國立雲林科技大學資訊管理系

機器學習 – 作業一

Department of Information Management National Yunlin University of Science & Technology Assignment

MNIST資料集和波士頓房價資料集分析

MNIST Data Set and

Boston Housing Price Data Set analysis

M11323039、M11323021、M11323023、M11323041

蔡昆彥、蔣育承、毛淑娟、林承勳

指導老師：許中川 博士

Advisor: Chung-Chian Hsu, Ph.D.

中華民國114年3月

March 2025

**摘要**

本研究探討神經網路在不同類型資料集上的應用，透過 MNIST 手寫數字辨識與波士頓房價預測實驗，分析隱藏層數、神經元數量、激活函數與超參數對模型效能的影響。在 MNIST 資料集中，結果顯示單隱藏層並搭配 ReLU 與 Softmax 激活函數即可達到 98.1% 的準確率，而增加隱藏層數可能導致過擬合（Overfitting），使準確率下降。在波士頓房價預測方面，研究發現較簡單的神經網路架構（單隱藏層 + ReLU & Sigmoid）可獲得穩定的平均絕對誤差（Mean Absolute Error, MAE）、均方誤差（Mean-Square Error, MSE）與均方根誤差（Root-Mean-Square Error, RMSE）表現，且避免運算資源浪費。整體而言，本研究強調不同資料集需要不同的模型配置，並建議根據資料特徵選擇適當的神經網路架構與超參數，以平衡模型效能與計算成本。

關鍵字：ReLU、神經網路、隱藏層、神經元

1. **緒論**
   1. **動機**

本實驗使用神經網路並搭配適當的激活函數研究MNIST資料集、波士頓房價資料集進行研究。

MNIST資料集是現今機器學習中最經典的手寫數字辨識基準，最常用於圖像分類跟深度學習模型的測試。當今已經有許多行業都在使用此技術，像銀行業處理帳戶資料和金額、郵局簽收包裹和醫院手寫病歷等，除了可以減少人工作業的時間也大大的減少錯誤的發生，所以本實驗透過分析MNIST資料集並從中學習最佳的參數，來提高訓練的正確性。

波士頓房價資料集也是機器學習迴歸分析中的經典資料集，常用於評估模型對房價預測的能力。由於現今都市發展快速，對於整個房產交通的規劃越來越完善方便，也因此也伴隨房價高漲的問題，因房價不僅僅是考量房屋的品質，還有考慮到周遭環境。因此本實驗欲透過分析波士頓房價資料集，觀察房價與外在因素之間的關係，以幫助政府政策考量，改善城市的規劃。

* 1. **目的**

本研究目的想以神經網路對MNIST資料集、波士頓房價資料集進行實驗。透過不同資料集的測試，探索適用的激活函數，並透過調整隱藏層層數與神經元數量、模型的超參數，以優化模型表現。最終根據相應的績效指標評估模型效能，從實驗結果中分析並提取有價值的資訊。

1. **方法**
   1. **實驗說明**

本實驗使用神經網路對MNIST資料集、波士頓房價資料集進行實驗。進行各個實驗前，需各自做適當的資料前處理，以獲得乾淨的資料，再開始建立神經網路，通過調整神經網路的隱藏層數、神經元數量、Batch Size、Epoch，其中激活函數的部分，調整隱藏層、輸出層使用ReLU、Softplus、Softmax、Sigmoid交叉比較，迴歸預測因目標為預測數值，而無需要激活函數，直接輸出數值即可，透過上述調整使模型更加優化，最後將調整好的模型對測試集進行績效衡量，對類別預測績效時，使用Precision、Recall及F1-score指標；數值預測績效則使用MAE、MSE與RMSE績效指標。

* 1. **操作說明**

本研究執行環境皆採用Python3.11.11，以Google Colab作為開發工具，神經網路進行研究分析，並使用Pandas、Numpy、Keras等函式庫來讀取資料。於各個資料前處理，利用Pandas套件功能，刪除空值、重複資料、數值標準化、對名目資料做獨熱編碼（One-Hot-Encoding）變成模型所需的型態。

1. **實驗**
2. **資料集**

**資料集名稱：**MNIST資料集

**原始資料筆數：**6000筆

**資料前處理後資料筆數：**6000筆

**表 1**

*MNIST資料集欄位介紹*

|  |  |  |
| --- | --- | --- |
| **欄位** | **屬性** | **內容** |
| 0 | pixel | Continuous |
| 1 | label | Nominal |

**資料集名稱：**波士頓房價資料集

**原始資料筆數：**404筆

**資料前處理後資料筆數：**404筆

**表 2**

*波士頓房價資料集欄位介紹*

|  |  |  |
| --- | --- | --- |
| **欄位** | **屬性** | **內容** |
| 0 | CRIM | Continuous |
| 1 | ZN | Continuous |
| 2 | INDUS | Continuous |
| 3 | CHAS | Continuous |
| 4 | NOX | Continuous |
| 5 | RM | Continuous |
| 6 | AGE | Continuous |
| 7 | DIS | Continuous |
| 8 | RAD | Continuous |
| 9 | TAX | Continuous |
| 10 | PTRATIO | Continuous |
| 11 | B | Continuous |
| 12 | LSTAT | Continuous |
| 13 | MEDV | Continuous |

1. **前置處理**
2. **MNIST資料集**

* **資料前處理**
  + 將資料集每張圖片展平為長度784（28×28）的一維向量，接下來再來將原始數據類型從uint8 轉換成float32，最後透過將所有數值除以255使像素值歸一化到[0, 1]範圍，使不同特徵以同標準表示出，能使神經網路更容易收斂。
  + 將訓練集和測試集做One-Hot Encoding，使原本的手寫數字標籤轉為長度為10的向量，這能避免數值順序影響學習。
  + 為了避免模型產生過擬合，將資料集進行分割分成訓練集80%和測試集20%，如此可以檢視模型是否過度訓練。

**表 3**

*MNIST資料集的圖片數字2資料處理前向量值*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

**表 4**

*MNIST資料集的圖片數字2資料處理後向量值*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

1. **波士頓房價資料集**

* **資料前處理**
  + 首先檢查數據中是否存在重複值或缺失值，並在發現後加以移除。然後對訓練集做正規化，算出訓練集的平均數跟標準差，然後對訓練集減平均數除標準差做Z-Score，做完後才對測試集減平均數除標準差做Z-Score，使最終將數值都壓縮到0到1的範圍內。
  + 為了避免模型產生Overfitting，將資料集進行分割分成訓練集80%和測試集20%，如此可以檢視模型是否過度訓練。

**表 5**

*部分資料處理後的波士頓房價資料集*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 特徵  資料 | **CRIM** | **ZN** | **INDUS** | … | **B** | **LSTAT** | **MEDV** |
| **No.0** | -0.272 | -0.483 | -0.435 | 0.448 | 0.824 | -0.781 |
| **No.1** | -0.403 | 2.988 | -1.332 | 0.431 | -1.328 | 2.161 |
| **No.2** | 0.125 | -0.483 | 1.027 | 0.22 | -1.307 | 2.997 |
| **No. 3** | -0.401 | -0.483 | -0.868 | 0.448 | -0.652 | -0.141 |
| **No. 4** | -0.006 | -0.483 | 1.027 | 0.389 | 0.263 | -0.51 |

1. **實驗設計**
   * 1. **MNIST資料集**

本研究對 MNIST 資料集進行資料前處理，首先將訓練集與測試集進行展平，將原始二維影像轉換為一維向量，以便輸入神經網路模型。接著，由於神經網路訓練時使用浮點數運算，因此先將數據類型轉換為 float32。此外，MNIST 圖像的像素值範圍為0至255，為了進行正規化處理，我們將所有數值除以255，使其壓縮至0到1之間，以提高模型的訓練穩定性。

接下來，對訓練集與測試集的數字標籤進行One-Hot Encoding，將標籤轉換為長度為10的向量，其中對應數字的位置為1，其餘則為0。在正式訓練模型前，為了降低Overfitting風險，先將訓練集分割成訓練集80%和測試集20%，然後進行神經網路訓練。最後，透過調整隱藏層層數、神經元數量、激活函數及超參數來尋找最佳模型，並使用 Precision、Recall、Accuracy、F1-score 等指標進行模型效能評估。

* + 1. **波士頓房價資料集**

本研究對波士頓房價資料集進行資料前處理，首先檢查數據中是否存在重複值或缺失值，並在發現後加以移除。接著，對數據進行Z-Score正規化，將數值轉換為至0到1之間，以確保不同特徵的數值範圍相近，提升模型的穩定性。此外，為了減少Overfitting的風險，將資料集分割成訓練集80%和測試集20%，以評估模型在未見過的數據上是否仍能保持良好表現。

隨後，開始訓練神經網路模型，透過調整隱藏層層數、神經元數量、激活函數及超參數來尋找最佳模型配置。最後，使用MAE、MSE、RMSE進行模型效能評估，以綜合衡量模型的預測準確性與誤差程度。

1. **實驗結果**

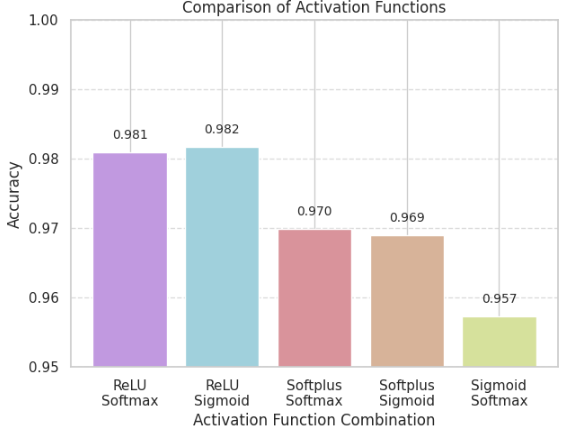
本實驗對各個資料集進行調整，主要研究模型的神經網路架構和設定調整其超參數，分類研究和迴歸研究的績效分別以Accuracy和MAE下去觀察，探討其影響的原因。

1. **MNIST資料集分析**

本研究為了有效調整架構跟參數，所以先固定神經網路的架構，比較調整隱藏層跟輸出層的激活函數的績效，從下**圖 1**和**表 6**，可以觀察出隱藏層使用ReLU以及輸出層使用Softmax這個組合的績效最佳，所以本實驗將激活函數固定為ReLU ＆ Softmax。

**圖 1**

*Comparison of Activation Functions1*



**表 6**

*Comparison of Activation Functions1*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **隱藏層** | ReLU | ReLU | Softplus | Softplus | Sigmoid |
| **輸出層** | Softmax | Sigmoid | Softmax | Sigmoid | Softmax |
| **Accuracy** | | 0.981 | 0.981 | 0.970 | 0.969 | 0.957 |
| **Loss** | | 0.064 | 0.065 | 0.096 | 0.100 | 0.146 |

接著調整神經網路架構，本研究設計了四種實驗，設定的參數值和績效於下方，從實驗結果下方**表 7**可觀察到研究結果顯示準確度皆維持在 0.98 左右，變動影響有限。但是當隱藏層數增加時，訓練時間從 55 秒上升至 367 秒，顯示計算成本大幅提高。此外，5 層隱藏層的 Loss 值反而增加至 0.174，可能出現Overfitting現象。因此，在準確度相近的條件下，較簡單的模型更具效能，能有效降低運算成本並避免Overfitting的風險。

**表 7**

*Comparison of deep and shallow models1*

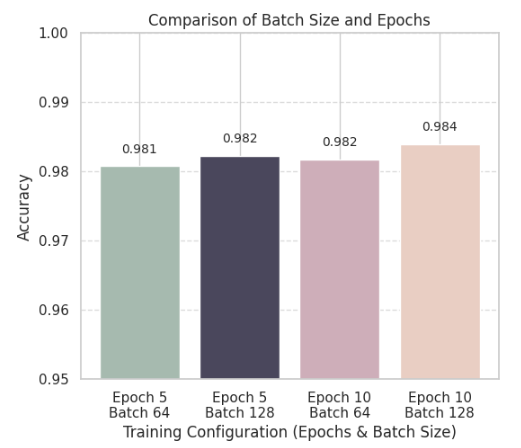
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **隱藏層層數** | 1 | 1 | 3 | 5 |
| **神經元數量** | 1000 | 500 | 1000 | 1000 |
| **Accuracy** | | 0.981 | 0.978 | 0.981 | 0.980 |
| **Loss** | | 0.120 | 0.134 | 0.127 | 0.174 |

最後調整超參數，本研究設計了四種實驗，設定的參數值和績效於下方，從實驗結果下方**圖 2**、**表 8**判斷，各組的模型隨著訓練週期（Epochs）與批次大小（Batch Size）的變化，測試集的準確度皆有小幅提升，但最佳與最差績效僅相差0.001，顯示影響相對有限。

整體而言，增加訓練週期（從 5 到 10）確實帶來準確度提升，而批次大小的影響較不明顯。由於不同組別間的準確度差異極小，本研究認為對此實驗可選擇較簡單的架構與較少的訓練週期，以降低計算成本並避免模型Overfitting。

**圖 2**

*Comparison of Batch Size and Epochs1*



**表 8**

*Comparison of Batch Size and Epochs1*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Epoch** | 5 | 10 | 5 | 10 |
| **Batch** | 128 | 128 | 64 | 64 |
| **Accuracy** | | 0.980 | 0.982 | 0.981 | 0.984 |
| **Loss** | | 0.063 | 0.064 | 0.063 | 0.064 |
| **訓練時間** | | 50s | 92s | 88s | 173s |

1. **波士頓房價資料集**

本研究為了有效調整架構跟參數，所以先固定神經網路的架構，比較調整第一隱藏層跟第二隱藏層的激活函數的績效，從下**表 9**，可以觀察出第一隱藏層使用ReLU以及第二隱藏層使用Sigmoid這個組合的績效最佳，所以本實驗將激活函數固定為ReLU ＆ Sigmoid。

**表 9**

*Comparison of Activation Functions2*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **第一隱藏層** | ReLU | ReLU | Softplus | Softplus | Sigmoid |
| **第二隱藏層** | Softmax | Sigmoid | Softmax | Sigmoid | Softmax |
| **MAE** | | 18.616 | 2.76 | 18.616 | 2.863 | 18.609 |
| **MSE** | | 429.795 | 19.001 | 429.815 | 20.496 | 429.522 |
| **RMSE** | | 20.732 | 4.359 | 20.732 | 4.527 | 20.725 |

**表 10**結果調整隱藏層數與神經元數對 MAE、MSE、RMSE 影響有限，各組間誤差變動幅度不大（MAE 介於 2.555～2.924）。此外，訓練時間維持在 9～12 毫秒，顯示該任務對於網路深度需求較低，較簡單的架構即可獲得穩定表現。因此，為提高計算效率，可選擇較輕量的網路，避免不必要的資源浪費，同時確保預測效果穩定。

**表 10**

*Comparison of deep and shallow models2*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **隱藏層層數** | 2 | 2 | 4 | 4 |
| **神經元數量** | 64 | 128 | 64 | 128 |
| **MAE** | | 2.653 | 2.924 | 2.555 | 2.930 |
| **MSE** | | 17.641 | 18.372 | 18.031 | 20.065 |
| **RMSE** | | 4.200 | 4.286 | 4.246 | 4.479 |

最後調整超參數，本研究設計了四種實驗，設定的參數值和績效於下方**表 11**，從實驗結果判斷，各組的模型隨著訓Epochs與Batch Size的變化，誤差表現有所不同。

整體而言，增加Epochs（從80到160）有助於降低誤差，而較大的Batch Size（64）可能會導致誤差上升。本研究認為在此實驗中，選擇較小的批次大小與較長的訓練週期較能達到良好的模型表現。

**表 11**

*Comparison of Batch Size and Epochs2*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Epoch** | 80 | 80 | 160 | 160 |
| **Batch** | 16 | 64 | 16 | 64 |
| **MAE** | | 2.796 | 3.928 | 2.458 | 2.827 |
| **MSE** | | 19.557 | 35.443 | 15.879 | 18.859 |
| **RMSE** | | 1.672 | 1.981 | 1.567 | 1.681 |

1. **結論**

機器學習在各領域的應用逐漸擴展，從數字識別到房價預測，都需要針對性地建立相應的機器學習模型。本次實驗通過分析兩個不同的資料集：MNIST數字識別和波士頓房價預測，探討了神經網絡模型在這些任務中的表現。

首先，針對MNIST數字識別資料集，實驗顯示，使用一個簡單的模型，僅設置單個隱藏層並設定1000個神經元，就能達到98.1 %的準確率，反而增加到五層時，準確率下降到98%。這是因為MNIST資料集較小，簡單模型已能有效捕捉關鍵特徵。因此，對於這類資料集，無需過度追求複雜的模型結構，過度的複雜化反而可能會引起Overfitting，簡單模型即可達到良好的結果。

接著，對波士頓房價預測資料集進行分析時，結果顯示，使用單隱藏層且神經元數量較少的模型能夠最快收斂，並且預測性能最佳。這可能是因為波士頓房價資料集較小，過於複雜的模型容易導致Overfitting。因此，對於這類中等規模的回歸問題，選擇簡單模型結構能達到較好的效果。

總結來說，不同特徵和規模的資料集需要不同的模型配置。對於小規模的資料集，較簡單的模型結構就能實現出色的性能，無需追求過於複雜的網絡架構。而對於較大規模的資料集，適當增加模型的複雜度有助於提高性能，但仍需注意模型的穩定性。在實際應用中，需要根據具體的任務和資料特徵，謹慎選擇適合的神經網絡模型及其超參數配置，從而達到最佳的效果。

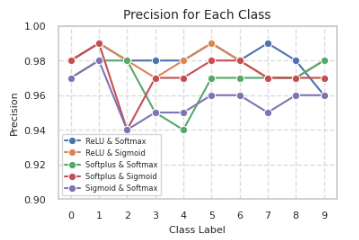
**附錄**

1. **MNIST資料集**

**調整激活函數**

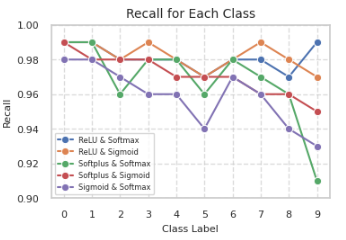
**圖 3**

*Precision for Each Class*

****

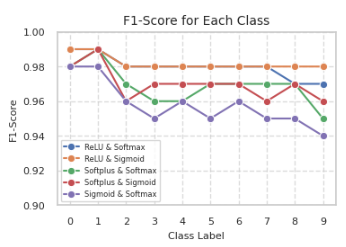
**圖 4**

*Recall for Each Class*

****

**圖 5**

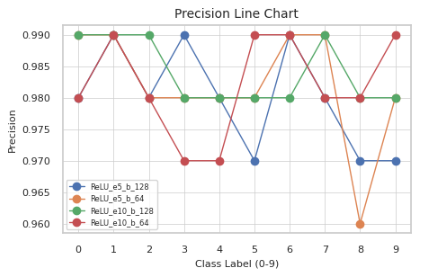
*F1-Score for Each Class*

****

**調整Epochs及Batch\_Size**

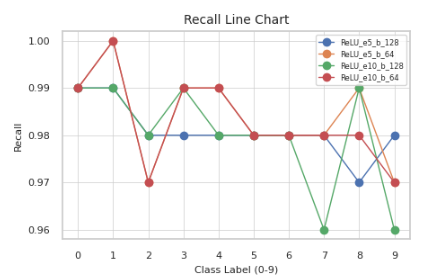
**圖 6**

*Precision Line Chart*



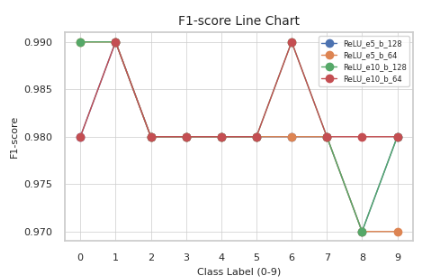
**圖 7**

*Recall Line Chart*



**圖 8**

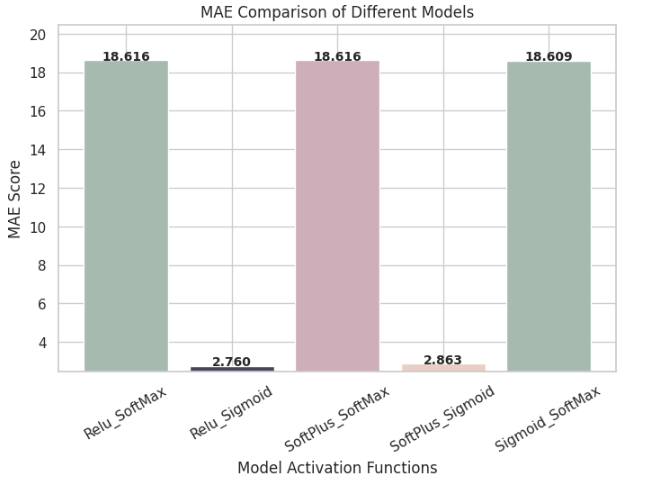
*F1-score Line Chart*



1. **房價**

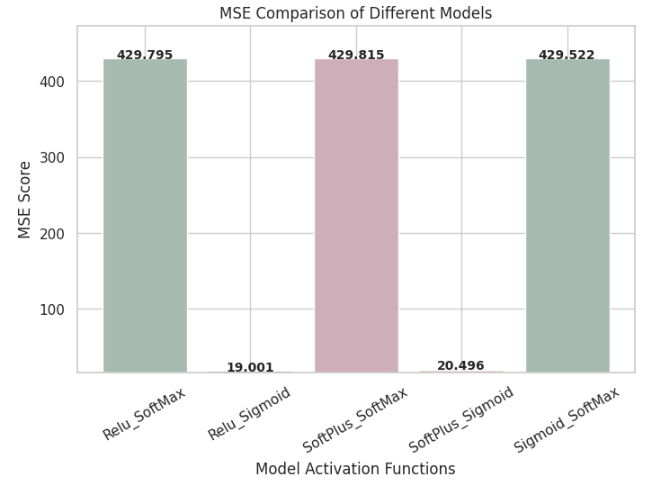
**圖 9**

*MAE Comparison of Different Models*



**圖 10**

*MSE Comparison of Different Models*



**圖 11**

*RMSE Comparison of Different Models*

