

行政院國家科學委員會專題研究計畫 成果報告

類神經網路演算法之改良(II) 研究成果報告(精簡版)

計畫類別：個別型
計畫編號：NSC 99-2221-E-216-040-
執行期間：99年08月01日至100年07月31日
執行單位：中華大學資訊管理學系

計畫主持人：葉怡成

計畫參與人員：碩士班研究生-兼任助理人員：劉佩玲
碩士班研究生-兼任助理人員：匡麗麗

報告附件：出席國際會議研究心得報告及發表論文

處理方式：本計畫可公開查詢

中華民國 100 年 07 月 29 日

行政院國家科學委員會補助專題研究計畫

成果報告

期中進度報告

類神經網路演算法之改良(II)

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC 99-2221-E-216-040

執行期間：99年8月1日至100年7月31日

執行機構及系所：中華大學資訊管理學系

計畫主持人：葉怡成

計畫參與人員：匡麗麗、劉佩玲

成果報告類型(依經費核定清單規定繳交)： 精簡報告 完整報告

本計畫除繳交成果報告外，另須繳交以下出國心得報告：

赴國外出差或研習心得報告

赴大陸地區出差或研習心得報告

出席國際學術會議心得報告

國際合作研究計畫國外研究報告

處理方式：除列管計畫及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年 二年後可公開查詢

中 華 民 國 100 年 7 月 30 日

摘要

本研究提出一個結合倒傳遞網路(Back-Propagation Network, BPN)與 ARIMA 優點的 ARIMA-BPN 神經網路，它是以 BPN 為模型，將 ARIMA 模式的輸入，包括前時 p 個時刻的數列值與前 q 個時刻的數列殘差值做為輸入值，組成非線性函數，以建立更準確的時間數列預測模型。因為數列殘差值在 BPN 的訓練過程中會因網路連結權值的調整而改變，因此必須修改 BPN 的演算法來適應此需求，即藉由不斷更新每次預測所得之殘差值做為網路的輸入值。本研究以六個人為設計的例題，及現實世界的各國股市時間數列預測來比較 ARIMA、BPN 和 ARIMA-BPN 三者的效能。研究結果顯示，ARIMA-BPN 神經網路演算法在部份例題比 ARIMA 與 BPN 方法更準確。

關鍵字：時間數列、ARIMA、倒傳遞神經網路。

Abstract

In this paper we proposed an ARIMA-BPN algorithm combining advantages of Back-propagation networks (BPN) and ARIMA. The algorithm is based on BPN and its inputs are the same as ARIMA. It can generate a non-linear function to create an accurate model to predict time series. The BPN algorithm must be modified because the residuals would be changed when the weights were changed during continuously training BPN. Therefore, the continuously updated residuals are used as the inputs of ARIMA-BPN. This study examined 6 artificial designed cases and stock index prediction in real world to evaluate the abilities of the ARIMA, BPN, and ARIMA-BPN. The results showed that ARIMA-BPN is the most accurate methods in the three methods.

Key words: time series, ARIMA, Back-Propagation Neural Network.

壹、前言

時間序列預測是一個具有廣泛應用的研究領域。ARIMA 模式，又稱 Box-Jenkins 模式，是一種用來處理複雜的時間序列的一種重要的建模方法(Pankratz, 1983)，其優點在於能精確掌握複雜的時間序列。使用 ARIMA 模式所處理的時間數列必須是穩態的時間序列。所謂穩態是指其統計參數如平均值與標準差不隨時間而變，因此有傾向或季節性的數列便不符合穩態的條件，但可先將數列去除傾向性與季節性後，得到穩態數列再加以處理。ARIMA 在建立時間數列預測模型時，以前 p 個時刻的數列值與前 q 個時刻的數列殘差值做為輸入值，組成線性函數：

$$y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} \quad (1)$$

其中 $y_t = t$ 時刻數列值； $\varepsilon_t = t$ 時刻殘差值； $\phi_0 =$ 常數項； $\phi_{t-p} =$ 第 $t-p$ 時刻數列值的係數； $\theta_{t-p} =$ 第 $t-p$ 時刻殘差值的係數。

在(1) 式中，數列值構成的部份稱為自我迴歸模式(Autoregressive Models, AR)，而殘差值構成的部份稱為移動平均模式(Moving-Average Models, MA)，故統稱自我迴歸移動平均模式(Autoregressiver Moving-Average Models, ARIMA)。

近年來，類神經網路(Artificial Neural Networks, ANN)逐漸成為時間序列預測的另一個重要的建模方法。一個類神經網路是由許多個人工神經元與其連結所組成，並且可以組成各種網路模式(network model)。其中以倒傳遞網路(Back-Propagation Network, BPN)應用最普遍。BPN 模式其基本原理是利用最陡坡將法(the gradient steepest descent method)的觀念，將誤差函數予以最小化(Law, 1999; Law, 2000)。BPN 是一種無回饋連結的前向式網路，其特色是加入隱藏層並改用平滑可微分的非線性轉換函數，以導出網路加權值。基本架構包含了輸入層、隱藏層以及輸出層。隱藏層數目依問題的複雜程度而定。根據經驗，一般問題可取一層隱藏層，較複雜問題則可取二層隱藏層。此外，隱藏層內的神經元數目越多，則可使誤差減小，但也可能發生過度配適(overfitting)，而影響實際的預測能力。BPN 在建立時間數列預測模型時，以前 p 個時刻的數列值做為輸入值，組成非線性函數：

$$y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-p}) \quad (2)$$

比較 ARIMA 與 BPN 兩者，前者多考慮了數列殘差值做為輸入值，但只能產生線性函數，後者則可產生非線性函數，但未考慮了數列殘差值做為輸入值。因此近年來有不少研究在比較二者的優劣。例如，呂志峰(1999)嘗試使用 ARIMA 和 BPN 兩種方法來預測品質成本變化的情形，目的是希望預測出其趨勢，並利用此趨勢，提出品質管理上的一些對策。結果顯示 BPN 的預測誤差低於 ARIMA。李振民(2000)、湯健文(2003)比較 BPN 和 ARIMA 二種方法之準確性。透過數十多個個案研究，結果顯示 BPN 可建立較精確的時間序列預測模型。

過去也有很多學者研究改良 BPN，以更適用在時間序列預測領域。例如，蔡裕春(2001)以 ARIMA 分析技術之有母數分析方法輔助 BPN 信賴區間之建構，彌補 BPN 僅能提供預測之點估計值，無法提供區間估計的缺陷。並以台灣地區營造工程物價指數之預測為例，進行驗證。結果顯示其 90% 及 95% 預測信賴區間包含了所有實際資料，成功地為 BPN 建構了預測值之信賴區間。

此外，也有學者企圖結合 BPN 與 ARIMA 的優點形成複合模式。這些研究大致可分成六類(參考圖 1)：

第 1 類：單模型前處理法

以 ARIMA 技術對數列作前處理，去除傾向性與季節性，產生的平穩化數列交由 ANN 建立預測模型。例如，Jain and Kumar (2007)即利用 ARIMA 去除水文資料的傾向性與季節性，再以 BPN 建模。此類方法的整合程度最低，只是一種很鬆散的整合。

第 2 類：單模型串列整合法 — ANN-AIMA

以 ARIMA 建立模型，此模型的預測值與殘差交由 ANN 建立預測模型。例如，Tseng et al. (2001)以此法處理臺灣的機械產業產值與軟性飲料產值的時間數列，研究結果發現，此法優於單獨使用 ARIMA 與 ANN。

第 3 類：單模型串列整合法 — ARIMA-ANN

以 ANN 建立模型，此模型的預測值與殘差交由 ARIMA 建立預測模型。此法為第 2 類方法的反向方法，即先用 ANN 再用 ARIMA。

第 4 類：雙模型平行加權法

以 ARIMA 與 ANN 各自建立一個模型，再以加權平均法等將二個模型的預測值整合成最終預測值。加權平均的權值一般可用模型的殘差平方和的倒數。例如，張百棧(2004)建立多元迴歸分析、倒傳遞類神經網路、以及自我迴歸移動平均整合模式，再將這三種預測模型所得到之結果透過一動態權重值加以結合。並以台灣地區股價指數之預測為例，進行驗證。研究結果發現，結合模型的預測誤差比其他三種預測模型有明顯改善。Inoue et al. (2001)則用此法整合 ANN 與最近鄰居法(kNN)於時間數列預測，研究結果發現，整合屬於 Global Model 的 ANN 與 Local Model 的 kNN 可以獲得更準確的預測。

第 5 類：雙模型串列疊加法 — ANN-ARIMA

以 ANN 建立第一個模型，此模型的殘差交由 ARIMA 建立第二個模型，最後以二個模型的預測值疊加成最終預測值。例如，王成財(2002)結合 BPN 與 ARIMA 發展「複合模式」。此模式先以 BPN 做預測，求出預測值與實際值的殘差，再利用 ARIMA 針對殘差值加以預測，最後將二者所得之預測值疊加後產生最終預測值。研究結果顯示，此複合模式優於 BPN 模式，而 ARIMA 最差。

第 6 類：雙模型串列疊加法 — ARIMA-ANN

以 ARIMA 建立第一個模型，此模型的殘差交由 ANN 建立第二個模型，最後以二個模型的預測值疊加成最終預測值。例如，Zhang (2003)用此法處理三個時間數列的經典例題：太陽黑子(sunspot)問題(1700~1987，288 筆年資料)、加拿大山貓(lynx)問題(1821~1934，114 筆年資料)、英磅美金匯率問題(1980~1993，731 筆週資料)。研究結果顯示，此法優於單獨使用 ARIMA 與 ANN。此外，Bo et al. (2007)擴充 Zhang 的方法，整合了 ARCH 族的方法(包括 ARCH, GARCH, EGARCH 等)與 ANN。而 He et al. (2006)也擴充 Zhang 的方法，整合了 ARIMA 與支持向量機(support vector machine)。這些研究都顯示複合模式優於單一模式。

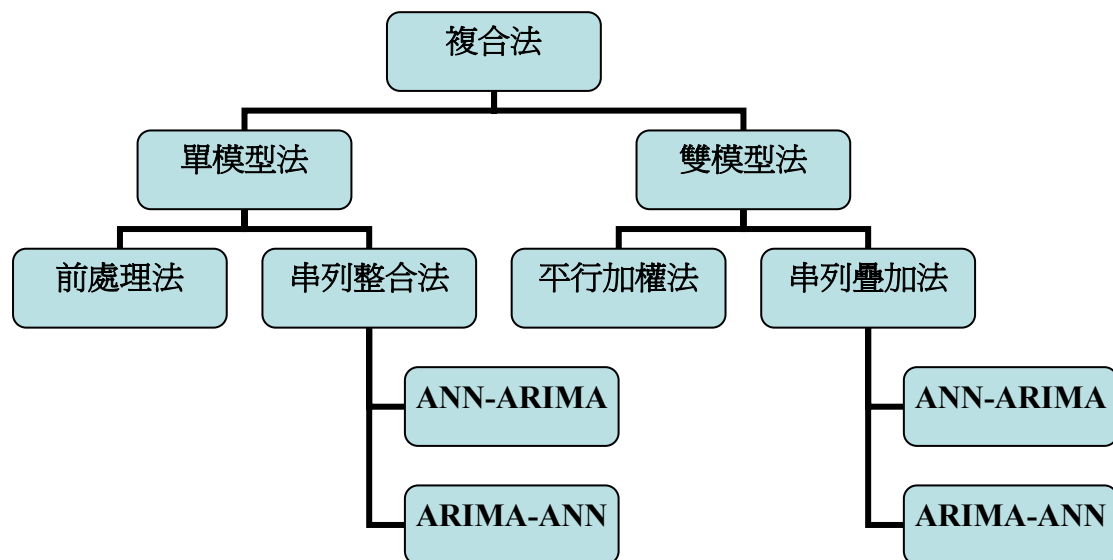


圖 1 ARIMA 與 ANN 的複合法之分類

由上述文獻探討可知，雖然已有學者提出結合 ARIMA 與 BPN 的複合模式，然而都屬於鬆散式結合，並沒有發展出真正結合 ARIMA 與 BPN 原理的複合模式。此外，上述的串列整合法、平行加權法、串列疊加法等複合模式的共同缺點是必需建立兩個以上的模型，在使用上較為複雜，也容易出錯。且因為模型過於複雜，其結論「複合模式優於單一模式」有過度配適(overfitting)的可能。有鑑於此，本文提出一個結合 ARIMA 之整合數列值與殘差值於同一模型，以及 BPN 的建構非線性模型能力優點的真正複合模式 — ARIMA-BPN 神經網路，以改善預測效果。ARIMA-BPN 以 BPN 為模型，將 ARIMA 模式的輸入，包括前時 p 個時刻的數列值與前 q 個時刻的數列殘差值，都做為 BPN 的輸入值，組成非線性函數 $y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-p}, \varepsilon_{t-1}, \varepsilon_{t-2}, \dots, \varepsilon_{t-p})$ ，以建立更準確的時間數列預測模型。因為數列殘差值在 BPN 的訓練過程中會因網路連結權值的調整而改變，因此必須修改 BPN 的演算法來適應此需求，即藉由不斷更新每次預測所得之殘差值做為網路的輸入值。

ARIMA-BPN 神經網路有別於前述六種複合法，它真正將 ARIMA 的概念融入 ANN 中，產生一種新的 BPN 演算法。而前述的單模型法只是透過模型間的資料傳遞來整合；而雙模型法只是整合 ARIMA 與 ANN 的預測結果。因此，無論單模型法、雙模型法都屬於鬆散式結合，而本文提出的 ARIMA-BPN 神經網路則屬於緊密式結合。

本文第二節將介紹 ARIMA-BPN 神經網路演算法。而第三節與第四節分別以六個數值例題與一個現實世界的例題進行實證。第五節為結論。

貳、ARIMA-BPN 神經網路

本研究所提出的 ARIMA-BPN 演算法是以 BPN 為基礎，並加入了殘差向量來改良預測效果的演算法。BPN 的理論已有大量的文獻(葉怡成，2005；葉怡成 2006)，在此不再贅述。

傳統的 BPN 在處理時間數列時是以前 p 個時刻的數據來預測第 $p+1$ 個數值，也就是輸入層有 p 個。ARIMA-BPN 則另外加入了 q 個殘差做為輸入，組成非線性函數 $y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-p}, \varepsilon_{t-1}, \varepsilon_{t-2}, \dots, \varepsilon_{t-p})$ ，以建立更準確的時間數列預測模型。因為數列殘差值在 BPN 的訓練過程中會因網路連結權值的調整而改變，因此必須修改 BPN 的演算法來適應此需求，即藉由不斷更新每次預測所得之殘差值做為網路的輸入值。圖 2 是 ARIMA-BPN 神經網路架構圖。

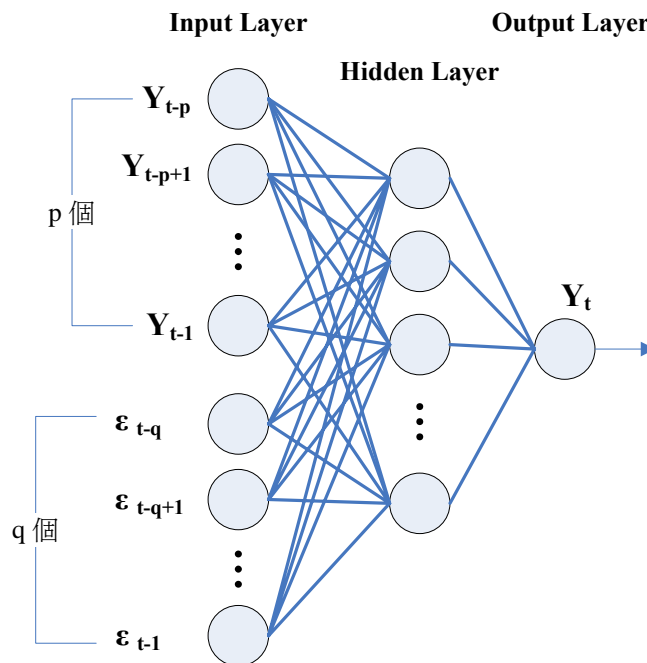


圖 2 ARIMA-BPN 神經網路架構圖

在第一次訓練時，由於還沒有預測值，所以輸入層的 q 個殘差皆為 0。另外輸入層還有 1 至 p 等 p 個時刻的數據作為輸入層。所以演算法用了 $p+q$ 個輸入層，透過隱藏層的運算(此時殘差皆為 0)得到第 $p+1$ 個時刻的預測數據。此時得到預測數據後，再將第 $p+1$ 個時刻的實際數據與預測數據相減，即可得到殘差。這時我們需要更新初始的 q 個殘差。假設有 $1 \sim q$ 個殘差，以第 $q-1$ 個取代第 q 個殘差，第 $q-2$ 個取代第 $q-1$ 個殘差，以此類推直到第 1 個取代第 2 個殘差。最後以本次的殘差取代第 1 個殘差，即再開始下個訓練。

在第二次訓練時，在輸入層的第 1 個殘差為前一時刻的殘差，其餘 $q-1$ 個殘差還是

為 0。第二次循環將預測第 $p+2$ 個時刻的數據，因此還以 2 至 $p+1$ 個時刻的數據做輸入。同樣地，經由隱藏層的運算，會得到新的殘差。更新殘差後，再接著下個訓練，直到訓練完最後一個範例。

在更新殘差後，可以發現若下次欲預測第 n 個時刻，則第 1 個殘差屬於第 $n-1$ 個時刻，而第 q 個殘差屬於第 $n-q$ 個時刻。而隱藏層在計算隱藏層的權重值時會用到第 $n-1$ 到 $n-p$ 個數列值，以及第 $n-1$ 到 $n-q$ 個殘差值，運算後即得到第 n 個時刻的預測值。

以下是 ARIMA-BPN 神經網路演算法的學習過程。

1. 設定網路參數：包括學習速率 η 、慣性因子 α 、AR 的階次 p 、MA 的階次 q 、時間序列長度 n 等。(此網路的輸入變數有 $p+q$ 個，輸出變數只有一個)
2. 以均佈隨機亂數設定加權值矩陣 W 、門限值向量初始值 θ 。
3. 設 $t=p+1$
4. 設定殘差向量初始值

$$\varepsilon_{t-1}, \varepsilon_{t-2}, \dots, \varepsilon_{t-q} = 0 \quad (3)$$

5. 決定輸入向量

$$\text{輸入向量 } X_i = Y_{t-i} \quad i = 1, 2, \dots, p \quad (4)$$

$$\text{輸入向量 } X_{p+i} = \varepsilon_{t-i} \quad i = 1, 2, \dots, q \quad (5)$$

$$\text{目標輸出 } \textit{desire_y} = Y_t \quad (6)$$

6. 計算推論輸出值 y

- (a) 計算隱藏層輸出向量 $\{h_1, h_2, h_3, \dots\}$

$$\textit{net}_k = \sum_k W_{ik} x_i - \theta_k \quad (7)$$

$$h_k = \frac{1}{1 + \exp(-\textit{net}_k)} \quad (8)$$

- (b) 計算推論輸出向量 y

$$\textit{net} = \sum_k W_k h_k - \theta \quad (9)$$

$$y = \frac{1}{1 + \exp(-\textit{net})} \quad (10)$$

7. 計算差距量

- (a) 計算輸出層差距量 δ

$$\delta = (\textit{desire_y} - y) \cdot y \cdot (1 - y) \quad (11)$$

- (b) 計算隱藏層差距量 δ_k

$$\delta_k = (\delta \cdot W_k) \cdot h_k \cdot (1 - h_k) \quad (12)$$

8. 計算加權值矩陣修正量，及門限值向量修正量

- (a) 計算輸出層加權值矩陣修正量，及門限值向量修正量

$$\Delta W_k(n) = \eta \delta h_k + \alpha \cdot \Delta W_k(n-1) \quad (13)$$

$$\Delta \theta(n) = -\eta \delta + \alpha \cdot \Delta \theta(n-1) \quad (14)$$

(b) 計算隱藏層加權值矩陣修正量，及門限值向量修正量

$$\Delta W_{ik}(n) = \eta \delta_k x_i + \alpha \cdot \Delta W_{ik}(n-1) \quad (15)$$

$$\Delta \theta_k(n) = -\eta \delta_k + \alpha \cdot \Delta \theta_k(n-1) \quad (16)$$

9. 更新加權值矩陣，及門限值向量

(a) 更新輸出層加權值矩陣，及門限值向量

$$W_k = W_k + \Delta W_k \quad (17)$$

$$\theta = \theta + \Delta \theta \quad (18)$$

(b) 更新隱藏層加權值矩陣，及門限值向量

$$W_{ik} = W_{ik} + \Delta W_{ik} \quad (19)$$

$$\theta_k = \theta_k + \Delta \theta_k \quad (20)$$

(c) 計算殘差值

$$\varepsilon_i = \text{desire}_i - y \quad (21)$$

10. 設 $t=t+1$

11. 檢查是否已到最後一個訓練範例，完成一個學習循環：如果 $t \leq n$ 回到步驟 5，否則到步驟 12。

12. 重覆步驟 3 至步驟 11，直到預設的學習循環數目。

以上介紹中，值得注意的是由於開始時尚未做預測，因此殘差向量設為 0。在預測第 $p+1$ 個時刻時，其 $p+q$ 個輸入變數中，前 p 個輸入為前 p 個數列值，後 q 個輸入為前 p 個時刻的殘差值。而學習過程中在更新殘差向量時，以第 $q-1$ 個取代第 q 個殘差，第 $q-2$ 個取代第 $q-1$ 個殘差，以此類推直到第 1 個取代第 2 個殘差，最後以本次的殘差取代第 1 個殘差。另外每次在開始一個新的學習循環時，要記得將殘差向量設為 0，因為循環開始時尚未有預測數據，殘差是未知的，故設為 0。

舉例來說，有 5 筆數據，以前 p 個時刻及前 q 個時刻的殘差值為神經網路之輸入，進而預測下個時刻之數據。本例子的 p 與 q 為 2。圖 3 為此範例之流程圖。在步驟一，欲預測值為第 3 個時刻，因此我們以第 1 與第 2 個時刻，也就是前 2 個數列值為輸入。另外，ARIMA-BPN 神經網路還需要前 q 個時刻之殘差值做為輸入，但是由於步驟一之前並未有任何的預測，所以我們將第 1 個與第 2 個時刻之殘差值 q_1 與 q_2 設為 0。輸入完成後 ARIMA-BPN 神經網路即可得到第 3 個時刻的預測值，並以倒傳遞演算法的通用差距法則(general delta rule)，即(13)~(16)式，修正網路的連結權值。

到了步驟二時，欲預測值為第 4 個時刻，因此我們以第 2 與第 3 個時刻，也就是前 2 個數列值為輸入。而在殘差值部分，將第 3 個時刻之實際值減去步驟一的第 3 個時刻的預測值得到第 3 個時刻的殘差值，然後以第 2 個與第 3 個時刻的殘差值為輸入。輸

入完成後 ARIMA-BPN 神經網路即可得到第 4 個時刻的預測值，並以通用差距法則修正網路的連結權值。

同樣地，到了步驟三時，欲預測值為第 5 個時刻，因此我們以第 3 與第 4 個時刻，也就是前 2 個數列值為輸入。而在殘差值部分，將第 4 個時刻之實際值減去步驟一的第 4 個時刻的預測值得到第 4 個時刻的殘差值，以第 3 個與第 4 個時刻的殘差值為輸入。輸入完成後 ARIMA-BPN 神經網路即可得到第 5 個時刻之預測值，並以通用差距法則修正網路的連結權值。

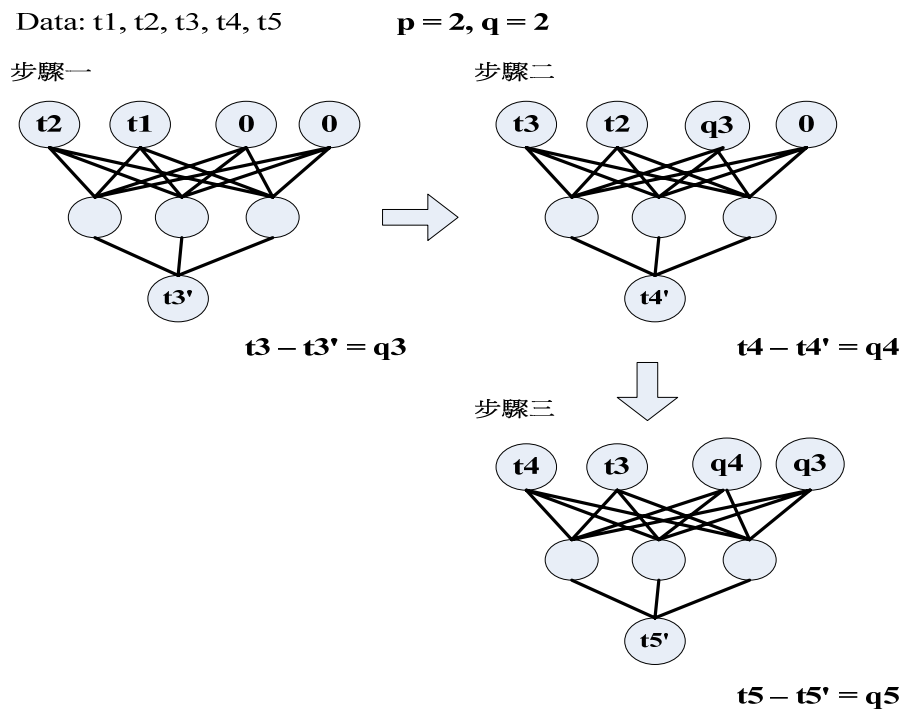


圖 3 ARIMA-BPN 神經網路簡例流程圖

參、數值例題

3.1 數值例題

爲了驗證 ARIMA-BPN 的效能，本節設計了六個以「殘差值」構成的「非線性」時間序列。這樣設計的原因是因為 ARIMA 無法處理「非線性」，而 BPN 無法處理「殘差值」。而 ARIMA-BPN 是在擅長建構非線性模型的 BPN 中加入 ARIMA 的以殘差值爲輸入之概念，因此有可能可以建立比 ARIMA 及 BPN 更準確的預測模型。這六個時間序列公式如下：

例題一

$$y_t = -0.6\varepsilon_{t-1} + 1.3\varepsilon_{t-2} + 2\varepsilon_{t-1}\varepsilon_{t-2} + \varepsilon_t \quad (22)$$

例題二

$$y_t = 2\varepsilon_{t-1}\varepsilon_{t-2} + \varepsilon_t \quad (23)$$

例題三

$$y_t = 1.3\varepsilon_{t-1}^2 - 0.6\varepsilon_{t-2}^2 + \varepsilon_t \quad (24)$$

例題四

$$y_t = -1.8\varepsilon_{t-1} + 0.9\varepsilon_{t-2} + 10\varepsilon_{t-1}\varepsilon_{t-2} + \varepsilon_t \quad (25)$$

例題五

$$y_t = 3\gamma_{t-1}\varepsilon_{t-1} + \gamma_t \quad (26)$$

$$\gamma_t = 0.9\gamma_{t-1} + 0.9\varepsilon_{t-1} + \varepsilon_t \quad (27)$$

例題六

$$y_t = \gamma_{t-1}^2 + \varepsilon_t + \gamma_t \quad (28)$$

$$\gamma_t = 8\varepsilon_{t-1}^2 + \varepsilon_t \quad (29)$$

每一個公式產生一個由 800 個數列值構成的時間序列，其中前 400 個將作為訓練範例，後 400 作為測試範例，以避免過度配適的問題。對所有例題，BPN 一律取前二個時刻的數列值為輸入，而 ARIMA-BPN 則除了取前二個時刻的數列值為輸入外，還取前二個時刻的殘差值為輸入，即取 $p = 2$ ， $q = 2$ 。而 ARIMA 則以典型的平穩化、模型鑑別、參數估計、殘差診斷等四個步驟建立最適模型。

3.2 結果

表 1 與表 2 為實驗結果，表中的結果都以測試範例為準。由表可得下列結論：

1. 誤差均方根的比較

由表 1 可知，除了數列二之外，ARIMA-BPN 的誤差均方根(RMS)都是三個方法中最小的，且在六個例題中，ARIMA-BPN 的誤差都低於 ARIMA。

2. 判定係數 R^2 的比較

由表 2 與圖 4 可知，除了數列二之外，ARIMA-BPN 的判定係數 R^2 都是三個方法中最大的，且在六個例題中，ARIMA-BPN 的判定係數都高於 ARIMA。此結論與觀察 RMS 的結論相似，且更顯著。

3. 散佈圖的比較

由圖 5 至圖 10 可知，ARIMA-BPN 的實際值與預測值的關係遠比 BPN 合理地散佈在 45 度對角線上。

表 1 各方法之 RMS 比較 (黑體字為三個方法中的最佳方法)

方法 例題	ARIMA (測試範例)	BP (測試範例)	ARIMA-BPN (測試範例)
例題 1	0.432	0.49	0.43
例題 2	0.33	0.32	0.32
例題 3	0.315	0.31	0.30
例題 4	1.085	1.101	1.100
例題 5	1.22	1.19	1.14
例題 6	1.32	1.23	1.20

表 2 各方法之判定係數 R^2 比較

例題	ARIMA (測試範例)	BP (測試範例)	ARIMA-BPN (測試範例)
例題 1	0.337	0.224	0.382
例題 2	0.010	0.123	0.105
例題 3	0.003	0.0456	0.0752
例題 4	0.0243	0.0216	0.0289
例題 5	0.222	0.244	0.287
例題 6	0.181	0.224	0.266

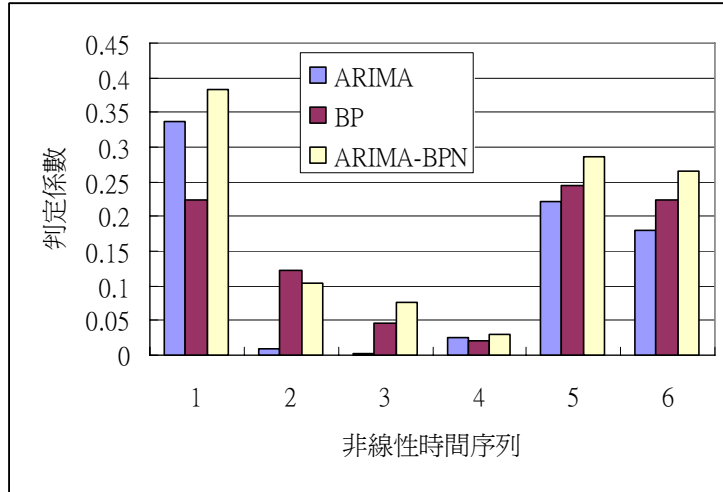


圖 4 各方法之 R^2 比較

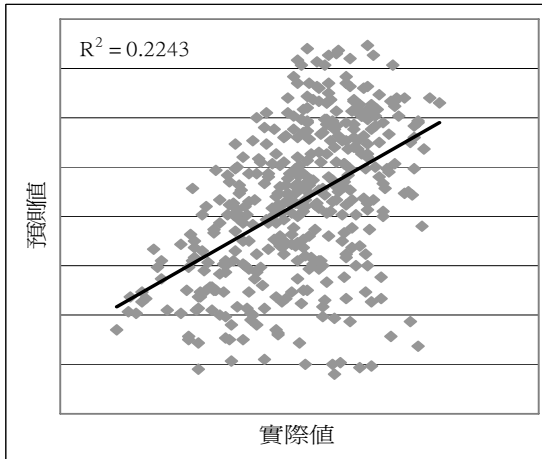


圖 5(a) BPN 測試範例之散佈圖

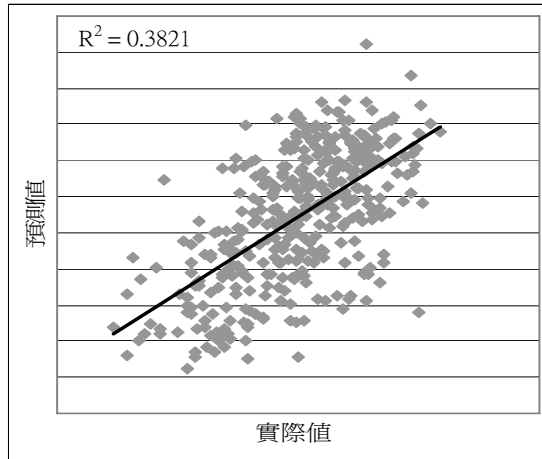


圖 5(b) ARIMA-BPN 測試範例之散佈圖

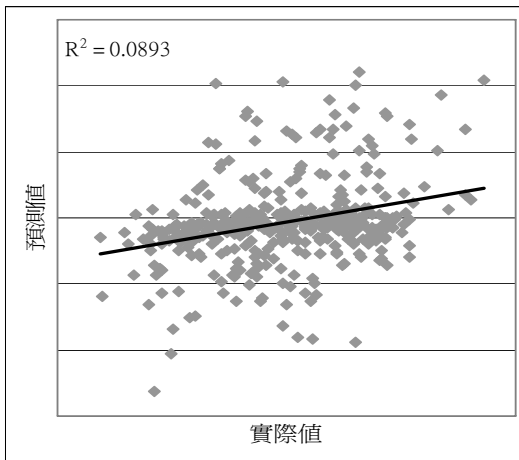


圖 6(a) BPN 測試範例之散佈圖

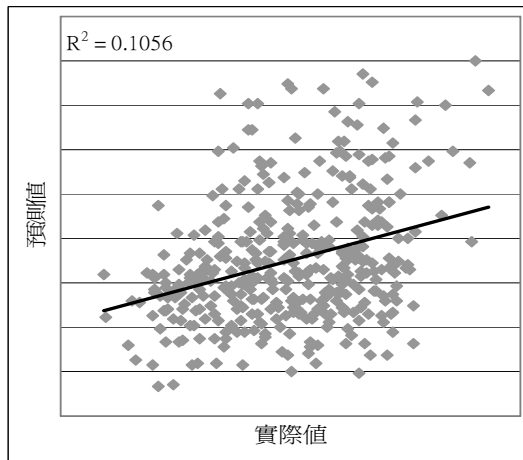


圖 6(b) ARIMA-BPN 測試範例之散佈圖

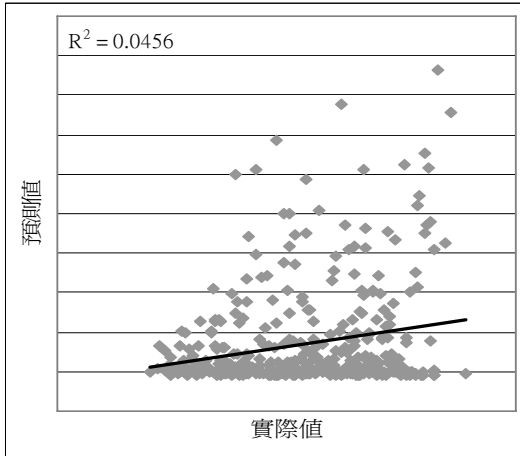


圖 7(a) BPN 測試範例之散佈圖

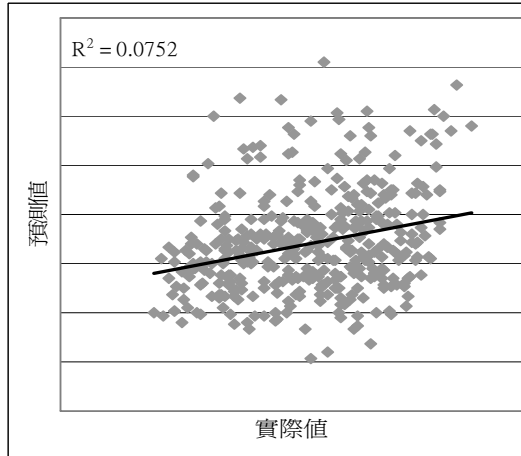


圖 7(b) ARIMA-BPN 測試範例之散佈圖

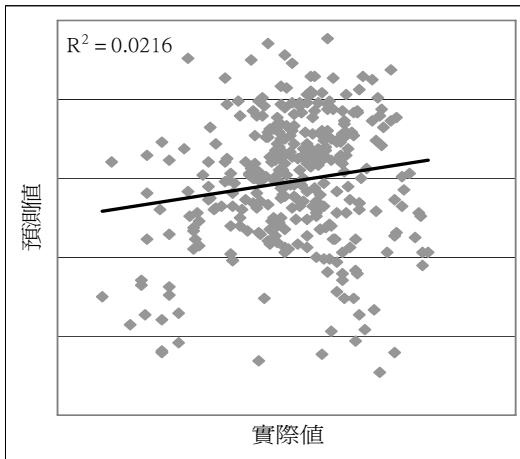


圖 8(a) BPN 測試範例之散佈圖

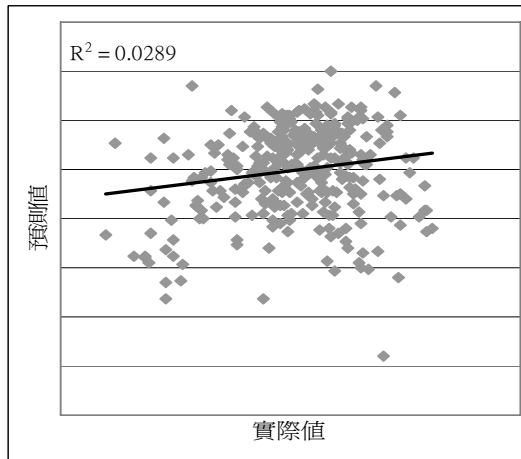


圖 8(b) ARIMA-BPN 測試範例之散佈圖

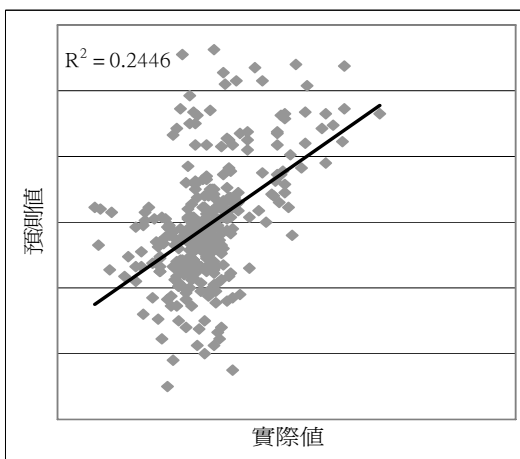


圖 9(a) BPN 測試範例之散佈圖

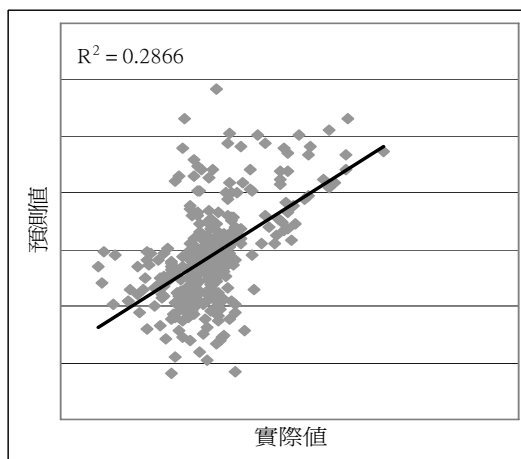


圖 9(b) ARIMA-BPN 測試範例之散佈圖

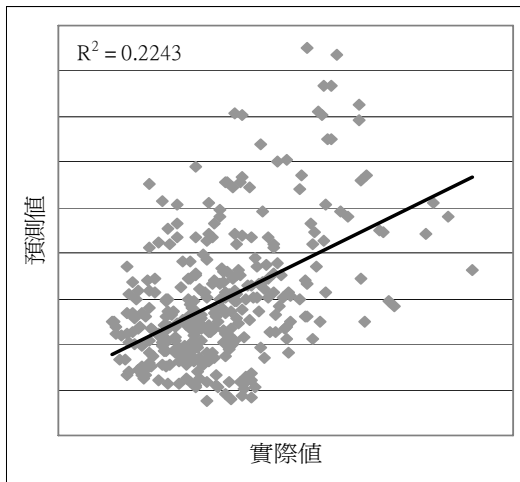


圖 10(a) BPN 測試範例之散佈圖

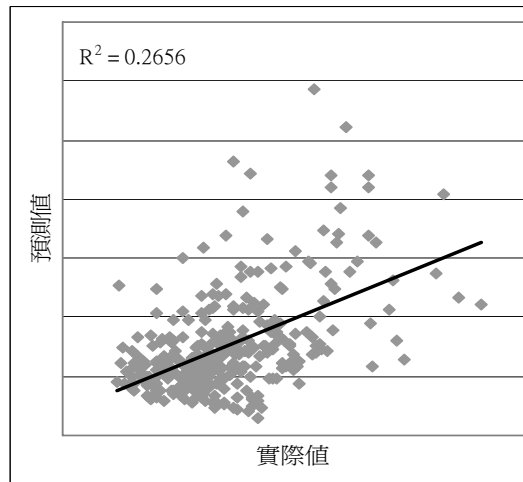


圖 10(b) ARIMA-BPN 測試範例之散佈圖

肆、各國股市時間數列預測

4.1 簡介

本節以包括美國、巴西、英國、香港、印尼、日本、韓國、馬來西亞、墨西哥、台灣等十個股票市場(表 3)，各約 2000 多日的收盤假時間數列，來評估 ARIMA-BP 的建模能力。

表 3 本研究選取的股價指數

國家	股市名稱	股票代號
日本	NIKKEI 225	N225
印尼	Composite Index	JKSE
巴西	IBOVESPA SAO PAULO	BVSP
馬來西亞	KLSE Composite	KLSE
墨西哥	IPC	MXX
韓國	KOSPI Composite Index	KS11
美國	DOW JONES COMPOSITE INDEX	DJA
英國	FTSE 100	FTSE
香港	HANG SENG INDEX	HSI
台灣	TSEC weighted index	TWII

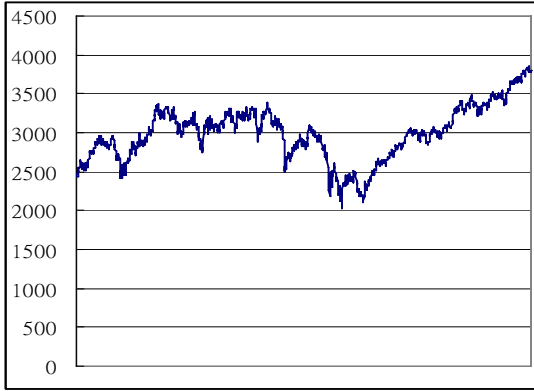


圖 11(a) America 原始數據折線圖



圖 11(b) Brazil 原始數據折線圖

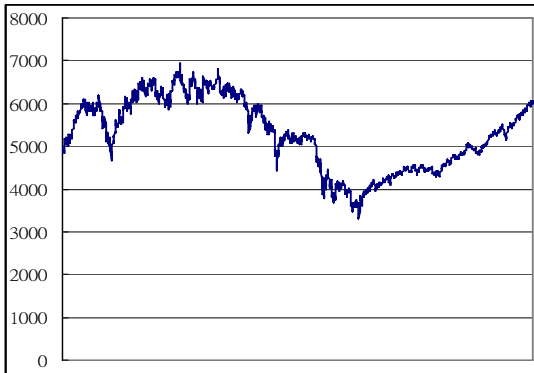


圖 11(c) England 原始數據折線圖

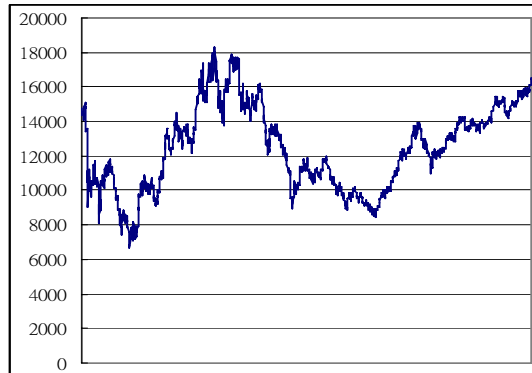


圖 11(d) Hong Kong 原始數據折線圖

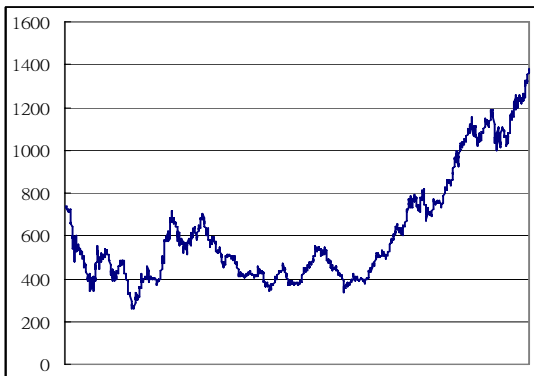


圖 11(e) Indonesia 原始數據折線圖

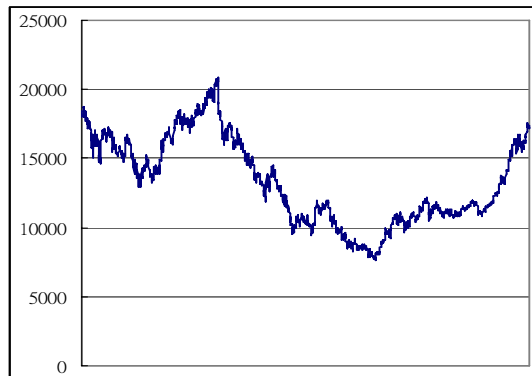


圖 11(f) Japan 原始數據折線圖

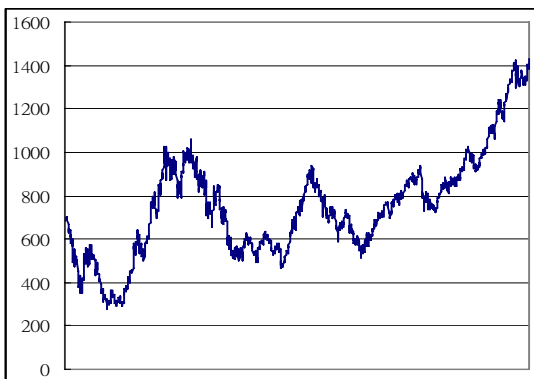


圖 11(g) Korea 原始數據折線圖

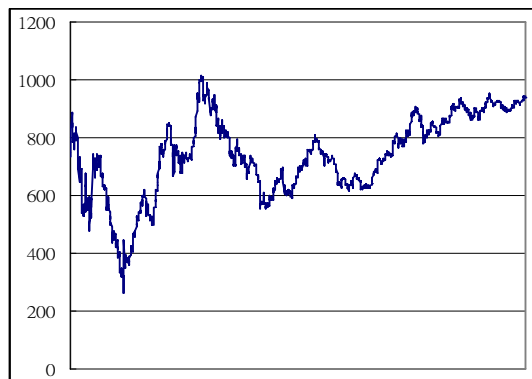


圖 11(h) Malaysia 原始數據折線圖

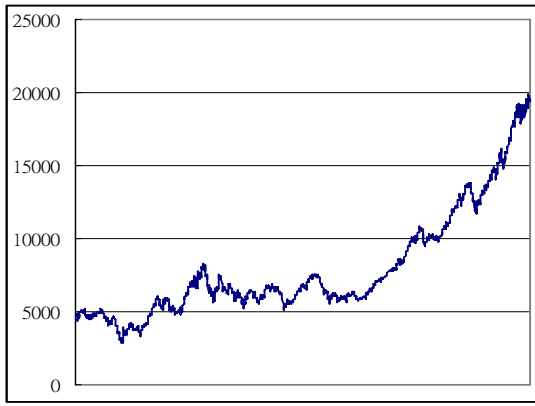


圖 11(i) Mexico 原始數據折線圖

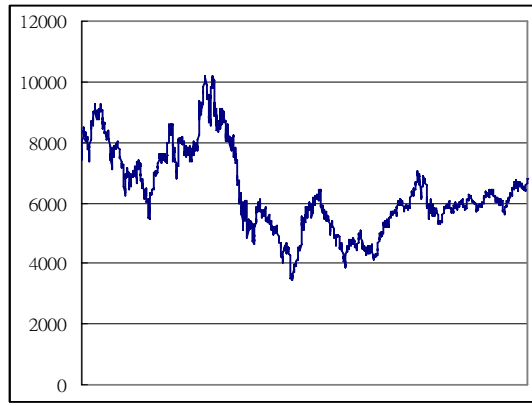


圖 11(j) Taiwan 原始數據折線圖

4.2 ARIMA 分析

在本例題的 ARIMA 分析實驗中，使用了 CPRE 軟體且一般差分設為 1。經過 ARIMA 分析後，可以得到四種模型的 \bar{R}^2 值，選擇 \bar{R}^2 值最高者為最佳模型。但是可以發現有不只一個最大 \bar{R}^2 的狀況，也就是 ARIMA 無法精確建模的情形，如 America 與 Mexico。觀察 RMS 值，發現還是無法判定 America 在何種模型中表現較佳。其他可發現 Indonesia 與 Mexico 為 MA1 模型，其餘全部為 AR2 模型。表 4 為經過 ARIMA 分析後所得的 \bar{R}^2 值與 RMS 值。圖 12 與圖 13 分別為 ACF 圖與 PACF 圖。

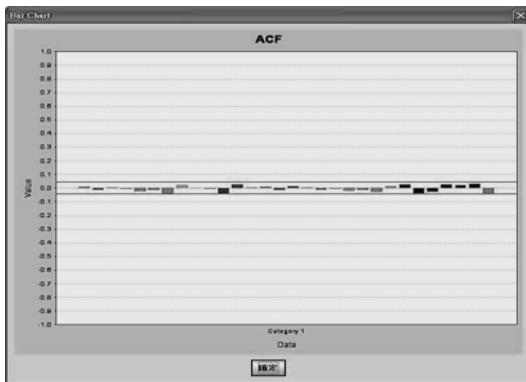


圖 12(a) America 股市 ACF 圖：一次差分

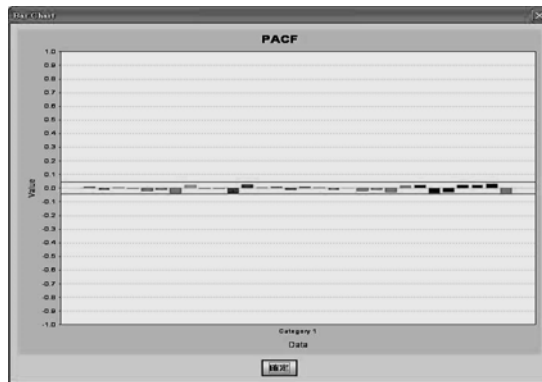


圖 13(a) America 股市 PACF 圖：一次差分

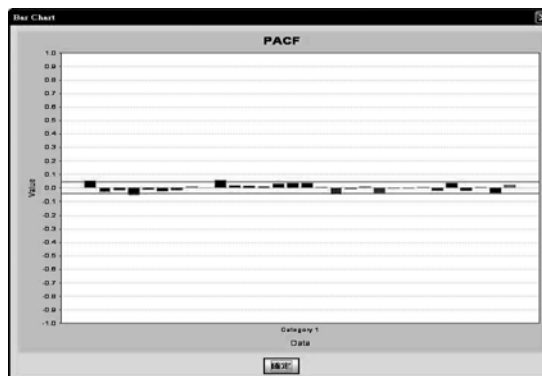
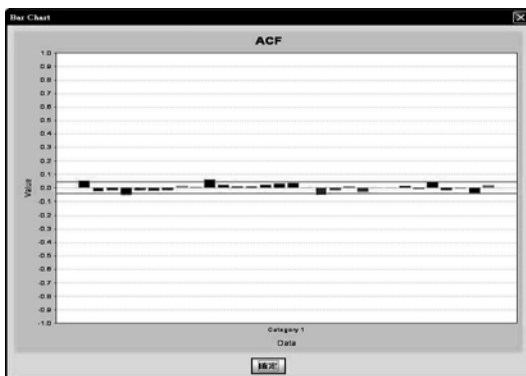


圖 12(b) Brazil 股市 ACF 圖: 一次差分

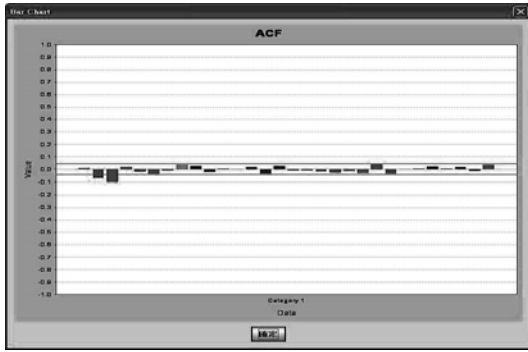


圖 13(b) Brazil 股市 PACF 圖: 一次差分

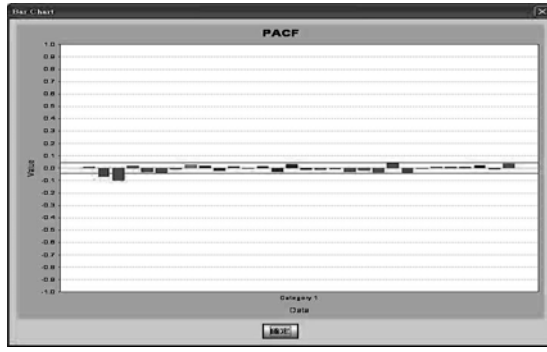


圖 12(c) England 股市 ACF 圖: 一次差分

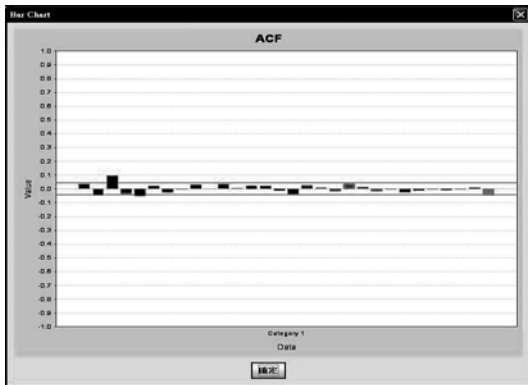


圖 13(c) England 股市 PACF 圖: 一次差分

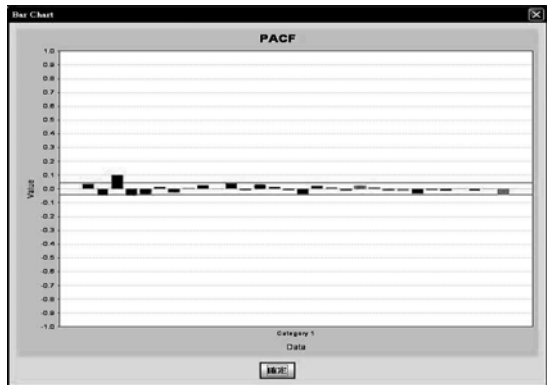


圖 12(d) Hong Kong 股市 ACF 圖: 一次差分

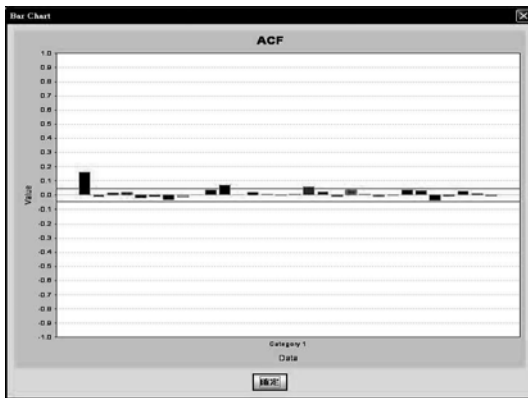


圖 13(d) Hong Kong 股市 PACF 圖: 一次差分

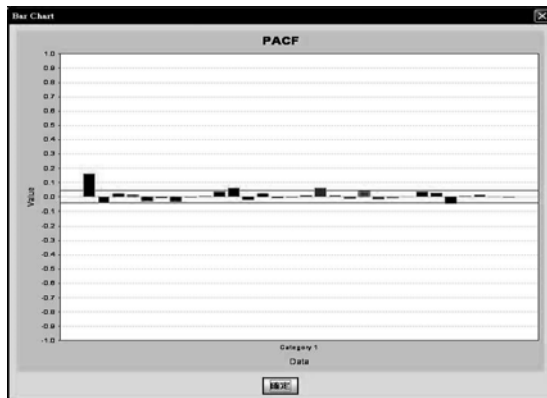


圖 12(e) Indonesia 股市 ACF 圖: 一次差分

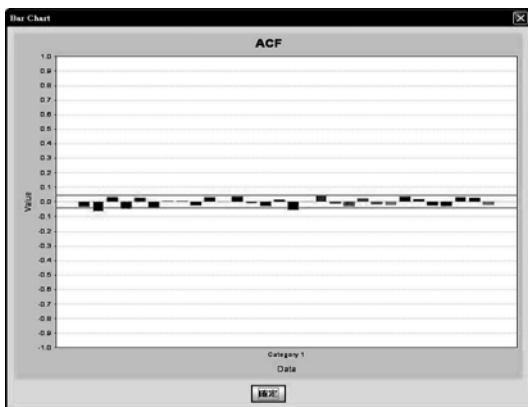


圖 13(e) Indonesia 股市 PACF 圖: 一次差分

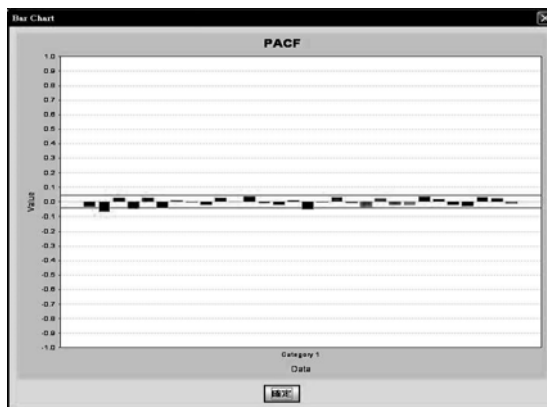


圖 12(f) Japan 股市 ACF 圖: 一次差分

圖 13(f) Japan 股市 PACF 圖: 一次差分

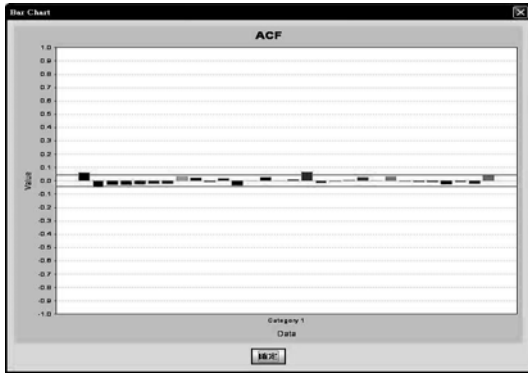


圖 12(g) Korea 股市 ACF 圖: 一次差分

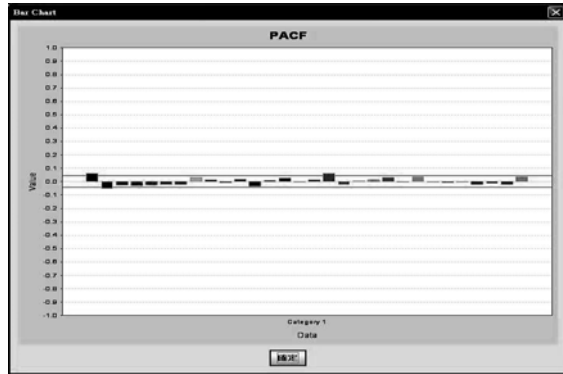


圖 13(g) Korea 股市 PACF 圖: 一次差分

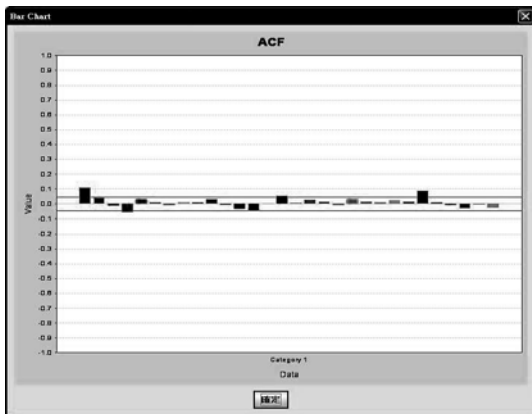


圖 12(h) Malaysia 股市 ACF 圖: 一次差分

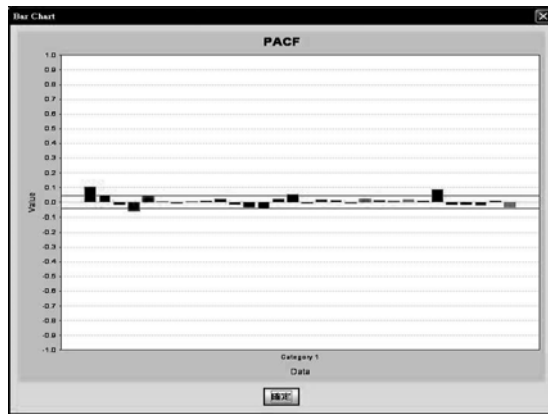


圖 13(h) Malaysia 股市 PACF 圖: 一次差分

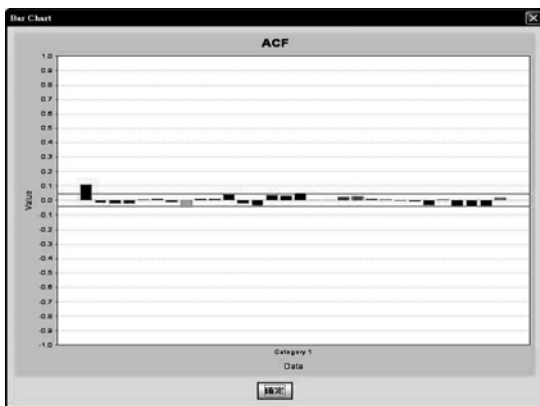


圖 12(i) Mexico 股市 ACF 圖: 一次差分

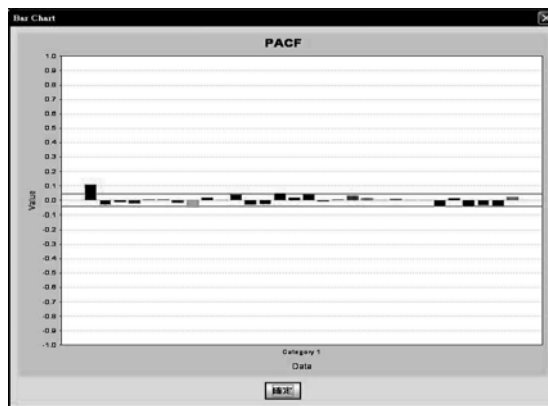


圖 13(i) Mexico 股市 PACF 圖: 一次差分

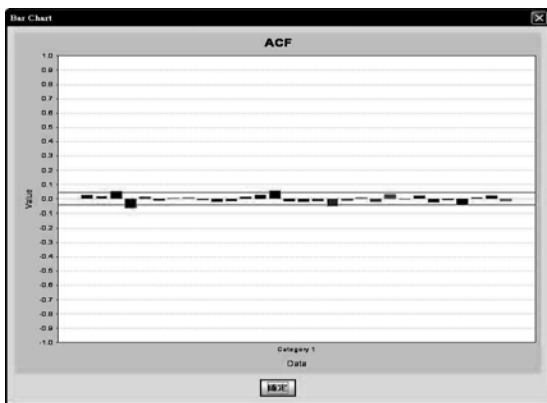


圖 12(j) Taiwan 股市 ACF 圖: 一次差分

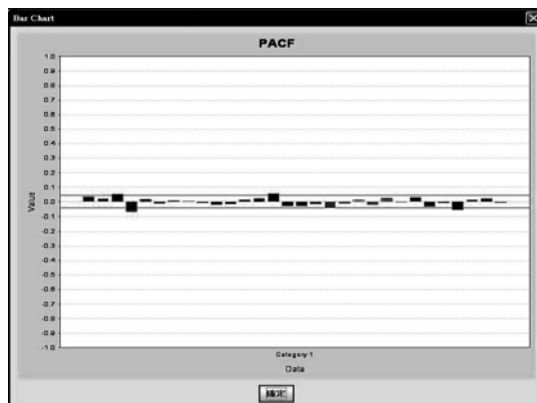


圖 13(j) Taiwan 股市 PACF 圖: 一次差分

表 4 ARIMA 分析各國股市之 \bar{R}^2 值與 RMS 值

		AR1	AR2	MA1	MA2
美國	全部範例判定係數	-0.000	0.000	-0.000	0.000
	全部範例 RMS	30.02	30.01	30.02	30.01
	訓練範例 RMS (1060)	31.95	31.92	31.95	31.98
	測試範例 RMS (1057)	27.821	27.83	27.820	27.87
巴西	全部範例判定係數	0.003	0.004	0.003	-0.000
	全部範例 RMS	322.10	322.04	322.07	322.61
	訓練範例 RMS (1060)	297.06	297.16	297.04	297.87
	測試範例 RMS (1057)	345.14	344.67	345.12	345.24
英國	全部範例判定係數	0.001	0.005	0.001	0.000
	全部範例 RMS	60.79	60.65	60.79	60.80
	訓練範例 RMS (1000)	71.73	71.29	71.72	71.86
	測試範例 RMS (998)	47.130	47.131	47.135	47.30
香港	全部範例判定係數	0.001	0.003	0.001	-0.000
	全部範例 RMS	205.51	205.31	205.50	205.68
	訓練範例 RMS (1060)	265.47	265.15	265.45	265.70
	測試範例 RMS (1057)	118.13	118.06	118.15	118.23
印尼	全部範例判定係數	0.026	0.027	0.028	-0.000
	全部範例 RMS	10.10	10.10	10.09	10.24
	訓練範例 RMS (1060)	10.7164	10.72	10.7162	10.86
	測試範例 RMS (1057)	9.4310	9.4059	9.41	9.401
日本	全部範例判定係數	0.001	0.006	0.001	0.000
	全部範例 RMS	203.03	202.55	203.01	203.10
	訓練範例 RMS (1060)	247.93	247.05	247.87	248.19
	測試範例 RMS (1057)	144.5394	144.56	144.5396	144.59
韓國	全部範例判定係數	0.005	0.007	0.005	0.001
	全部範例 RMS	15.38	15.36	15.37	15.41
	訓練範例 RMS (1060)	17.596	17.575	17.59	17.65
	測試範例 RMS (1057)	12.765	12.75	12.763	12.77

表 4 ARIMA 分析各國股市之 \bar{R}^2 值與 RMS 值(續)

		AR1	AR2	MA1	MA2
韓國	全部範例判定係數	0.005	0.007	0.005	0.001
	全部範例 RMS	15.38	15.36	15.37	15.41
	訓練範例 RMS (1060)	17.596	17.575	17.59	17.65
	測試範例 RMS (1057)	12.765	12.75	12.763	12.77
馬來西亞	全部範例判定係數	0.013	0.050	0.013	0.035
	全部範例 RMS	10.60	10.39	10.59	10.47
	訓練範例 RMS (1060)	14.02	13.72	14.02	13.82
	測試範例 RMS (1057)	5.29	5.271	5.29	5.272
墨西哥	全部範例判定係數	0.013	0.014	0.014	0.001
	全部範例 RMS	110.81	110.79	110.78	111.50
	訓練範例 RMS (1060)	105.55	105.40	105.44	106.58
	測試範例 RMS (1057)	115.573	115.63	115.576	116.05
台灣	全部範例判定係數	0.004	0.006	0.004	0.004
	全部範例 RMS	105.79	105.69	105.80	105.77
	訓練範例 RMS (1060)	129.13	128.94	129.15	129.06
	測試範例 RMS (1057)	75.48	75.470	75.472	75.55

將各國人均 GDP(美元, 2003)(United Nations Statistics Division, 2003)與 AR(1)模型的判定係數繪成散佈圖(圖 14), 可發現, 除了巴西之外, 人均 GDP 的對數值與 AR(1)模型的判定係數成反比, 可見高所得(高度經濟發展)國家的股票市場的市場效率比低所得(低度經濟發展)者高, 這與一般對效率市場的看法是吻合的。

表 5 各國人均 GDP(美元，2003) 與 AR(1)模型的判定係數之關係

國家	Log ₁₀ GDP	全部範例判定係數
美國	4.567	0.000
巴西	3.431	0.003
英國	4.482	0.001
香港	4.354	0.001
印尼	2.975	0.026
日本	4.529	0.001
韓國	4.043	0.005
馬來西亞	3.626	0.013
墨西哥	3.774	0.013
台灣	4.098	0.004

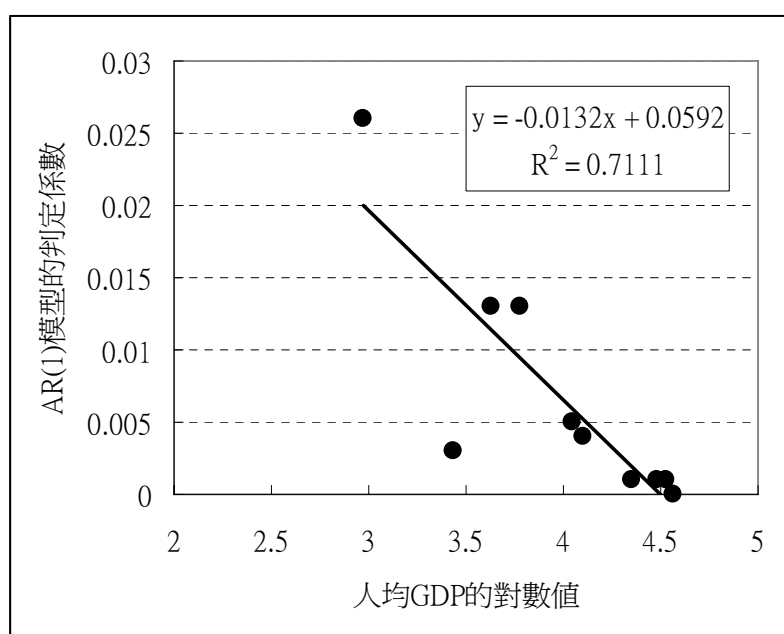


圖 14 各國人均 GDP(美元，2003) 與 AR(1)模型的判定係數之關係

4.6.3 BPN 分析

本研究以倒傳遞類神經預測十國股市時，網路架構設定為 2 個輸入層，8 個隱藏層，以及 1 個輸出層。其他設定如 1 次非季節差分，學習循環數目為 3000，學習速率初始值為 1，學習速率折減係數為 0.95，學習速率下限值為 0.1。表 6 為倒傳遞類神經網路預測十國股市之實驗結果。

表 6 倒傳遞類神經網路預測十國股市之 RMS 值

	所有範例	訓練範例	測試範例	BP
美國	30.09 (2117)	31.96 (1060)	28.08 (1057)	(1,0,2)
巴西	322.22 (2117)	297.16 (1060)	345.53 (1057)	(1,0,2)
英國	60.82 (2117)	71.24 (1060)	48.15 (1057)	(1,0,2)
香港	205.38 (2117)	264.94 (1060)	118.67 (1057)	(1,0,2)
印尼	10.112 (2117)	10.71 (1060)	9.48 (1057)	(1,0,2)
日本	203.37 (2117)	247.00 (1060)	147.16 (1057)	(1,0,2)
韓國	15.42 (2117)	17.64 (1060)	12.81 (1057)	(1,0,2)
馬來西亞	10.61 (2117)	14.03 (1060)	5.31 (1057)	(1,0,2)
墨西哥	110.86 (2117)	105.57 (1060)	115.93 (1057)	(1,0,2)
台灣	106.31 (2117)	129.64 (1060)	76.03 (1057)	(1,0,2)

4.6.4 ARIMA-BP 分析

表 7 為 ARIMA-BP 神經網路預測十國股市的結果。比較表 6 與表 7 中的測試範例之 RMS 值，可以發現 ARIMA-BP 的表現與倒傳遞神經網路不相上下，但是並未優於 ARIMA。

表 7 ARIMA-BP 預測十國股市之 RMS 值

	所有範例	訓練範例	測試範例	ARIMA-BP
美國	30.10 (2117)	31.96 (1060)	28.10 (1058)	(1,0,2,2)
巴西	322.26 (2117)	297.14 (1060)	345.63 (1058)	(1,0,2,2)
英國	60.82 (2117)	71.25 (1060)	48.15 (1057)	(1,0,2,2)
香港	205.26 (2117)	264.78 (1060)	118.65 (1057)	(1,0,2,2)
印尼	10.111 (2117)	10.70 (1060)	9.48 (1057)	(1,0,2,2)
日本	203.32 (2117)	246.97 (1060)	147.07 (1057)	(1,0,2,2)
韓國	15.41 (2117)	17.63 (1060)	12.80 (1057)	(1,0,2,2)
馬來西亞	10.58 (2117)	13.99 (1060)	5.30 (1057)	(1,0,2,2)
墨西哥	110.88 (2117)	105.48 (1060)	116.04 (1057)	(1,0,2,2)
台灣	106.23 (2117)	129.55 (1060)	75.96 (1058)	(1,0,2,2)

4.6.5 比較

比較表 4，表 6，以及表 7 的測試範例之 RMS 值，可以清楚發現 ARIMA、BP、ARIMA-BP 的表現不分軒輊。

伍、結論

改善時間序列預測模型的研究從未間斷過，在時間序列預測的研究中，ARIMA 模型已經成爲最爲人接受的方法之一。近年來，BPN 網路也證明了在時間序列預測的應用價值，特別是對非線性模型有非常顯著的處理能力。但相對於 ARIMA 而言，BPN 網路只使用到 AR 方面的資訊，而忽略 MA 方面的資訊，因此並不能保證它在各種預測環境下是最佳的方法。因此，本文提出了一種結合 ARIMA 與 BPN 網路來處理時間序列的 ARIMA-BPN 網路。其要點是將 MA 方面的資訊也成爲 BPN 網路的輸入，因此導致必須修改 BPN 的學習演算法。

本研究以六個人爲設計的例題，及一個現實世界的例題來比較 ARIMA、BPN 和 ARIMA-BPN 三者的效能。研究結果顯示：

1. 對以殘差值構成的非線性時間序列而言，ARIMA-BPN 有穩定且優於 ARIMA 與 BPN 的預測能力，在六個例題中有五個例題 ARIMA-BPN 是最佳方法。
2. 對真實世界的實例而言，ARIMA-BPN 並未優於 ARIMA 與 BPN。

由上述實驗結果可知，加入殘差值到 BPN 模型的 ARIMA-BPN 演算法能結合 BPN 與 ARIMA 的優點，改善 BPN 的預測能力，建立更精確的時間序列預測模型。但真實世界的實例如果不是穩定的非線性系統，則 ARIMA-BPN 並未優於 ARIMA。

參考文獻

- Bo, W., Shouyang, W., and Lai, K. K. (2007) "A hybrid ARCH-M and BP neural network model for GSCI futures price forecasting," *Lecture Notes in Computer Science*, Vol.4489, 917-924.
- Box, G. E. and Jenkins, G. M. (1970). *Time Series Analysis: Forecasting and Control*, San Francisco: Holden-Day.
- He, Y., Zhu, Y., and Duan, D. (2006) "Research on hybrid ARIMA and support vector machine model in short term load forecasting," *ISDA '06. Sixth International Conference on Intelligent Systems Design and Applications*, Jinan, China, 1, 804-809.
- Inoue, H., Fukunaga, Y., and Narihisa, H. (2001) "Efficient hybrid neural network for chaotic time series prediction," *Lecture Notes in Computer Science*, Vol. 2130, 712-718.

- Jain, A. and Kumar, A.M. (2007) "Hybrid neural network models for hydrologic time series forecasting," *Applied Soft Computing*, 7, 585–592.
- Law, R. (2000). "Back-propagation learning in improving the accuracy of neural network-based tourism demand forecasting," *Tourism Management*, 21(4) 331-340.
- Law, R. and Au, N. (1999). "A neural network model to forecast Japanese demand for travel to Hong Kong," *Tourism Management*, 20(1), 89-97.
- Pankratz, A. (1983). *Forecasting with Univariate Box-Jenkins Models: Concepts and Cases*. New York: John Wiley.
- Tseng, F.M., Yu, H.C., and Tzeng, G.H. (2001) "Combining neural network model with seasonal time series ARIMA model," *Technological Forecasting and Social Change*, 69(1), 71–87.
- Zhang, G. P. (2003) "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, 50 , 159–175.
- 王成財(2000)，林陳彥，洪水鴻，游珮瑛，蕭佩玉，莊峻瑜，潘清民，「結合類神經網路與時間序列預測台北地區臭氧濃度之研究」，第七屆人工智慧與應用研討會 (TAAI2002)論文集，53-58。
- 李振民(2001)，「以類神經網路為基礎之預測系統之研究」，中華大學土木工程研究所，碩士論文。
- 呂志峰(1999)，品質成本與品質管理之關聯性研究-關聯性研究-類神經網路類神經網路預測模式之運用，義守大學管理科學研究所，碩士論文。
- 張百棧，王彥文，楊雯寧(2004)，「混合式預測模式之發展—預測台灣股價指數之波動」，*Journal of the Chinese Institute of Industrial Engineers*, 21(4), 358-368.
- 湯健文(2003)，「類神經網路於因果關係模型與時間數列模型之應用」，中華大學土木工程研究所，碩士論文。
- 葉怡成(2005)，應用類神經網路，台北：儒林書局。
- 葉怡成(2006)，應用類神經網路—應用與實作，台北：儒林書局。
- 蔡裕春(2001)，台灣地區營造工程物價指數預測之研究—以類神經網路與 ARIMA 模式，輔仁大學應用統計學研究所，碩士論文。

國科會補助專題研究計畫項下出席國際學術會議心得報告

日期：100 年 7 月 28 日

計畫編號	NSC99-2221-E-216-040-		
計畫名稱	類神經網路演算法之改良(II)		
出國人員姓名	葉怡成	服務機構及職稱	中華大學資管系教授
會議時間	100 年 2 月 26 日至 100 年 2 月 28 日	會議地點	新加坡
會議名稱	(中文)2011 年第三屆機器學習與計算國際會議 (英文)2011 3rd International Conference on Machine Learning and Computing (ICMLC 2011)		
發表論文題目	(中文) 橢圓機率神經網路 (英文) Elliptical Probabilistic Neural Networks		

一、參加會議經過

此次 2011 ICMLC 國際研討會在 2011/2/26-28 新加坡召開。本人發表 Elliptical Probabilistic Neural Networks 這一篇是國科會 NSC-99-2221-E-216-040 的相關成果。

第一天 (2/26) 註冊

第二天 (2/27) 聽取學者演講

第一天 (2/28) 上午論文口頭報告

二、與會心得

這次會議的研究主題包含了許多機器學習的相關議題，並請學者專題演獎，其中 SVM、蟻群演算法似乎是重要趨勢。

三、建議

新加坡是一個風景秀麗的城市，觀光資源豐富。外國觀光客很多。

四、攜回資料名稱及內容

(1) 會議論文光碟一份

(2) 議程文件一份

Notification of Acceptance of the ICMLC 2011

February 26-28, 2011, Singapore

<http://www.icmlc.org/>



Dear I-Cheng Yeh, Kuan-Cheng Lin, Kuan-Chieh Huang, Xinying Zhang and Chong Wu,

Paper ID : L257

Paper Title : Elliptical Probabilistic Neural Networks

Congratulations! The review processes for 2011 3rd International Conference on Machine Learning and Computing (ICMLC 2011) have been completed. The conference received submissions from nearly 20 different countries and regions, which were reviewed by international experts, and about 200 papers have been selected for presentation and publication. Based on the recommendations of the reviewers and the Technical Program Committees, we are pleased to inform you that your paper identified above has been accepted for publication and oral presentation. You are cordially invited to present the paper orally at ICMLC 2011 to be held on, February 26-28, Singapore.

The ICMLC 2011 is sponsored by International Association of Computer Science and Information Technology (IACSIT), Singapore Institute of Electronics (SIE) and IEEE. The ICMLC 2011 has been listed in the IEEE Conference Calendar.

(Important) So in order to the register the conference and have your paper included in the proceeding successfully, you must finish following SIX steps.

1. Revise your paper according to the Review Comments in the attachment carefully.
2. Format your paper according to the Template carefully.
<http://www.icmlc.org/ieee.doc> (DOC Format)
3. Download and complete the Registration Form.
<http://www.icmlc.org/reg.doc> (English)
4. Finish the payment of Registration fee at the Bank. (The bank transfer information can be found in the Registration form)
<http://www.icmlc.org/reg.doc> (English)
5. Finish the IEEE Copyright Form
<http://www.icmlc.org/ieeecopyright.doc>

2011 3rd International Conference on Machine Learning and Computing(ICMLC 2011)

6. Send your final papers (both .doc and .pdf format), filled registration form (.doc format), copyright form (.jpg format) to us at icmlc@vip.163.com. **(Before January 15, 2011)**

ICMLC 2011 will check the format of all the registered papers first, so the authors don't need to upload the paper to the IEEE. After the registration, we will send all qualified papers to the IEEE for publishing directly.

If the above requirements are met by the set deadlines, All registered papers of ICMLC 2011 will be published into Conference proceeding by the IEEE Press. The accepted papers will be included in IEEE Xplore and indexed by INSPEC, ISI and EI Compindex.

Maybe some unforeseeable events could prevent a few authors not to attend the event to present their papers, so **if you and your co-author(s) could not attend ICMLC 2011 to present your paper for some reasons, please inform us. And we will send you, the official receipt of registration fee and proceedings after ICMLC 2011 free of charge.**

Please strictly adhere to the format specified in the conference template while preparing your final paper. If you have any problem in preparing the final paper, please feel free to contact us via icmlc@vip.163.com. For the most updated information on the conference, please check the conference website at <http://www.icmlc.org/> The Conference Program will be available at the website in Late January, 2011.

Finally, we would like to further extend our congratulations to you and we are looking forward to meeting you in Singapore!

Yours sincerely,



ICMLC 2011 Organizing Committees

<http://www.icmlc.org/>

Singapore

IEEE

Elliptical Probabilistic Neural Networks

I-Cheng Yeh / *Department of Information Management,
Chung Hua University*
Hsin Chu, Taiwan 30067, R.O.C.
E-MAIL: icyeh@chu.edu.tw

Kuan-Cheng Lin / *Department of Information
Management, Chung Hua University*
Hsin Chu, Taiwan 30067, R.O.C.
E-MAIL: m09410024@chu.edu.tw

Kuan-Chieh Huang / *Department of Computer Science
and Information Engineering, Cheng-Kung University*
Tainan, Taiwan 30067, R.O.C.
E-MAIL: m9104041@chu.edu.tw

Xinying Zhang / *Department of Management Science
and Engineering in School of Economy and
Management, Harbin Institute of Technology*
Harbin, China
E-MAIL: alpha3068@sina.com

Chong Wu / *Department of Management Science
and Engineering in School of Economy and Management,
Harbin Institute of Technology*
Harbin, China
E-MAIL: wuchong@hit.edu.cn

Abstract—The traditional Probabilistic Neural Networks (PNN) believes that all the variables have the same status, making the contour of probabilistic density function round. In this study, variable weights are added into the probabilistic density function of Elliptical Probabilistic Neural Network (EPNN), so that the kernel function can be adjusted into arbitrary hyper-ellipse to match the various shapes of classification boundaries. Although there are three kinds of network parameters in EPNN, including variable weights representing the importance of input variables, the core-width-reciprocal representing the effective range of data, and data weights representing the data reliability, in this study the principle of minimizing error sum of squares is used to derive the supervised learning rules for all the parameters with a unified mathematic theoretical framework. The performance of EPNN is testified and compared with MLP and PNN with 15 real classification applications. The results show that EPNN is more accurate than MLP and PNN.

Keywords- *probabilistic neural network, variable importance, classification, learning rule*

THEORY

Probabilistic Neural Network (PNN) [1-6] has a wide range of applications in model identification, time series prediction, as well as fault diagnosis and other fields [7-11] and the algorithm also has a number of different variations [3, 8-14]. For example, Specht [3] pointed out that PNN learned quickly from examples in one pass and asymptotically achieved the Bayes-optimal decision boundaries. The major disadvantage of a PNN was that it required one node or neuron for each training pattern. Various clustering techniques were proposed to change this requirement into one node per cluster center.

Berthold and Diamond [13] proposed a constructive training algorithm for probabilistic neural networks, a special type of radial basis function networks. In contrast to other algorithms, predefinition of the network topology was not required. The proposed algorithm introduced new

hidden units whenever necessary and adjusted the shape of existing units individually to minimize the risk of misclassification. This led to smaller networks compared to classical PNNs and therefore enabled the use of large data sets. Song, et al. [8] proposed a modified probabilistic neural network (PNN) for brain tissue segmentation with magnetic resonance imaging (MRI). In this approach, covariance matrices were used to replace the singular smoothing factor in the PNN's kernel function, and weighting factors were added in the pattern of summation layer. Rutkowski [10] proposed a new class of PNN working in nonstationary environment. He formulated the problem of pattern classification in nonstationary environment as the prediction problem and designed a probabilistic neural network to classify patterns with time-varying probability distributions. Rutkowski [11] also proposed a new class of generalized regression neural networks working in nonstationary environment, and proved convergence of the General Regression Neural Network (GRNN) based on general learning theorems.

The proposed EPNN has three kinds of network parameters:

- (1) The variable weights decide the shape of probabilistic density function so that the contour lines are not round but elliptical. The larger the variable weight of the variable, the more important the variable, and the smaller the radius of the ellipse in the direction of the variable;
- (2) The core-width-reciprocal is equivalent to the reciprocal of the smooth parameter in the traditional probabilistic density function, that is, the width-reciprocal of probabilistic density function. The larger core-width (the smaller core-width-reciprocal), the larger the effective range of the sample;
- (3) The data weight is equivalent to the height of probabilistic density function. The larger the height, the higher the credibility of the sample.

These three kinds of parameters, adjusted through training, can improve the accuracy of the model and can be employed to estimate the importance of input variables.

The contributions and innovations of this study are as follows.

(1) Hyper-elliptical probabilistic density function

The traditional PNN believes all the variables have the same status, making the contour of probabilistic density function round. In this study, variable weights are added into the probabilistic density function so that we can adjust the kernel function into arbitrary hyper-ellipse to match the various shapes of classification boundaries. This variable weight can be regarded as the kernel shape parameter of the probabilistic density function. Based on several public data sets, this study proves that the probabilistic density function with the shape parameter is able to greatly improve the accuracy of classification model; EPNN is significantly more accurate than a traditional PNN and even a Multi-Layered Perception.

(2) Unified mathematic theoretical framework

The traditional PNN often decides the smooth parameters (kernel radius parameters) of probabilistic density function with the trial and error method. Although there are three kinds of network parameters in EPNN, in this study, the principle of minimizing error sum of squares is used to derive the supervised learning rules for all the parameters, including kernel radius, shape, and height parameters of the probabilistic density function, with a unified mathematic theoretical framework.

In Section II, the learning rules of EPNN are derived. In the third section, we testify the performance of EPNN and compare it with MLP and PNN with 15 real classification applications. In Section IV, we make a summary of the testing results in the entire study.

THEORY

Probabilistic Neural Network (PNN) is a common neural network model, the theory of which is based on Bayesian classifier and the probabilistic density function [1-5]. Assuming that a classification problem has k classes as C_1, C_2, \dots, C_k , the classification rules are made by m -dimensional feature vectors as follows,

$$X = (X_1, X_2, \dots, X_m) \quad (1)$$

In this m -dimensional sample space, the classification probabilistic density function is the function of feature vectors $f_1(X), f_2(X), \dots, f_k(X)$, and for all $j \neq i$, the decision-making formula of Bayesian classifier is

$$h_i c_i f_i(X) > h_j c_j f_j(X) \quad (2)$$

where f_k is the probabilistic density function of the k -th category; c_k represents the cost of error in which the k -th category is misjudged; h_k is the prior probability of the k -th category.

Generally, the probabilistic density function above is the normal probabilistic density function, shown as follow,

$$f_a(X) = \frac{1}{(2\pi)^{\frac{m}{2}} \sigma^m} \left(\frac{1}{n_a} \right) \sum_{p=1}^{n_a} \exp\left(-\frac{(X - X_{ap})'(X - X_{ap})}{2\sigma^2}\right) \quad (3)$$

where $f_a(X)$ represents the value of probabilistic density function of Category A at point X ; m represents the number of input variables; σ represents the smooth parameter; n_a represents the number of training vectors in Category A; X represents the testing data vector; X_{ap} represents the p -th training data in Category A.

Because

$$\frac{1}{(2\pi)^{\frac{m}{2}} \sigma^m} \left(\frac{1}{n_a} \right) = \text{Constant} = h, \quad \text{and} \quad (4)$$

$$(X - X_{ap})'(X - X_{ap}) = \sum_{i=1}^m (x_i - x_i^{ap})^2 \quad (5)$$

the probabilistic density function can be simplified as follows,

$$f_a(X) = h \sum_{p=1}^{n_a} f_{ap} \quad (6)$$

and

$$f_{ap} = \exp\left(-\frac{\sum_{i=1}^m (x_i - x_i^{ap})^2}{2\sigma^2}\right) \quad (7)$$

where x_i represents the value of i -th input variable in the testing sample; x_i^{ap} represents the i -th input variable of the p -th sample of Category A in the sample base.

Although PNN can deal with classification problems, it cannot fulfill the function mapping. Hence, Specht [6] improved the network output with the weighted average method as follows,

$$y = \frac{\sum_{p=1}^n f_p \cdot t_p}{\sum_{p=1}^n f_p} \quad (8)$$

where t_p = the known output variable of the p -th sample in the sample base; f_p = the weight of the p -th sample in the sample base; n = number of samples in the sample base.

In this study, the probabilistic density function is modified as follows,

$$f_p = h_p^2 \exp\left(-\frac{\sum_{i=1}^m W_{ip}^2 (x_i - x_i^p)^2}{2\sigma_p^2}\right) \quad (9)$$

x_i = the i -th input variable of the unknown samples;
 x_i^p = the i -th input variable of the p -th sample in the sample base;
 W_{ip} = the i -th input variable weight of the p -th sample in the sample base, which decides the shape of the probabilistic density function so that the contour lines are not round but elliptical;
 h_p = the data weight of the p -th sample in the sample base, which is equivalent to the height of the probabilistic density function;
 σ_p = the smooth parameter of the p -th sample in the sample base, which is equivalent to the width of the probabilistic density function.

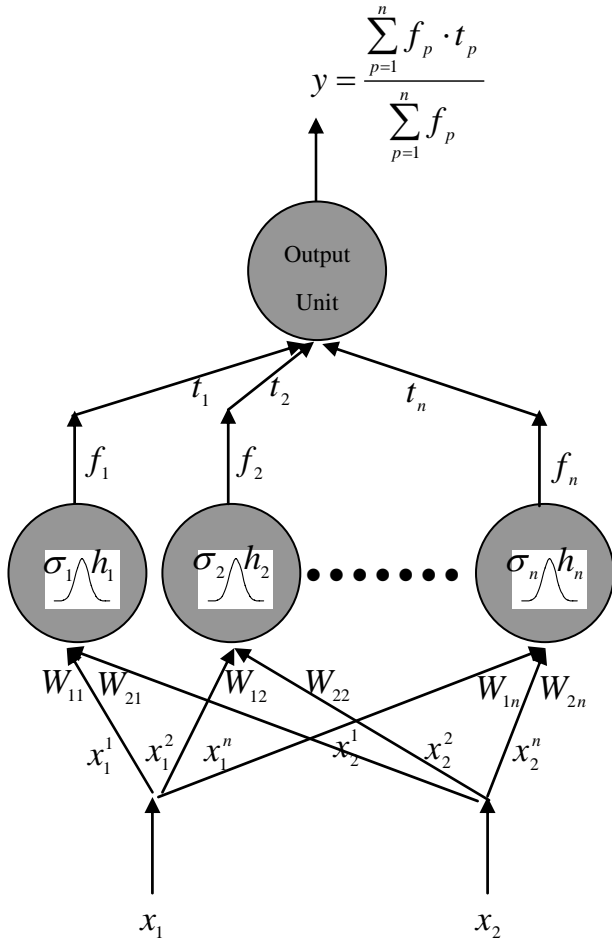


Figure 1. Elliptical Probabilistic Neural Network

The smooth parameter may be approaching to 0 in the learning process so as to make the denominator be zero. In

order to avoid the problem, the core-width-reciprocal is used instead of the smooth parameter.

Let

$$V_p = \frac{1}{\sqrt{2\sigma_p}} \quad (10)$$

where V_p = the core-width-reciprocal of the p -th sample in the sample base, which is equivalent to the width-reciprocal of the probabilistic density function,

then

$$f_p = h_p^2 \exp\left(-V_p^2 \sum_{i=1}^m W_{ip}^2 (x_i - x_i^p)^2\right) \quad (11)$$

Let

$$D_p \equiv V_p^2 \sum_{i=1}^m W_{ip}^2 (x_i - x_i^p)^2 \quad (12)$$

then

$$f_p = h_p^2 \exp(-D_p) \quad (13)$$

The output of the network is the weighted summation of the known value of output variable of the samples in the training set, as shown in Figure 1.

$$y_j = \frac{\sum_p f_p \cdot t_{pj}}{\sum_p f_p} \quad (14)$$

where t_{pj} = the known value of the j -th output variable of the p -th sample of the training set.

In order to derive the supervised learning rules to adjust the parameters in the equation above, let the error function be

$$E = \frac{1}{2} \sum_{j=1}^n (t_j - y_j)^2 \quad (15)$$

where n = the number of output variables; t_j = the known value of the j -th output variable of the training data; y_j = the inference value of the j -th output variable of the training data.

The goal to adjust the network parameters is to minimize the error function. Therefore, the supervised learning rules of network parameters can be derived with the steepest descent method as follows.

(1) Variable weights

$$\Delta W_{ip} = -\eta \frac{\partial E}{\partial W_{ip}}$$

(16)
where η = the learning rate.

Based on the chain rule in the partial differential, we conclude that

$$\frac{\partial E}{\partial W_{ip}} = \sum_j \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial W_{ip}}$$

(17)

Based on the chain rule in the partial differential, we conclude that

$$\frac{\partial y_j}{\partial W_{ip}} = \frac{\partial y_j}{\partial f_p} \frac{\partial f_p}{\partial D_p} \frac{\partial D_p}{\partial W_{ip}}$$

(18)

Substitute Eq. (18) into Eq. (17), we get

$$\frac{\partial E}{\partial W_{ip}} = \sum_j \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial f_p} \frac{\partial f_p}{\partial D_p} \frac{\partial D_p}{\partial W_{ip}}$$

(19)

where

$$\frac{\partial E}{\partial y_j} = -(t_j - y_j)$$

(20)

$$\frac{\partial y_j}{\partial f_p} = \frac{t_{pj} - \left(\frac{\sum_p f_p t_{pj}}{\sum_p f_p} \right)}{\sum_p f_p} = \frac{t_{pj} - y_j}{\sum_p f_p}$$

(21)

$$\frac{\partial f_p}{\partial D_p} = -h_p^2 \exp(-D_p)$$

(22)

and

$$\frac{\partial D_p}{\partial W_{ip}} = V_p^2 (2W_{ip} (x_i - x_i^p)^2)$$

(23)

Substitute Eq. (20), (21), (22), and (23) into Eq. (19), and then substitute them into Eq. (16), we can get

$$\Delta W_{ip} = -\eta \frac{\partial E}{\partial W_{ip}} = -2\eta \sum_j (t_j - y_j) \frac{t_{pj} - y_j}{\sum_p f_p} h_p^2 \exp(-D_p) \cdot V_p^2 \cdot W_{ip} \cdot (x_i - x_i^p)^2$$

(24)

Let δ_p be the gap between the p -th predictive output value, then its actual output value is

$$\delta_p \equiv \sum_{j=1}^n (t_j - y_j) (t_{pj} - y_j) h_p \exp(-D_p)$$

(25)

Eq. (24) can be simplified as follow,

$$\Delta W_{ip} = -2\eta \frac{1}{\sum_p f_p} \delta_p h_p V_p^2 W_{ip} (x_i - x_i^p)^2$$

(26)

(2) **Data weights**

$$\Delta h_p = -\eta \frac{\partial E}{\partial h_p} = 2\eta \sum_j (t_j - y_j) \frac{t_{pj} - y_j}{\sum_p f_p} h_p \exp(-D_p)$$

(27)

(27)

The equation above can be simplified by Eq. (25)

$$\Delta h_p = 2\eta \frac{1}{\sum_p f_p} \delta_p$$

(28)

(3) **Core-width-reciprocal**

$$\Delta V_p = -\eta \frac{\partial E}{\partial V_p} = -2\eta \sum_j (t_j - y_j) \frac{t_{pj} - y_j}{\sum_p f_p} h_p^2 \exp(-D_p) \cdot V_p \cdot \sum_i W_{ip}^2 (x_i - x_i^p)^2$$

(29)

(29)

The equation above can be simplified by Eq. (25), and we can get

$$\Delta V_p = -2\eta \frac{1}{\sum_p f_p} \delta_p \cdot h_p \cdot V_p \cdot \sum_i W_{ip}^2 (x_i - x_i^p)^2$$

(30)

)

These algorithms can gradually optimize the kernel shape, height, and radius parameters of the probabilistic density function in the learning process, so that each input variable of each sample has a various variable weight, making the probabilistic density function of each sample in an appropriate ellipse based on the training data, and thus eliminate the error of the model. This is the so called Elliptical Probabilistic Neural Network (EPNN).

APPLICATION EXAMPLES

In this section, we test the 15 real data sets in the UCI Machine Learning Repository [15] listed in Table 1 to compare the performances of EPNN, MLP and PNN. In order to test the effectiveness of learning, we use the 10-fold cross-validation, that is, all the data are randomly divided into ten sets. The test is performed ten times, and each time

a set is chosen as the testing data while the remaining nine sets are used as the training data. The integrated performance of all the test data is used to assess the accuracy of each network model.

Table 1. The descriptions of the data sets

Data set	Input	Output	Data
Iris	4	3	150
Insurance Company	7	2	700
Statlog (Shuttle)	9	3	5000
Glass Identification	9	6	214
Vowel Recognition	10	11	990
Wine	13	3	178
Forest Cover Type	14	7	4000
Letter Recognition	16	26	2000
Image Segmentation	18	7	2310
Statlog (Vehicle Silhouettes)	18	4	846
Statlog (German Credit)	19	2	1000
Statlog (Heart)	20	2	270
Thyroid Disease	21	3	7200
Statlog (Landsat Satellite)	36	6	6000
SPAMBASE	57	2	4500

Table 2. The comparison of the error rates in the application examples

Data set	MLP	EPNN	PNN	SVM
Iris	2.7	4	4	2.7
Insurance Company	36.49	34.48	35.63	33.65
Statlog (Shuttle)	0.35	0.3	0.33	0.4
Glass Identification	26.56	25.0	29.69	33.6
Vowel Recognition	41.13	42.64	48.48	40.52
Wine	1.72	0.0	3.45	0.56
Forest Cover Type	23.2	19.4	25.4	19.5
Letter Recognition	34.6	17.4	20.0	19.2
Image Segmentation	4.2	4.07	6.91	3.98
Statlog (Vehicle)	12.77	21.63	26.6	19.4
Statlog (German Credit)	25.67	25.1	27.0	23.65
Statlog (Heart)	15.71	12.86	15.71	15.87
Thyroid Disease	1.95	4.08	6.48	2.15
Statlog (Landsat Satellite)	5.09	7.17	10.04	8.7
SPAMBASE	3.63	5.11	12.33	6.6
Average	15.72	14.88	18.13	15.37

Table 2 shows the results of the models established by EPNN, PNN, MLP [19], and SVM (support vector machine) [16-18], illustrating that EPNN is more accurate than MLP and SVM, and much more accurate than PNN.

CONCLUSIONS

This study proposes Elliptical Probabilistic Neural Networks (EPNN) which has three kinds of network parameters: variable weights representing the importance of input variables, the core-width-reciprocal representing the effective width, and data weights representing the data reliability. These three kinds of parameters can be adjusted through training. We tested 15 real applications, and

compare EPNN with MLP, PNN, and SVM. The results showed that EPNN is more accurate than MLP and SVM, and much more accurate than PNN for the real classification applications.

Acknowledgments

This study is sponsored by the National Science Council (Project No. 99-2221-E-216-040).

References

1. Specht, D. F. "Probabilistic Neural Networks for Classification, or Associative Memory," *Proceedings of the 1988 IEEE International Conference on Neural Networks*, San Diego, Vol. 1, pp. 525-535 (1988).
2. Specht, D. F. "Probabilistic Neural Networks," *Neural Networks*, Vol. 3, No.1, pp. 109-118 (1990).
3. Specht, D.F. "Enhancements to probabilistic neural networks," *1992 International Joint Conference on Neural Networks*, Baltimore, Vol. 1, pp. 761-768 (1992).
4. Patra, P. K., Nayak, M., Nayak, S. K., and Gobbak, N. K., "Probabilistic Neural Network for Pattern Classification," *Proceedings of the 2002 International Joint Conference on Neural Networks*, Honolulu, Vol. 2, pp. 1200-1205 (2002).
5. Parzen, E. "On Estimation of a Probability Density Function and Mode," *Annals of Mathematical Statistics*, Vol. 33, No.3, pp. 1065-1076 (1962).
6. Specht, D. F. "A General Regression Neural Network," *IEEE Transactions on Neural Networks*, Vol. 2, No.6, pp. 568-576 (1991).
7. Adeli, H. and Panakkat, A. "A probabilistic neural network for earthquake magnitude prediction," *Neural Networks*, Vol. 22, No. 7, pp. 1018-1024 (2009).
8. Song, T., Jamshidi, M.M., Lee, R.R., and Huang, M. "A Modified Probabilistic Neural Network for Partial Volume Segmentation in Brain MR Image," *IEEE Transactions on Neural Networks*, Vol. 18, No. 5, pp. 1424-1432 (2007).
9. Yu, S. N. and Chen, Y. H. "Electrocardiogram beat classification based on wavelet transformation and probabilistic neural network," *Pattern Recognition Letters*, Vol. 28, No. 10, pp. 1142-1150 (2007).
10. Rutkowski, L. "Adaptive probabilistic neural networks for pattern classification in time-varying environment," *IEEE Transactions on Neural Networks*, Vol. 15, No. 4, pp. 811-827 (2004).
11. Rutkowski, L. "Generalized Regression Neural Networks in Time-Varying Environment," *IEEE Transactions on Neural Networks*, Vol. 15, No. 3, pp. 576-596 (2004).
12. Chen, N., Sun, F., Ding, L., and Wang, H. "An adaptive PNN-DS approach to classification using multi-sensor information fusion," *Neural Computing & Applications*, Vol. 18, No. 5, pp. 455-467 (2009).
13. Berthold, M. R. and Diamond, J. "Constructive training of probabilistic neural networks," *Neurocomputing*, Vol. 19, No. 1, pp. 167-183 (1998).

14. Burrascano, P. "Learning Vector Quantization for the Probabilistic Neural Network," *IEEE Transaction on Neural Networks*, Vol. 2, No.4, pp. 458-461 (1991).
15. Frank, A. & Asuncion, A. UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science (2010).
16. Chang, C.C., and Lin, C. J. LIBSVM: a library for support vector machines, Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm> (2001).
17. Hsu, C. W. and Lin, C. J. "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, Vol. 13, No. 2, 415-425 (2002).
18. Meyer D., Leisch, F., and Hornik, K. "The support vector machine under test," *Neurocomputing*, Vol. 55, No. 1-2, 169-186 (2003).
19. Yeh, I-Cheng and Cheng, Wei-Lun, "First and Second Order Sensitivity Analysis of MLP," *Neurocomputing*, Vol.73, No.10, 2225– 2233 (2010).

國科會補助計畫衍生研發成果推廣資料表

日期:2011/07/03

國科會補助計畫	計畫名稱: 類神經網路演算法之改良(II)
	計畫主持人: 葉怡成
	計畫編號: 99-2221-E-216-040- 學門領域: 人工智慧
無研發成果推廣資料	

99 年度專題研究計畫研究成果彙整表

計畫主持人：葉怡成		計畫編號：99-2221-E-216-040-				
計畫名稱：類神經網路演算法之改良(II)						
成果項目		量化			備註(質化說明：如數個計畫共同成果、成果列為該期刊之封面故事...等)	
		實際已達成數 (被接受或已發表)	預期總達成數 (含實際已達成數)	本計畫實際貢獻百分比		
國內	論文著作	期刊論文	2	1	20%	1. I-Cheng Yeh and Chung-Chih Chen (2011). 'Hybrid Transfer Function Networks,' Journal of Information Technology and Applications, Vol.5, No.1, 37-45. 2. I-Cheng Yeh, Kuan-Cheng Lin, Kuan-Chieh Huang, Xinying Zhang, and Chong Wu (2011), 'Elliptical Probabilistic Neural Networks,' Journal of Information Technology and Applications. (accepted)
		研究報告/技術報告	1	1	10%	
		研討會論文	0	0	10%	
		專書	0	0	10%	
	專利	申請中件數	0	0	10%	件
		已獲得件數	0	0	10%	
	技術移轉	件數	0	0	10%	件
		權利金	0	0	10%	
	參與計畫人力 (本國籍)	碩士生	2	0	10%	人次
		博士生	0	0	10%	
		博士後研究員	0	0	10%	
		專任助理	0	0	10%	

國外	論文著作	期刊論文	4	2	20%	篇	1. I-Cheng Yeh, Che-hui Lien, Tao-Ming Ting, Yi-Yun Wang, Chin-Ming Tu (2010/10), ' Analysis of Cosmetics Purchase Behavior Using Association Reasoning Neural Networks,' Expert Systems with Applications, Vol. 37, No. 10, pp. 7219-7226. (SCI)
		研究報告/技術報告	0	0	10%		2. I-Cheng Yeh, Che-hui Lien, Yi-Chen Tsai (2011/1), ' Evaluation Approach to Stock Trading System Using Evolutionary Computation,' Expert Systems with Applications, Vol. 38, No.1, 794-803. (SCI)
		研討會論文	1	1	10%		3. I-Cheng Yeh and Kuan-Cheng Lin (2011), ' Supervised learning probabilistic neural networks,' Neural Processing Letters. (SCI) (accepted)
		專書	0	0	10%		4. Yeh, I-Cheng., Chen, C., Zhang, X., Wu, C., & Huang, K. (2010). ' Adaptive radial basis function networks with kernel shape parameters,' Neural Computing Applications. (SCI) Springer London. Retrieved from http://www.springerlink.com/content/d561202510w65742/
	專利	申請中件數	0	0	10%	件	I-Cheng Yeh and Che-hui Lien (2011). ' Fuzzy Rule-Based Stock Trading System,' 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), Taipei, June 27-30, 2011.
		已獲得件數	0	0	10%		
	技術移轉	件數	0	0	10%	件	
		權利金	0	0	10%		
	參與計畫人力 (外國籍)	碩士生	0	0	10%	人次	
		博士生	0	0	10%		
		博士後研究員	0	0	10%		
		專任助理	0	0	10%		

其他成果
 (無法以量
 化表達之
 成果如辦
 理學術活
 動、獲得獎
 項、重要國
 際合作、研
 究成果國
 際影響力
 及其他協
 助產業技
 術發展之
 具體效益
 事項等,請
 以文字敘
 述填列。)

無

	成果項目	量化	名稱或內容性質簡述
科 教 處 計 畫 加 填 項 目	測驗工具(含質性與量 性)	0	
	課程/模組	0	
	電腦及網路系統或工 具	0	
	教材	0	
	舉辦之活動/競賽	0	
	研討會/工作坊	0	
	電子報、網站	0	
	計畫成果推廣之參與 (閱聽)人數	0	

國科會補助專題研究計畫成果報告自評表

請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）、是否適合在學術期刊發表或申請專利、主要發現或其他有關價值等，作一綜合評估。

1. 請就研究內容與原計畫相符程度、達成預期目標情況作一綜合評估

達成目標

未達成目標（請說明，以 100 字為限）

實驗失敗

因故實驗中斷

其他原因

說明：

2. 研究成果在學術期刊發表或申請專利等情形：

論文： 已發表 未發表之文稿 撰寫中 無

專利： 已獲得 申請中 無

技轉： 已技轉 洽談中 無

其他：（以 100 字為限）

3. 請依學術成就、技術創新、社會影響等方面，評估研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）（以 500 字為限）

本研究提出一個結合倒傳遞網路與 ARIMA 優點的 ARIMA-BPN 神經網路，它是以 BPN 為模型，將 ARIMA 模式的輸入，包括前時 p 個時刻的數列值與前 q 個時刻的數列殘差值做為輸入值，組成非線性函數，以建立更準確的時間數列預測模型。因為數列殘差值在 BPN 的訓練過程中會因網路連結權值的調整而改變，因此必須修改 BPN 的演算法來適應此需求，即藉由不斷更新每次預測所得之殘差值做為網路的輸入值。研究結果顯示，ARIMA-BPN 神經網路演算法在部份例題比 ARIMA 與 BPN 方法更準確。本研究與其它文獻不同之處與創新的貢獻簡述如下：

1. ARIMA-BPN 神經網路有別於其他複合法，它真正將 ARIMA 的概念融入 ANN 中，產生一種新的 BPN 演算法。

2. 財務領域有大量的時間數列問題，例如匯率、股價波動等，以往常以 ARIMA 或 BPN 建模。ARIMA-BPN 的非線性時間數列建模能力或許是一個可以突破這類難題的工具。