類神經網路：理論與實務
期末專題報告
鑽石價格預測

統計學程一 r08h41001 廖容嫻
統計學程一 r08h41009 邱以安
統計學程一 r08h41015 吳東霖

December 30, 2019

目錄

1 緒論

鑽石，又稱金剛石，由碳元素組成，是目前已知最硬的天然物質。鑽石硬度極高與導熱性極高的特性，使其在工業切割、積體電路上有廣泛的應用。雖然在人工鑽石的製程問世後，天然鑽石在工業應用之價值逐漸下降；然而其通透亮麗的光澤仍具極高的收藏價值。此外，一顆美麗的鑽石因為有著純淨的色澤、且在燈光下顯得閃閃生輝，因而廣受世人喜愛，往往被人製成裝飾品穿戴於身。「鑽石恆久遠，一顆永流傳」，在此著名廣告詞問世後，鑽石彷彿成為人們證明感情、擁有榮華富貴的固定象徵，許多名人爭相配戴，使得其價格也跟著水漲船高。有著商機，便自然有人投入，以此衍生的產業鍊，從開採原石、加工、珠寶商的銷售，到鑑定寶石，提供許多工作機會，深深影響了現代社會的經濟與發展。

2 研究動機與目的

鑽石有著無窮的魅力，但若對購買的鑽石有美觀上的需求，往往需要撒下大筆金錢，也相對產生了巨大的經濟壓力，因此價格成為了大部分買家首要考慮的條件。越是美麗的鑽石，價格也相對越高，但「美麗」並沒有明確的定義，只能量測鑽石的物理量、專家評價做為參考；此外，鑽石屬於奢侈品，若缺乏也不致生活困難，買家基本生活無虞，因此對買家而言計畫預算時不需完全精確，只需概略範圍便可進行預算控制，買到想要的鑽石。因此適合以類神經網路方法預測鑽石價格。

隨著機器學習研究與發展，以及現代電腦性能的加強，類神經網路已然成為當代顯學之一，不僅方法日趨成熟，處理大量資料的耗時也降低許多。本研究將藉由鑽石量測出的大小、重量、切割、色澤、淨度，使用不同方法預測鑽石的價格：

- 倒傳遞類神經網路 (BPNN)
- 輻狀基底函數類神經網路 (RBFNN)
- 支援向量機推廣之支持向量迴歸 (SVR)

並以適當的指標衡量模型的優劣：

- 均方誤差 (MSE)：衡量實際值與預測值的誤差程度

- 決定係數 (R-square)：衡量此模型比起以平均數預測而言的優劣程度

進而在三者中分別找出使得模型最準確的參數，最後在最佳參數的情況下比較三個模型的優缺。

3 參考文獻

張斐章、張麗秋 (民 103)。類神經網路導論：原理與應用。新北市：滄海

4 資料來源與變數介紹

我們的資料是來自於 Kaggle 這個開放的網站，從這個網站上我們找到了鑽石價格與其他特性的資料，以下顯示部分資料：

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

其網址為：https://www.kaggle.com/shivam2503/diamonds

接著，我們將依序介紹每個變數代表的含意：

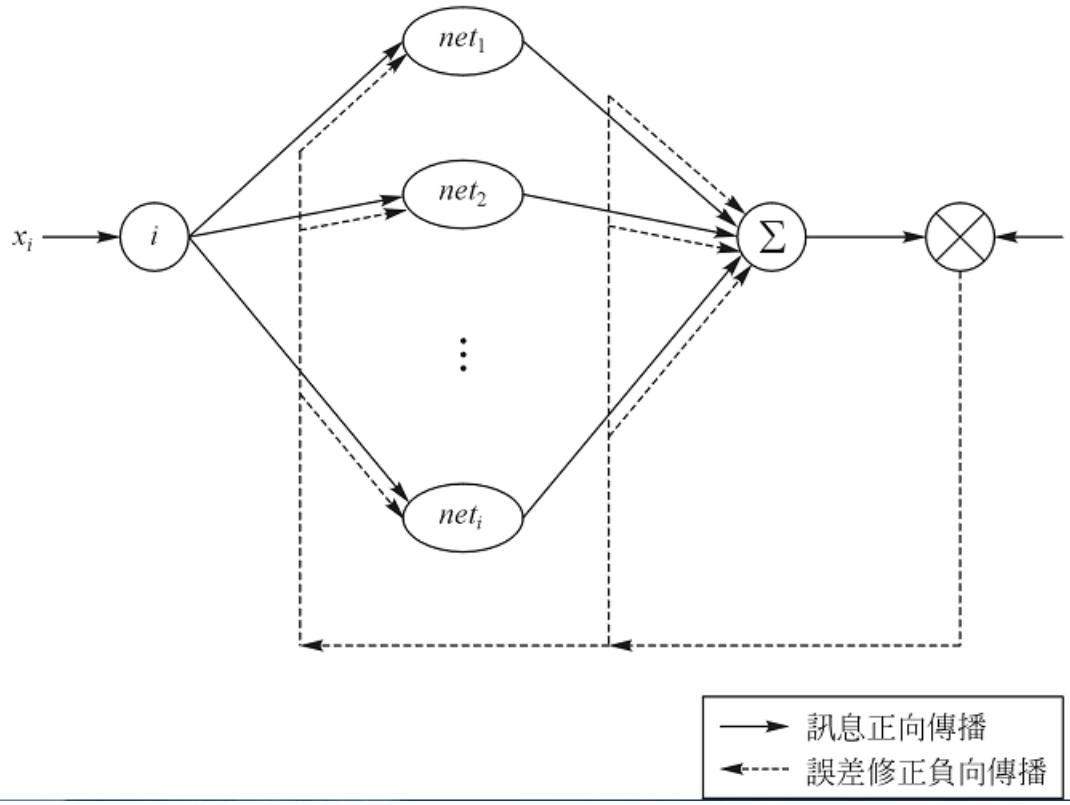
- 價格 price：以美元計 ($326 – $18,823)
- 克拉 carat：鑽石的重量單位 (0.2–5.01)
- 切割 cut：鑽石切割程度的好壞 (由壞到好分成 Fair, Good, Very Good, Premium, Ideal)
- 顏色 color：鑽石的顏色，從 J (最差) 到 D (最好)
- 清晰程度 clarity：鑽石的清晰程度 (由最差到最好分為 I1, SI2, SI1, VS2, VS1, VVS2, VVS1, IF)
- 長度 x：以毫米為單位 (0–10.74)
- 寬度 y：以毫米為單位 (0–58.9)
- 深度 z ：以毫米為單位 (0–31.8)

- 深度 (百分比) depth：以鑽石腰部為 100%，看深度佔其多少比例，其公式為 z / mean(x, y) = 2 * z / (x + y) (43–79)
- 桌面 (百分比) table：以鑽石腰部為 100

5 分析方法

5.1 倒傳遞類神經網路 (BPNN)

倒傳遞類神經網路屬於多層前饋式網路，並且屬於監督式的學習。其網路架構包含一層輸入層、一個至多個隱藏層與一層輸出層。而倒傳遞類神經的特點在於，由於模型在學習的過程中有實際的輸出值可以與估計的輸出值進行比較，若估計的輸出值與實際值相差太多，模型會以最陡波降法進行權重與偏權值的修正，因此才被稱謂倒傳遞的類神經網路。



- 第 n 層第 j 個神經元的輸入值為 n-1 層的輸出值總和後的函數：

$$y_j^n = f(net_j^n)$$

- 順時誤差 E 定義為網路實際輸出值與目標輸出值的差：

$$E = \frac{1}{2}\sum_k (d_k - y_k)^2$$

- 連結權重的修正值為：

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} = \eta \delta_j^n y_i^{n-1}$$

其中，

當 wji 在輸出層與隱藏層之間：

$$\delta_j^n = (d_j - y_j^n) f'(net_j^n)$$

當 wji 在隱藏層與輸入層之間：

$$\delta_j^n = \Big[\sum_k \eta_k^{n+1} w_{kj} \Big] f'(net_j^n)$$

- 權重更新方式：

$$w_{ji}(p) = w_{ji}(p-1) + \Delta w_{ji}$$

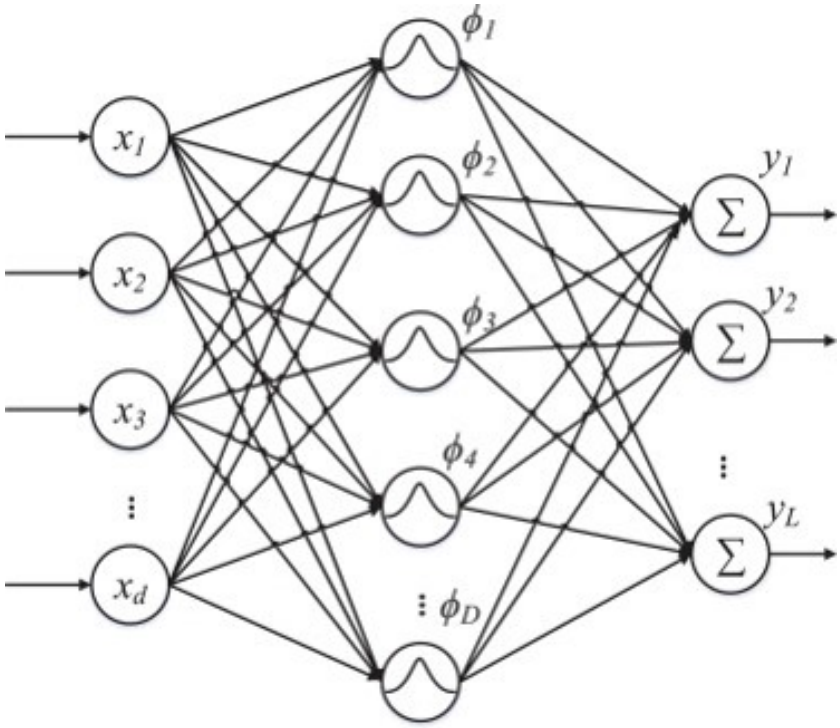
而在倒傳遞類神經網路的使用上，其優點為學習精度高、回想速度快、輸入/輸出值可為連續與不連續值，以及能處理複雜的非線性函數合成問題等。而其缺點是學習速度較慢、執行時間較長以及可能陷入局部最小值等問題。

5.2 輻狀基底函數類神經網路 (RBFNN)

輻狀基底函數 (Radial Basis Function) 類神經網路 (RBFNN)，或稱為半徑式類神經網路，其特質主要再模擬大腦皮質層軸突的局部調整功能，具備良好的映射能力；其架構與多層感知器相同，具有輸入層、一層隱藏層與輸出層，是屬於基本前饋式類神經網路的組合，其優點在於可大量減少學習時間。

其中輸入層為輸入資料與網路連結的介面；隱藏層式將輸入資料經過非線性活化函數 (本次使用 ReLu 及 sigmoid 為例) 轉換到隱藏層，也就是將輸入空間進行非線性映射到隱藏層，此神經元個數會直接影響輸入與輸出間的關係，越高維度的隱藏層空間可得到越精確的近似推估值；輸出層則是將隱藏層的輸出進行線性組合 (或線性映射) 以獲得輸出值。

RBF 類神經網路以函數逼近 (curve fitting) 的方式建構網路，最早衍生於解決多變量內插 (interpolation) 問題，以不規則位置的資料點求得相對應的幅狀基底函數 ($\phi(\|x - c\|)$) 來解決。



- 隱藏層個神經元輸出：

$$z_j(x) = \phi(\|x - c_j\|)$$

其中 ϕ 表幅狀基底函數，c_j 表養藏層地 j 個神經元中心點，$\|x - c_j\|$ 表 x 與 c_j 間之歐式距離

- 網路輸出值為：

$$y = \sum_j w_j \cdot \phi(\|x - c_j\|) + w_0$$

- 隨機選取法：

從範例資料隨機選取固定個數的中心點，並採用伸展高度相同的高斯函數

$$\phi(\|x - c_j\|) = exp(-\frac{m_1}{2d_{max}^2}\|x - c_j\|^2), i = 1, 2, ..., m_1$$

優點為快速容易，缺點為須具備大量資料才夠隨機選取

- 最小平方法求權重向量：

$$E = \frac{1}{2}\sum_p (d(p) - y(p))^2$$

$d(p)$ 和 $d(p)$ 為第 p 個訓練範例資料的目標輸出值及網路輸出值，求訓練範例資料誤差平方和最小。

5.3 支援向量機推廣之支持向量迴歸 (SVR)

支持向量機 (SVM) 屬於前饋式類神經網路，是一種監督式學習之分類法，屬於二元分類之學習機 (learning machines)，在處理可分離分類問題時，搜尋最佳超平面 (optimal hyperplanes) 使得兩類別間之決策間隔最大，讓兩不同分類之資料點的區分程度越高；處理非線性可分離問題時，先將原始資料轉換至高維度空間後再進行前述分類處理。

而將支持向量機演算法推廣至通用的迴歸模式，稱為「支持向量迴歸」，使用損失函數使迴歸模式可產生之誤差容忍區，此時系統輸出為數值型態。其計算方法簡述如下：

- 損失函數定義：

$$L_\varepsilon(d,y) = |d-y|_\varepsilon = max(0, |d-y| - \varepsilon)$$

- 迴歸模式定義為：

$$y = w^t \Phi(x) + b$$

- 對於 N 筆訓練資料，風險函數定義為：

$$minf = \frac{1}{2}\|w\|^2 + C\Sigma_{i=1}^{N}|d-y|_\varepsilon$$

$$minf(w,\xi,\xi_i') = \frac{1}{2}\|w\|^2 + C\Sigma_{i=1}^{N}(\xi_i + \xi_i')$$

其中 C 為常數，用於決定訓練資料誤差與懲罰項之權衡，ξ_i 與 ξ_i' 為兩組寬鬆變數，若 d_i 與 y_i 誤差小於 ε，則 $\xi_i = 0$,$\xi_i' = 0$。

最佳化問題使用 Lagrange 乘數並對微分為 0，將迴歸方程式改寫並定義核心函數 (kernel functions)，可以用不同方程式帶入。

6 分析結果

為使分析結果容易比較，在進行三種不同的分析方法前，先將資料使用 Python 打亂後，再依照 60%、20% 與 20% 的比例依序將資料分成三等份，並且往後的三個方法均使用此分配好的資料集進行模型的建構，以示公平。

首先，因為此份資料中有類別變數，因此我們先進行資料的前處理。得先將其轉換成啞變數後才能放入模型中，接著為使連續型資料在模型中的影響力都相同，可將每個變數都進行標準化的程序，也就是各自減去其變數的平均值在除以標準差，而類別型的變數可以省去此流程。得到標準化後的數據如下：

carat	depth	table	price	x	y	z	cut_Fair	cut_Good	...	color_I	color_J	clarity_I1	clarity_IF	clarity_SI1	clarity_SI2
-0.121363	-0.032446	-1.103130	-0.457455	0.088941	0.116526	0.100990	0	0	...	0	1	0	0	0	0
-0.818707	-1.353750	1.142503	-0.735473	-0.775771	-0.813658	-0.923505	0	0	...	0	0	0	0	0	0
1.083141	-0.588785	-0.654004	1.909591	1.283491	1.150065	1.097416	0	0	...	0	0	0	0	0	0
0.322402	-3.648647	1.591630	-0.179186	0.712959	0.659134	0.199229	0	1	...	0	0	0	0	0	1
0.385797	-2.118716	-0.204877	-0.501154	0.721874	0.598844	0.367639	1	0	...	0	0	1	0	0	0

將資料標準化後，下一步便是將資料切割為 trainx 與 trainy，trainy 為在訓練資料集中，我們想要預測的價格 (price) 此一變數，而 trainx 則為除去價格 (price) 後的所有變數。

6.1 倒傳遞類神經網路 (BPNN)

1. 建立模型

   將資料完成處理前期後，就可以開始建立模型了。在倒傳遞類神經網路的架構下，使用的是一層隱藏層與一層輸出層的模型。其輸入層有 25 個變數 (含所有啞變數)，中間的隱藏層設置了 20 個神經元，而輸出層便是要被預測的價格這一變數。

   而一開始權重的設定是採用從均勻分布 (Uniform[0,1]) 中隨機取 20 個數值作為初始權重，而活化函數則是採用整流線性單位函数 (Rectified Linear Unit, ReLU)，損失函數的部分是用均方誤差 (Mean Square Error)，而優化函數則是採用 Adam 的方式進行。

   接下來會進行兩種不同的方式建立模型並進行比較，在同樣都是迭代 100 次的情況下，觀察不同模型對於測試資料集的表現有何不同。

2. 使用單一隱藏層，比較 5、10 與 20 個神經元模型的預測能力

隱藏層神經元個數	5 個神經元	10 個神經元	20 個神經元
測試的 RMSE	645.62	602.22	571.33
測試的 R 平方	0.935	0.933	0.993

在使用單一隱藏層，使用不同數目的神經元建立模型時，能發現當使用越多的神經元，其 RMSE 會隨之下降，且 R 平方會提高。十分符合我們在課堂上所學到的理論。

而接下來便要測試，當增加了隱藏層層數時，模型預測的狀況會如何改變。

3. 使用兩層隱藏層，每層使用 0-3 個神經元，比較其模型的預測能力

神經元個數	(0 , 0)	(0 , 1)	(0 , 2)	(0 , 3)
測試的 RMSE	X	762.21	654.15	642.85
測試的 R 平方	X	0.957	0.973	0.963

神經元個數	(1 , 0)	(1 , 1)	(1 , 2)	(1 , 3)
測試的 RMSE	1093	753.62	799.08	718.14
測試的 R 平方	0.931	0.955	0.954	0.966

神經元個數	(2 , 0)	(2 , 1)	(2 , 2)	(2 , 3)
測試的 RMSE	1036.8	668.29	654.73	664.19
測試的 R 平方	0.929	0.968	0.972	0.971

神經元個數	(3 , 0)	(3 , 1)	(3 , 2)	(3 , 3)
測試的 RMSE	645.62	788.27	636.19	630.35
測試的 R 平方	0.948	0.918	0.986	0.958

由以上的結果可以發現，在所有 15 種的神經元個數組合下所建立的模型，為單一層有 3 個神經元的隱藏層、兩層隱藏層分別為 (3 , 2) 個神經元以及兩層隱藏層分別為 (3 , 3) 個神經元此三組模型的預測能力最佳，即其 RMSE 最小，分別為 642.85、636.19 及 630.35。

由於使用單層隱藏層的結果較好，因此我們再來比較一下若使用不同活化函數結果會如何?

4. 將活化函數變更為 Sigmoid，且使用單一隱藏層，比較 5、10 與 20 個神經

元模型的預測能力

隱藏層神經元個數	5 個神經元	10 個神經元	20 個神經元
測試的 RMSE	633.64	601.08	598.06
測試的 R 平方	0.982	0.980	0.989

由以上的測試可知，在使用倒傳遞類神經網路下所建構的預測模型，在其他參數不變的情況下 (包括使用相同的函數與迭代次數)，單就隱藏層個數來看，其實使用單一層隱藏層 10 個神經元 (RMSE = 602.22)，就能比使用兩層隱藏層、每層各 3 個神經元 (RMSE = 630.35) 的預測能力來的更好，而使用單一隱藏層 20 個神經元 (RMSE = 571.33) 的預測能力又比 10 個提昇非常多。

再來就比較同樣是單一隱藏層同樣個數的神經元，但比較不同的活化函數的預測能力。其中使用 ReLu 的 RMSE 均比 Sigmoid 的 RMSE 來的小。而其中最小的還是使用單一隱藏層 20 個神經元且活化函數為 ReLu 的表現最好。

另外，使用單一隱藏層的模型訓練時間，雖然神經元個數較多，但時間卻沒有比使用兩層隱藏層的模型還來的多。因此個人認為，在較乾淨且簡單的資料下，使用倒傳遞類神經網路訓練模型時，只要使用一層隱藏層即可有十分優異的結果。

6.2 輻狀基底函數類神經網路 (RBFNN)

與 BPNN 相同，以 Adam 優化器訓練，迭代 100 次數，分別比較單層不同神經元 (5,10,20)、不同活化函數 (ReLu,Sigmoid)，以及兩層 0 到 3 個神經元組合。我們以 RMSE 和 R-squared 為判斷模醒好壞的參考數值，以下列出比較表格和 Sigmoid 的圖例，

- ReLu 函數下不同神經元

隱藏層神經元個數	5 個神經元	10 個神經元	20 個神經元
測試的 RMSE	1690	2025.37	1327.08
測試的 R 平方	0.819	0.74	0.888

- Sigmoid 函數下不同神經元

隱藏層神經元個數	5 個神經元	10 個神經元	20 個神經元
測試的 RMSE	3623.06	3503.89	2852.76
測試的 R 平方	0.167	0.22	0.483

我們可以看見在 ReLu 及 Sigmoid 函數中越神元數目越多的誤差越小，估計越準確，符合 RBFNN 隱藏層維度越高越準確的網路架構。又不同活化函數的比較中 ReLu 表現優於 Sigmoid。

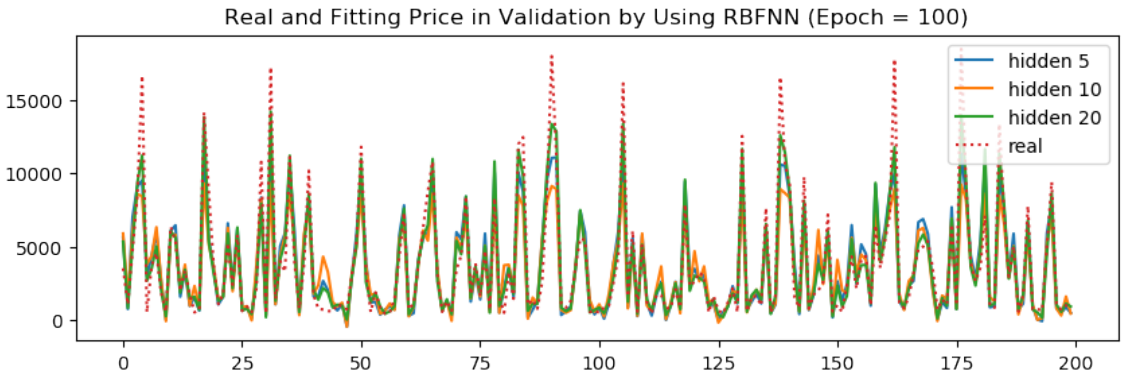
- 雙層隱藏層

雙層隱藏層的設置中第一層為 RBFNN，活化函數為 ReLu，第二層為 BP，活化函數為 Sigmoid，我們可以看見神經元個數有較大的影響力，不論是同一層或不同層，增加一個神經元所減少的誤差值都相對大。

神經元個數	(1 , 0)	(1 , 1)	(1 , 2)	(1 , 3)
測試的 RMSE	3968.84	1229.99	1183.62	1434.73
測試的 R 平方	0	0.904	0.911	0.87

神經元個數	(2 , 0)	(2 , 1)	(2 , 2)	(2 , 3)
測試的 RMSE	1452	1014	1347.16	1347.07
測試的 R 平方	0.86	0.935	0.885	0.885

神經元個數	(3 , 0)	(3 , 1)	(3 , 2)	(3 , 3)
測試的 RMSE	1117.46	1002.34	1134.61	926.6
測試的 R 平方	0.921	0.936	0.918	0.945



使用 RBFNN，活化函數為 ReLu，一層隱藏層分別為 5、10、20 個神經元，取測試集前 200 筆畫圖，相較 BPNN，RBFNN 的 RMSE 較大，我們從圖中可以看

出 RBFNN 的估計值較平滑，對於價格較高者會低估。各模型之計算時間皆在一分鐘以內。

6.3 支援向量機推廣之支持向量迴歸 (SVR)

支持向量迴歸 (SVR)

- 建立模型

  在支持向量迴歸的架構下訓練模型，其輸入層有 25 個變數 (含所有啞變數)，輸出層為欲預測之鑽石價格。

  以四種不同的核心函數 (kernel function) 建立模型，分別為：linear(線性函數)、poly(多項式函數，預設為三次)、rbf(輻狀基底函數)、sigmoid 函數，在設立同樣停止準則的情況下，比較不同模型對於測試資料集的表現有何不同。

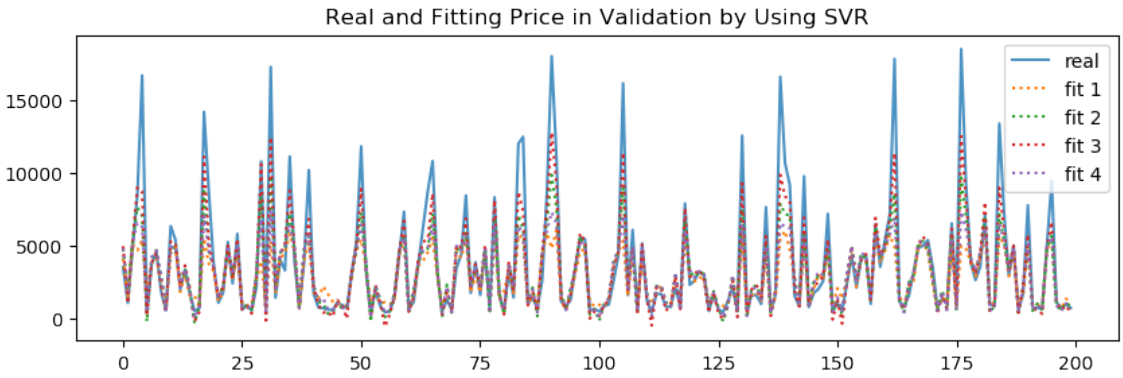
- 使用訓練集訓練模型完畢後，發現在參數選擇方面，由於限制迭代次數過少會使模型運算成效不彰，RMSE 與 R-square 的表現皆差強人意，無法進行良好的比較，因此不限制迭代次數，只設立停止準則：容忍量為 0.001。

- 在測試步驟，使用 linear(線性函數)、poly (多項式函數，預設為三次)、rbf(輻狀基底函數)、sigmoid 函數，觀察其 RMSE 與 R-square，比較其預測能力。

核心函數	rbf	poly	linear	sigmoid
測試 RMSE	4449.915	4711.118	5037.435	4570.476
測試 R 平方	0.459	0.723	0.85	0.613

由以上的測試可知，RMSE 的表現以 rbf(輻狀基底函數) 為最佳；R 平方的表現以 linear(線性函數) 為最佳。

另外，以模型訓練時間討論，四者的模型運算時間都不長，rbf 為 72 秒，polynomial 為 54 秒，linear 為 47 秒 sigmoid 為 83 秒，顯示在此問題中以 linear 為相對最快且最準確的核心函數，圖型如下：

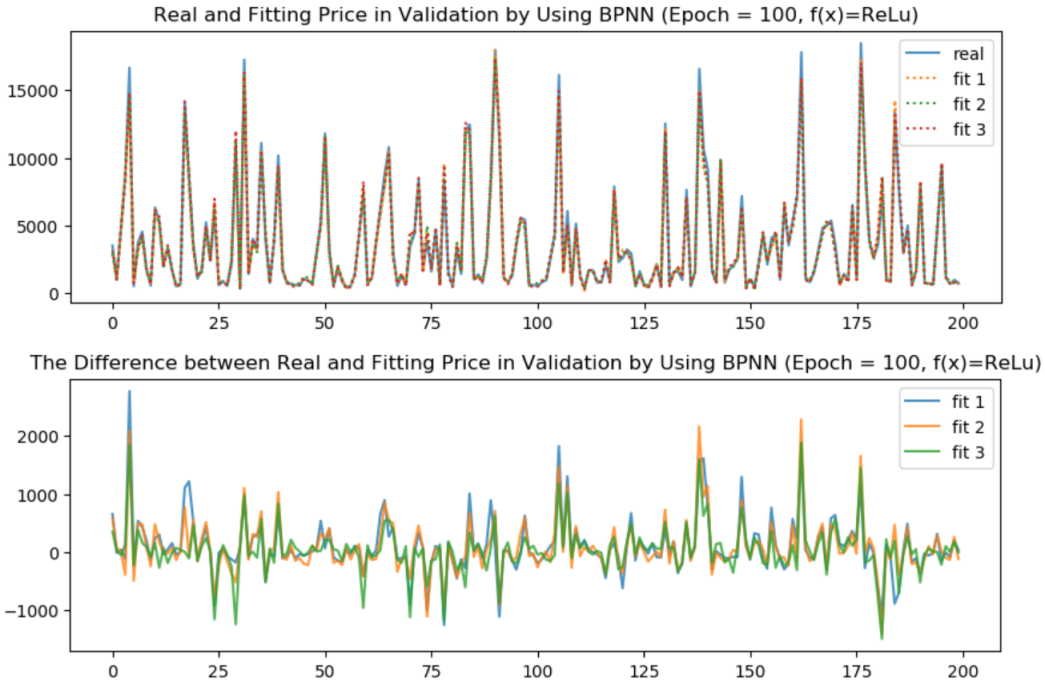


7 結論

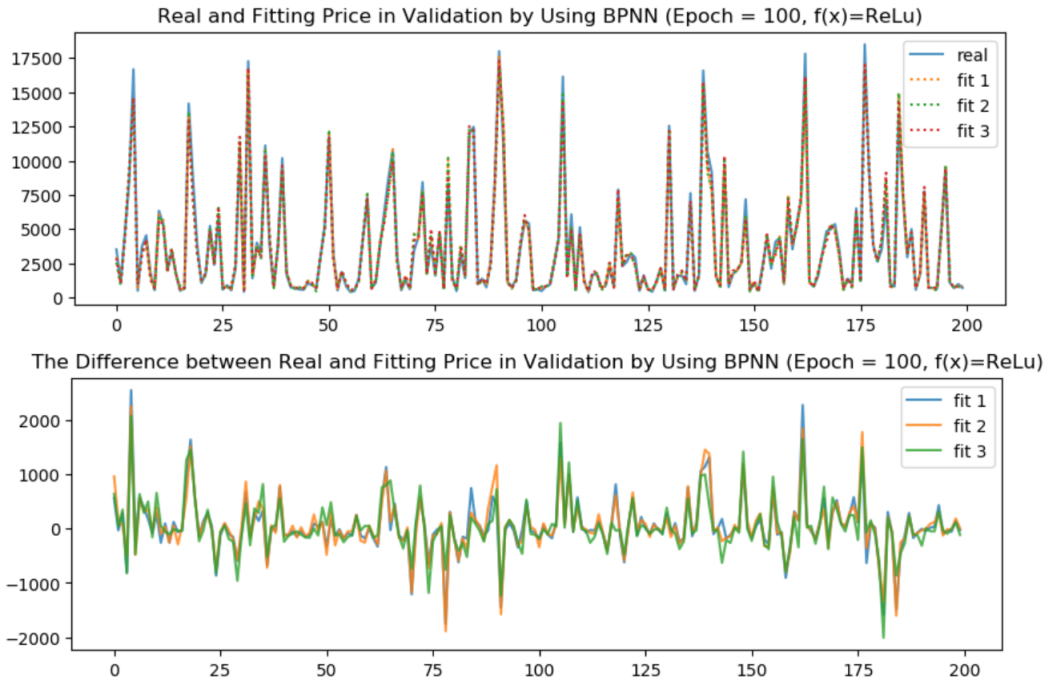
在三種方法的比較中，BPNN 的運算時間最長，單層時間約為 8 到 10 分鐘，雙層約 20 分鐘，RMSE 最小，結果最為準確；RBFNN 運算時間短，單層時間約為 1 分鐘，雙層約 2 分鐘，結果尚可，對於較極端的值容易高估或低估，SVR 計算時間短，結果較差，可嘗試更多參數做訓練。

8 附錄

使用 BPNN，活化函數為 ReLu，一層隱藏層，fit1 fit3 分別為 5、10、20 個神經元，第一張為實際值，第二張為誤差圖。



使用 BPNN，活化函數為 ReLu，兩層隱藏層，fit1 fit3 分別為兩層分別使用 0-3、3-2、3-3 個神經元，第一張為實際值，第二張為誤差圖。



使用 BPNN，活化函數為 Sigmoid，一層隱藏層，fit1 fit3 分別為 5、10、20 個神經元，第一張為實際值，第二張為誤差圖。