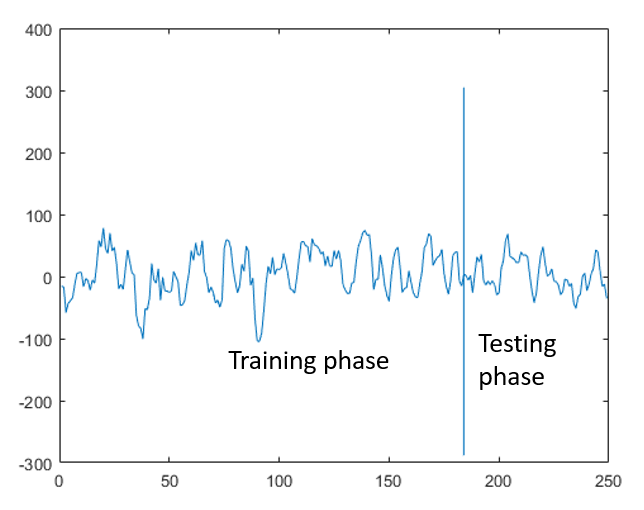


(a)



(b)

1. 範例二實際數值和模型輸出(a)DJIA (2003年) (b)TAIEX (2003年)。x軸為交易日，y軸為股票價格，可看出訓練和測試階段皆有著不錯的效果。



1. 範例二誤差圖。可從圖中看出每個交易日模型預測與實際數據的差異。

DJIA效能比較表(RMSE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method Year** | **2001** | **2002** | **2003** | **2004** |
| SVR (two models, each with single output) [6] | 101.44 | 117.95 | 82.76 | 71.49 |
| ANFIS (two models, each with single output) [6] | 105.56 | 111.69 | 72.09 | 68.00 |
| ANFIS (one model with two outputs) [6] | 128.20 | 142.05 | 90.37 | 83.69 |
| RBF (two models, each with single output) [6] | 106.33 | 131.24 | 97.58 | 81.79 |
| RBF (one model with two outputs) [6] | 181.79 | 136.28 | 154.14 | 148.11 |
| CNFS(5)-ARIMA (one model with two outputs) [6] | 103.06 | 103.42 | 70.70 | 66.55 |
| SCNFS(proposed) training phase | 91.95 | 98.69 | 69.66 | 97.99 |
| SCNFS(proposed) testing phase | 94.31 | 85.52 | 56.26 | 61.64 |

TAIEX效能比較表(RMSE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method Year** | **2001** | **2002** | **2003** | **2004** |
| SVR (two models, each with single output) [6] | 162.46 | 67.72 | 59.47 | 58.81 |
| ANFIS (two models, each with single output) [6] | 147.36 | 70.17 | 72.61 | 65.33 |
| ANFIS (one model with two outputs) [6] | 151.62 | 78.27 | 81.69 | 70.54 |
| RBF (two models, each with single output) [6] | 134.32 | 65.15 | 60.41 | 102.86 |
| RBF (one model with two outputs) [6] | 137.58 | 78.54 | 115.92 | 126.48 |
| CNFS(5)-ARIMA (one model with two outputs) [6] | 115.82 | 64.34 | 57.69 | 55.56 |
| SCNFS(proposed) training phase | 92.03 | 100.26 | 69.96 | 99.11 |
| SCNFS(proposed) testing phase | 89.59 | 86.81 | 55.34 | 60.30 |

## Example 3—Quadruple Time Series of Daily Taiwan Stock Exchange Capitalization Weighted Stock Index

在這個範例中，我們也是對現實世界的時間序列數據來驗證模型的效能，這次預測的目標為APPLE Computer Inc.、International Business Machines Corporation (IBM)、Dell Inc.、Microsoft Inc.，四個股票的收盤價，期間為2003年2月10號至2005年1月21號，總共為492筆，經過一次差分得到491筆，並以每日漲跌從4組原始資料取出30個特徵，共有120個特徵，每個特徵會有460筆資料。為了和其他論文比較效能，我們用2003年2月10號至2004年9月10號的資料當作訓練資料，共433筆，剩餘的當作測試資料，共59筆。本篇論文所提出的模型可以一次有多個複數型態的輸出，故可以預測多個目標，我們這個範例中使用的第一個複數目標實數部分為IBM的收盤價，複數部分則為APPLE的收盤價，第二個目標實數部分為Dell的收盤價，複數部分為Microsoft的收盤價。

結構學習部分，則將每個被挑選的特徵，用減數分群演算法[27]分群。並透過第二章所介紹的神經元挑選方法，從原本的81個神經元篩選到剩下9個神經元。整體模型在結構學習後的參數，如表XIII所示。PSO–RLSE混合方法的機器學習參數設定，如表XIV所示。

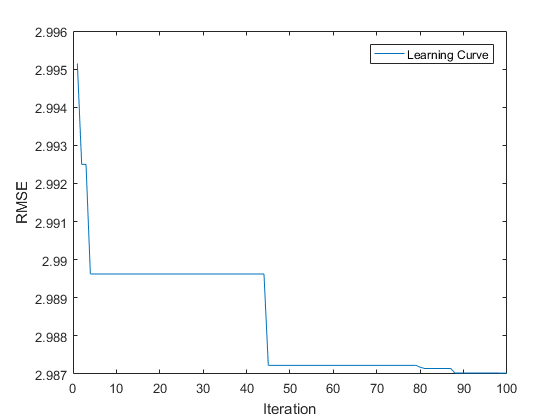
範例三模型設定

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | { } |
| Number of fuzzy sets of each input | 3、3、3、3 |
| Number of targets (real–valued) | 4 |
| Number of outputs (complex–valued) | 2 |
| Number of neurons (before selection) | 81 |
| Type of neurons | Sphere complex fuzzy set |
| Number of neurons (after selection) | 9 |
| Number of neuron parameters | 48 |
| Number of aim object | 3 |
| Type of T–S neurons | Takagi–Sugeno |
| Number of T–S neurons | 3 |
| Number of T–S neuron parameters | 15 |

機器學習參數設定

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 64 |
| Iterations | 100 |
|  | 0.8 2.0 2.0 |
|  | Random in [0,1] |
| Initial position | Given by the subtractive clustering algorithm |
| Initial velocity | 0 |
| **RLSE** | |
|  |  |
|  | 151 zero vector |
|  | **I** |
| **I** | 1515 identify matrix |

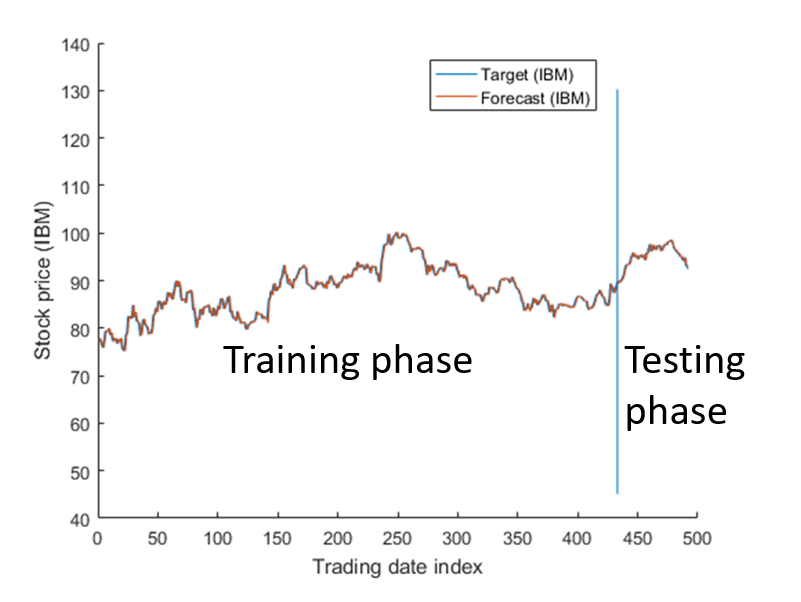
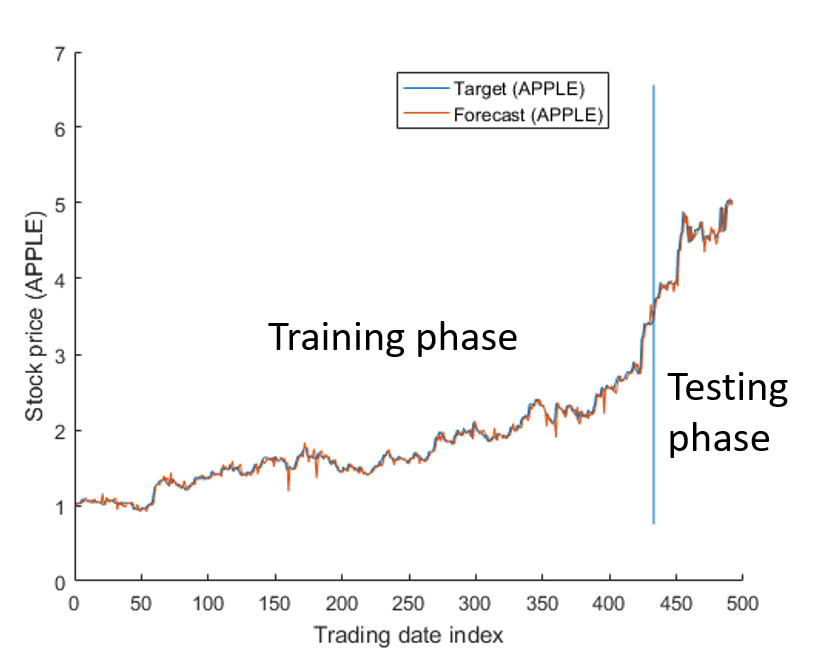
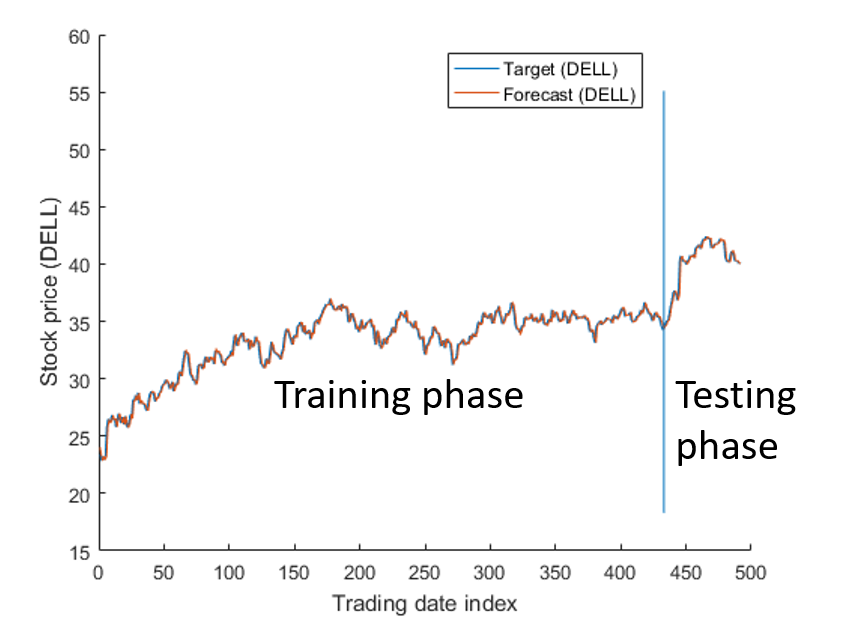
本次範例的結果將與其他論文所提的方法做比較，像是HiMMI[29]、ANN-GA-HMM-Interpolation[29]、ANN-GA-HMM-WA[29]、ARIMA[30]、Bayesian ANN[30]。所以我們將使用模型的第一組輸出的實數及複數部分和第二輸出的實數部份與其他論文做比較，結果如表XVI所示。模型的機器學習曲線，如Fig. 8所示;目標與模型輸出的結果，如Fig. 9所示;誤差圖如Fig. 10所示。



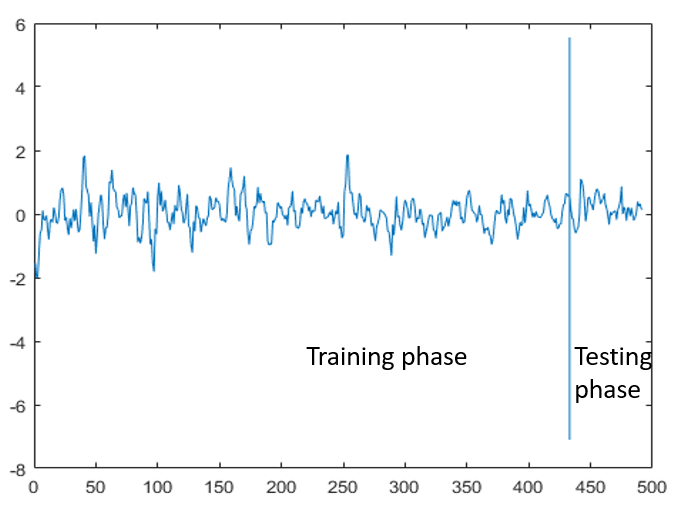
1. 範例三學習曲線。藉由此圖可看出RMSE的變化。x軸為機器學習迭帶，y軸為RMSE。

十次效能表(實驗三)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Performance(MAPE) | | | |
| Trials | APPLE | IBM | DELL | Microsoft |
| 1 | 1.8909 | 0.8156 | 0.6173 | 0.7007 |
| 2 | 2.3692 | 0.8286 | 0.5957 | 0.7111 |
| 3 | **1.8453** | **0.8051** | **0.6187** | **0.8591** |
| 4 | 2.0273 | 0.8172 | 0.6184 | 0.7048 |
| 5 | 2.0476 | 0.8183 | 0.6100 | 0.7099 |
| 6 | 2.2186 | 0.8005 | 0.6270 | 0.7399 |
| 7 | 2.0239 | 0.8313 | 0.6219 | 0.7128 |
| 8 | 2.2186 | 0.8005 | 0.6270 | 0.7399 |
| 9 | 2.0239 | 0.8313 | 0.6219 | 0.7128 |
| 10 | 1.8636 | 0.8159 | 0.6159 | 0.7045 |

(a)  (b)  (c)

1. DJI的實際數值和模型輸出(a) IBM收盤價 (b) APPLE收盤價 (c) DELL收盤價。x軸為交易日，y軸為股票價格，可看出訓練和測試階段皆有不錯的效果。



1. 範例三誤差圖。可從圖中看出每個交易日，模型預測與實際數據的差異。

預測效能比較表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **MAPE** | | | | | | |
| **Method** |  | **APPLE** |  | **IBM** |  | **DELL** |
| HiMMI[29] |  | 2.8373 |  | 1.2186 |  | 1.0117 |
| ANN-GA-HMM-Interpolation[29] |  | 2.1649 |  | 1.0555 |  | 0.8446 |
| ANN-GA-HMM-WA[29] |  | 1.9247 |  | 0.8487 |  | 0.6992 |
| Bayesian ANN[30] |  | 1.9688 |  | 0.7441 |  | - |
| ARIMA[30] |  | 1.8009 |  | 0.9723 |  | 0.6604 |
| SCNFS(proposed) training phase |  | 2.4175 |  | 1.1758 |  | 0.8955 |
| SCNFS(proposed) testing phase |  | 1.8453 |  | 0.8051 |  | 0.6187 |

# Discussion

本研究提出以PSO-RLSE混合演算法優化模型參數，並且在訓練資料進入模型前，事先進行特徵選取，選出對預測目標較為有利之特徵資料進行時間序列之預測。根據3個實驗的測試後，顯示本研究的研究方法對於時間序列的預測擁有較佳的預測性能。

在多目標特徵挑選部分，引用夏農資訊熵[15]的理論。計算出資料特徵對於各個目標資料所能提供的資訊量，並考慮與已被選取的特徵的冗餘資訊量，選取最大增益量，代表最後該特徵被選取後能夠提供給目標資訊量，此外使用者可透過第二次挑選，選擇要進入模型的特徵數量，以提升整體效率，減少模型對於太多目標所產生的預測負擔。

結構學習的過程，透過資料密度量的概念，選擇出較重要的神經元並建構，使用者可透過上下界設定，從挑選後的神經元中選擇個數，此方法減少了大量的神經元數量，同時也降低了運算所需的時間，且可針對不同的輸入資料形成不同的模型大小，提升模型適應性能力。透過模型參數表格可發現，神經元從近百位數下降至個位數，很顯著的控制模型大小。

球式複數模糊類神經系統結合球式複數模糊集合以及T-S模糊系統建立。T-S模糊系統能處理較為模糊的資訊，以一種非線性的方式描述輸入資料的強度，並且以線性規則代表模型輸出，使其系統可以較為人類所理解。球式複數模糊集合，提供了最少四組複數型態歸屬程度，比起傳統模糊集合的單目標預測，可以使模型同時預測八個目標，此外亦可將複數型態值解構，同時預測更多目標，有著較高的延展性及資料豐富性。從實驗中可發現，同時針對四個目標的預測，模型誤差仍然比其他文獻好。

參數學習部分，利用PSO演算法結合RLSE分別對模型的兩部分參數優化，PSO具有三種特性，第一，可隨著群體智慧自動調整步伐;第二，更新速度時具有隨機參數參與其中，可增進粒子活躍性;第三，會跟著一定的正確方向活動，具備著篤定性。但有著搜尋維度較低的缺點，故使用分治法，配合RLSE進行參數優化。RLSE利用參考前次計算結果，尋找線性函數，使資料點與該函數的平方誤差達到最小值，藉由此方法不斷的遞迴，使得模型輸出近似目標，最佳化參數。

透過三個實驗可以發現，本論文提出的模型SCFNS確實有多目標預測的能力，並可以發現透過一次對四個目標做時間序列的預測，各個目標的效果不亞於其他論文所提出的方法，甚至更好。證明了，不同資料都能在此模型中，被有效的預測。也代表PSO-RLSE的混合方法，有著一定的水準。機器學習的部分可能會受限於PSO本身的特性，收斂快速、較容易掉入區域最佳解，從Fig. 2中不難發現，PSO演算法在前幾個回合就已經接近最後一回合的RMSE，故在資料量龐大的情況下，所需搜尋維度增多，PSO效能就會較差，可能會限制住模型的整體發揮。

# Concluding Remarks

## Conclusions

本研究提出POS-RLSE複合演算法，結合PSO演算法以及RLSE，用於優化球式複數型態模糊類神經模型系統的參數集合。而模型採用球式複數模糊集合、T-S系統以及類神經網路概念。資料前處理以特徵選取，減少資料對模型的過多負擔。

1. 多目標特徵挑選，降低模型負擔

本研究根據夏農資訊熵[15]的觀念，開創一種對於多目標特徵選取的方式。讓輸入資料在進入模型前，針對該資料所產生的候選特徵進行篩選。避免冗餘的輸入資料進入模型，耗費運算效能。

2. 球式複數模糊集合，使模型具多個複數型態輸出

本研究採用球式複數模糊集，其複數型態的歸屬程度，讓模型可以有複數值的輸出。有別於一般模糊集合，提升資訊量的豐富度，讓模型能夠有同時進行多目標預測的能力。三個實驗可證明此方式的貢獻。

3. PSO-RLSE複合演算法，降低搜尋維度

模型系統的參數決定預測結果的優劣。透過分治法概念，透過不同機器學習演算法學習不同區域參數，將問題最小化，分而治之，降低演算法搜尋維度，並增進模型整體效能。

## Future Research

PSO演算法仍然有著維度過小，收斂速度快的缺點，在未來，可以透過不同的機器學習演算法結合SCFNS，像是隨機優化演算法 (Random Optimization, RO)[25]、合作型粒子群演算法 (Cooperative Particle Swarm Optimization, CPSO)[26]、螞蟻演算法 (Ant Colony Optimization, ACO)[]等，或許能克服目前PSO所遇到的問題，做出更精準的預測。近年來深度學習的發展快速，六層式神經網路相對其他文獻來說算是較小的架構，未來可擴充更多的類神經網路隱藏層，透過更複雜的運算，嘗試尋找出更優異的預測效果。

##### Acknowledgment

This work was supported by the research projects MOST 105–2221–E–008–091 and MOST 104–2221–E–008–116, Ministry of Science & Technology, Taiwan.

##### References

1. A. J. Patton, “A review of copula models for economic time series,” Journal of Multivariate Analysis, vol. 110, pp. 4–18, Sep. 2012.
2. S. S. Jones, R. S. Evans, T. L. Allen, A. Thomas, P. J. Haug, S. J. Welch and G. L. Snow, “A multivariate time series approach to modeling and forecasting demand in the emergency department,” Journal of Biomedical Informatics, vol. 42, no. 1, pp. 123–139, Feb. 2009.
3. P. Aboagye–Sarfo, Q. Mai, F. M. Sanfilippo, D. B. Preen, L. M. Stewart and D. M. Fatovich, “A comparison of multivariate and univariate time series approaches to modeling and forecasting emergency department demand in Western Australia,” Journal of Biomedical Informatics, vol. 57, pp. 62–73, Oct. 2015.
4. S.–T. Li and T.–C. Cheng, “Deterministic fuzzy time series model for forecasting enrollments,” Computers and Mathematics with Application, vol. 53, no. 12, pp. 1904–1920, Jun. 2007.
5. C. Li and J.–W. Hu, “A new ARIMA–based neuro–fuzzy approach and swarm intelligence for time series forecasting,” Engineering Applications of Artificial Intelligence, vol. 25, no. 2, pp. 295–308, Mar. 2012.
6. C. Li and T.–W. Chiang, “Complex neurofuzzy ARIMA forecasting—a new approach using complex fuzzy sets,” IEEE Transtraction on fuzzy systems. vol. 21, no. 3, pp.567–584, Jun. 2013.
7. L. J. Herrera, H. Pomares, I. Rojas, A. Guillen, J. Gonzalez, M. Awad and A. Herrera, “Multigrid–based fuzzy systems for time series prediction:CATS competition,” Neurocomputing, vol. 70, no. 13–15, pp. 2410–2425, Aug. 2007.
8. I. Sugiarto and S. Natarajan, “Parameter estimation using least square method for MIMO Takagi–Sugeno neuro–fuzzy in time series forecasting,” Jurnal Teknik Elektro, vol. 7, no. 2, pp. 82–87, Sep. 2007.
9. M. Z.–Kermani and M. Teshnehlab, “Using adaptive neuro–fuzzy inference system for hydrological time series prediction.” Applied Soft Computing, vol. 8, no. 2, pp. 928–936, Mar. 2008.
10. H. J. Rong,N. Sundararajan,G. B. Huang and P. Saratchandran, “Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction,” Fuzzy Set Syst, vol.157, no. 9, pp. 1260–1275, May 2006.
11. L. A. Zadeh, “Fuzzy sets”, Information and Control , vol. 8, no. 3, pp. 338–353, Jun. 1965.
12. D. Ramot, R. Milo, M. Friedman, and A, Kandel, “Complex fuzzy sets,” IEEE Transactions on Fuzzy Systems, vol. 10, no. 2, pp. 171–186, Apr. 2002.
13. C. Li, T.–W. Chiang. J.–W. Hu, and T. Wu, “Complex neuro–fuzzy intelligent approach to function approximation,” Advanced Computational Intelligence. 2010 3rd International Workshop on Advanced Computational Intelligence, pp. 151–156, 2010.
14. C. Li and T.–W. Chiang, “Complex fuzzy computing to time series prediction—A multi–swarm PSO learning approach,” Lecture Notes in Computer Science, vol. 6592, pp. 242–251, 2011.
15. E. C. Shannon, “A mathematical theory of communication,” The Bell System Technical Journal, vol. 27, pp. 379–423, Jul. 1948.
16. C. Li, “Multi–target feature selection,” research training material for graduate students in the dept. of Information Management, National Central University, Taiwan, unpublished, 2017.
17. T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC–15, no. 1, pp. 116–132, Jan.–Feb. 1985.
18. J. Kennedy and R. Eberhart, “Particle swarm optimization,” Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948, Nov.–Dec. 1995.
19. J. S. R. Jang, C. T. Sun and E. Mizutani, “Neuro–fuzzy and soft computing: A computational approach to learning and machine intelligence,” IEEE Transactions on Automatic Control, vol. 42, no. 10, Oct. 1997.
20. C. Li and T.–W. Chiang, “Complex neuro–fuzzy self–learning approach to function approximation,” Lecture Notes in Computer Science, vol. 5991, pp. 289–299, 2010.
21. J. S. R. Jang, “ANFIS: Adaptive–network–based fuzzy inference system,” IEEE Transactions on System, Man, and Cybernetics, vol. 23, no. 3, pp. 665–685, May/Jun, 1993.
22. D. S. Broomhead and D. Lowe, “Multivariable functional interpolation and adaptive networks,” Complex System, vol. 2, pp. 321–355, 1998.
23. A. J. Smola and B. Scholkopf , “A tutorial on support vector regression,” Static. Comput., vol. 14, no. 3, pp. 199–222, 2004.
24. N. I. Sapankevychand and R. Sankar, “Time series prediction using support vector machines: A survey,” IEEE Computational Intelligence Magazine, vol. 4, no. 2, pp. 24–38, May 2009.
25. Z. B. Zabinsky, “Random search algorithms.” Technical report, University of Washington, Seattle, Apr. 2009.
26. F. van den Bergh and A.P. Engelbrecht, “A cooperative approach to particle swarm optimization,” IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 225–239, Jun. 2004.
27. S. Chiu, “Fuzzy model identification based on cluster estimation,” Journal of Intelligent and Fuzzy Systems, pp. 267–278 Jan. 1994.
28. R.Clausius, “Ueber eine veränderte Form des zweiten Hauptsatzes der mechanischen Wärmetheorie,” Annalen der Physik und Chemie, vol. 93, no. 12, pp. 481–506, 1854.
29. R. Hassan, B. Nath and M. Kirley, “A fusion model of HMM, ANN and GA for stock market forecasting,” ELSEVIER. Expert Systems with Applications, vol. 33, no. 1, pp. 171-180, Jul. 2007.
30. L. Ticknor, “A Bayesian regularized artificial neural network for stock market forecasting,” Expert Systems with Applications, vol. 40, no. 14, Oct. 2013.