

倒傳遞神經網路 演算法實作

學號：7105029031

姓名：賴念翔

說明

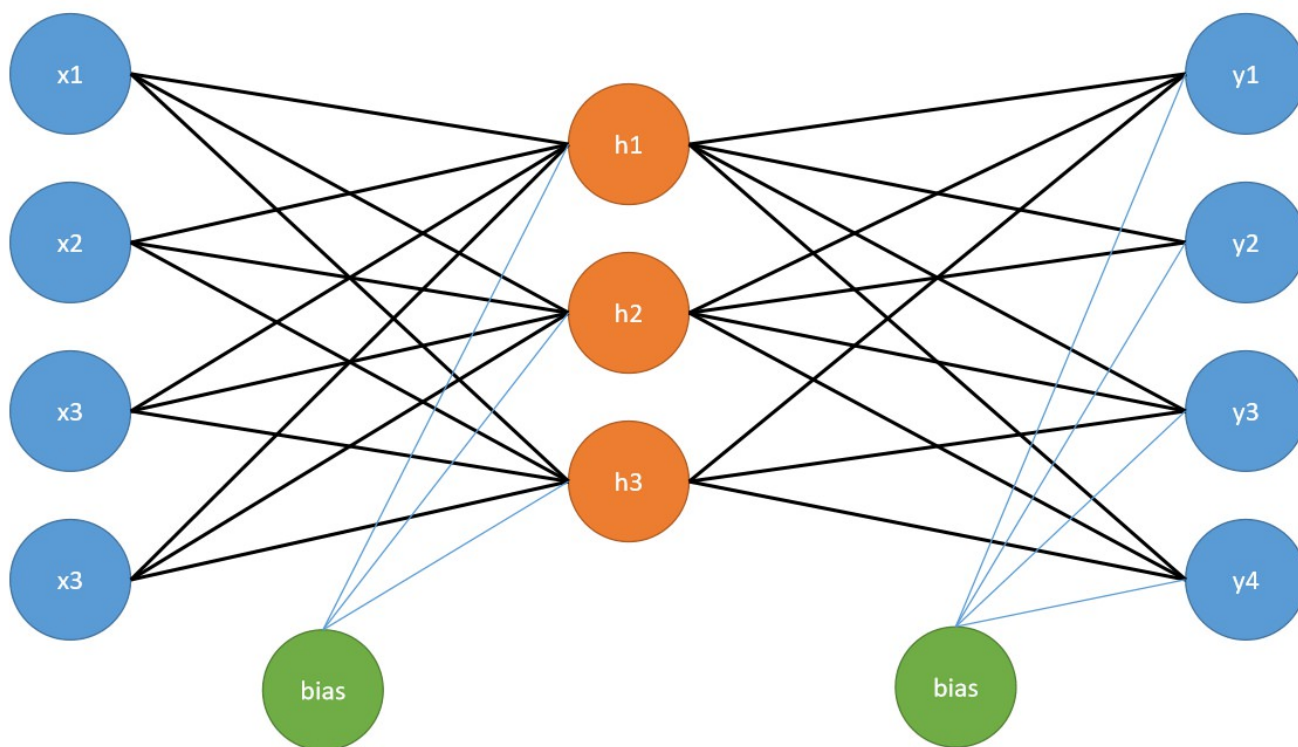
利用 Python 實作 Machine Learning 演算法 - 倒傳遞神經網路（BPN）

資料集（Dataset）

使用 UCI 的 Iris（鳶尾花）資料集

網路說明

網路架構圖



網路架構（Neural Network Architecture）

- 輸入層（Input Layer）
 - 總共 4 個節點
 - 使用 Iris 的四個特徵值作為輸入層，為一個（150 * 3）的矩陣
- 隱藏層（Hidden Layer）

- 總共 3 個節點
- 設定為一個 $(150 * 3)$ 的矩陣
- 輸出層 (Output Layer)
 - 總共 3 個節點
 - 使用 Iris 的輸出為三類，為一個 $(150 * 4)$ 的矩陣

權重 (Neural Network Weight)

- 輸入層至隱藏層 (Input Layer to Hidden Layer)
 - 為一個 $((4 + 1) * 3)$ 的矩陣
 - 加上一個 bias
- 隱藏層至輸出層 (Hidden Layer to Output Layer)
 - 為一個 $((3 + 1) * 3)$ 的矩陣
 - 加上一個 bias

輸出結果

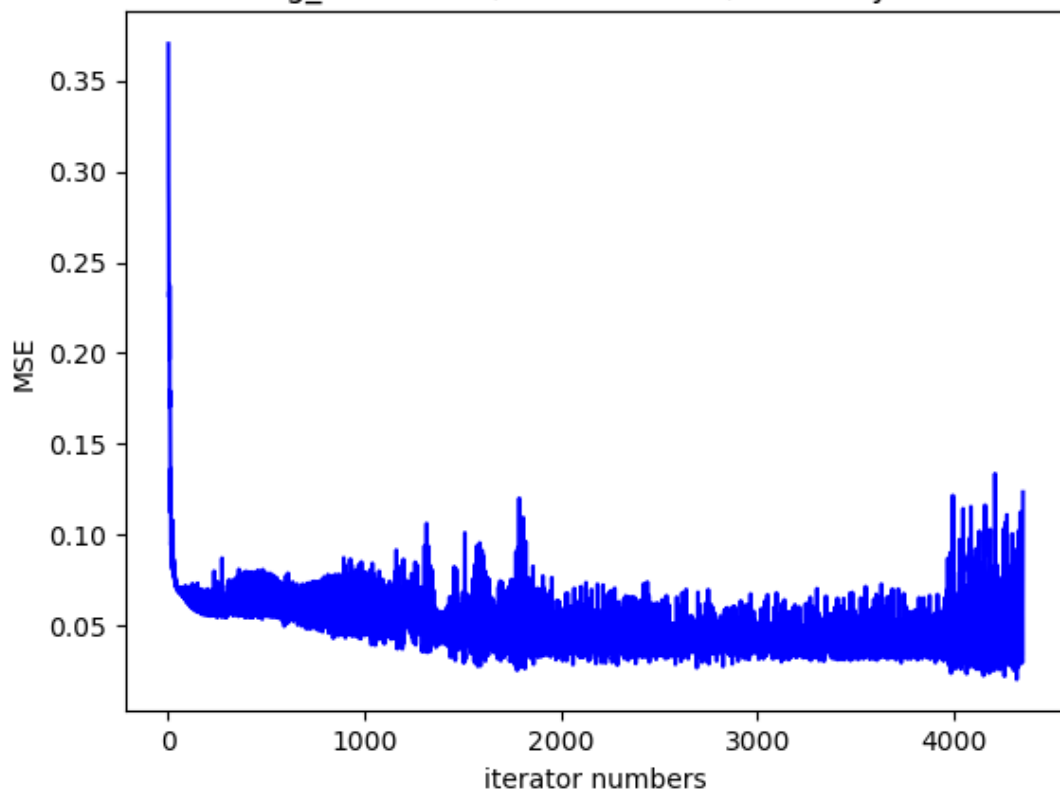
訓練走勢圖

收斂條件

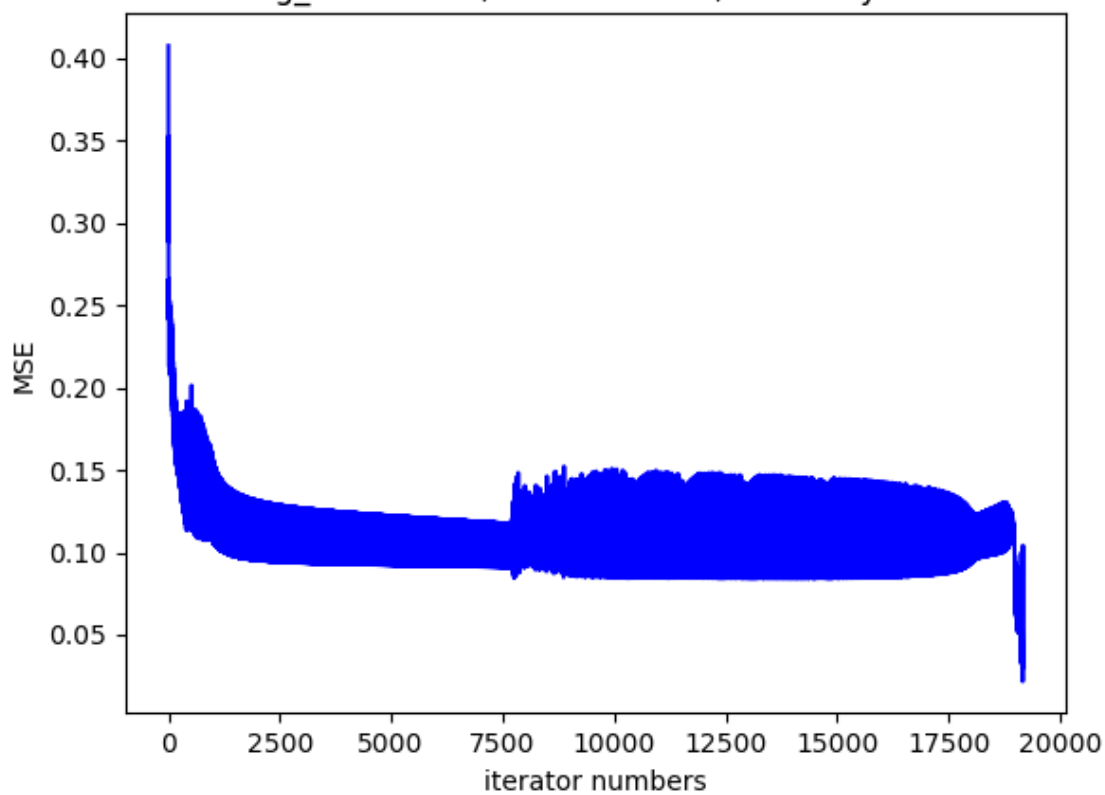
1. 迭代次數達 20000 次
2. 正確率達 98 % 以上 (包含 98 %)

	學習速率	迭代次數	MSE	正確率
1	0.5	47	0.114	96.66%
2	0.1	19170	0.03	100%
3	0.08	9559	0.016	100%
4	0.05	10210	0.016	100%
5	0.03	1578	0.024	100%
6	0.01	4980	0.024	100%

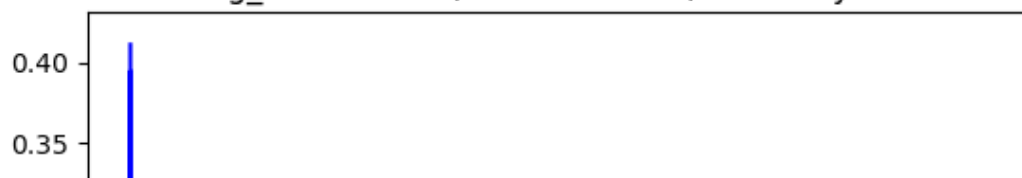
learning_rate = 0.5 , MSE = 0.123 , Accuracy = 97 %

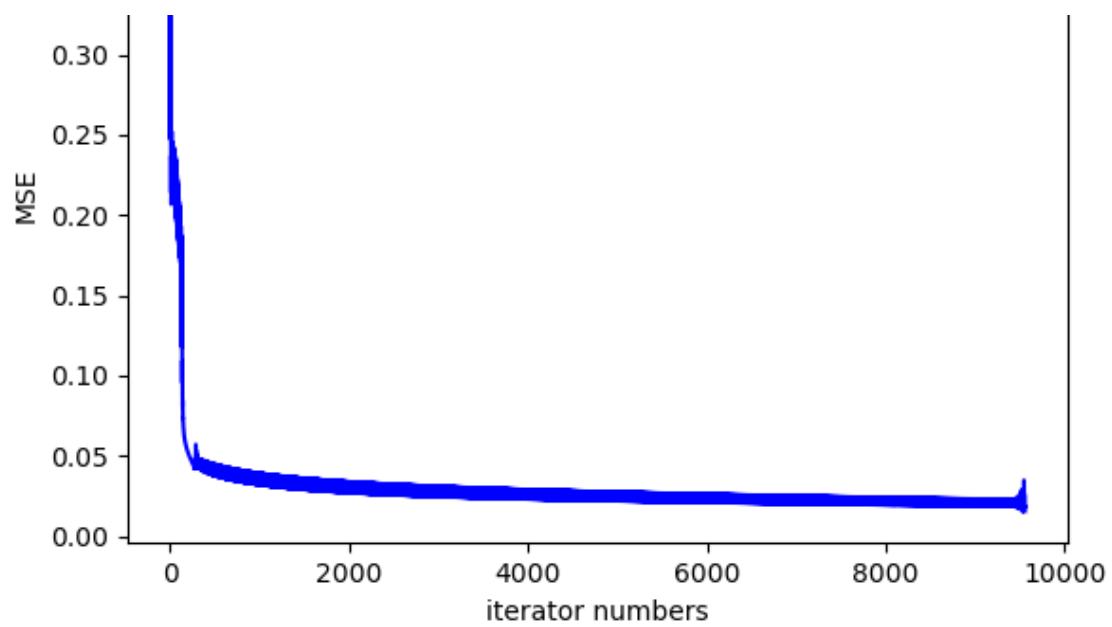


learning_rate = 0.1 , MSE = 0.030 , Accuracy = 100 %

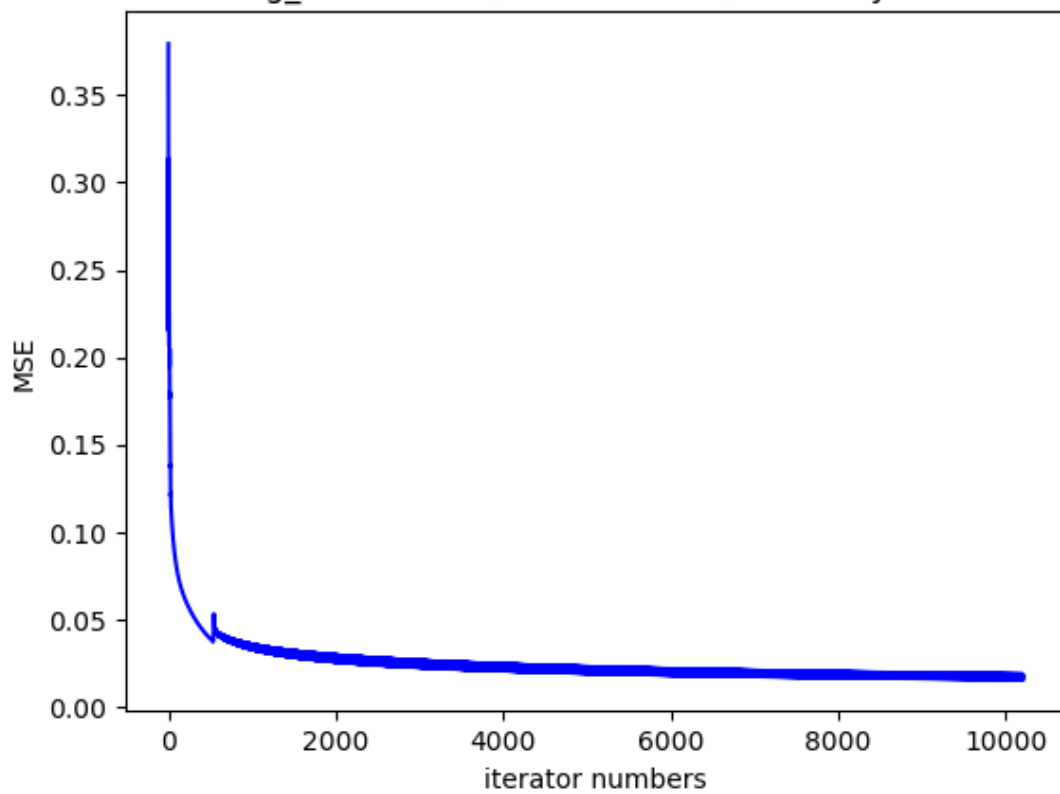


learning_rate = 0.08 , MSE = 0.016 , Accuracy = 100 %

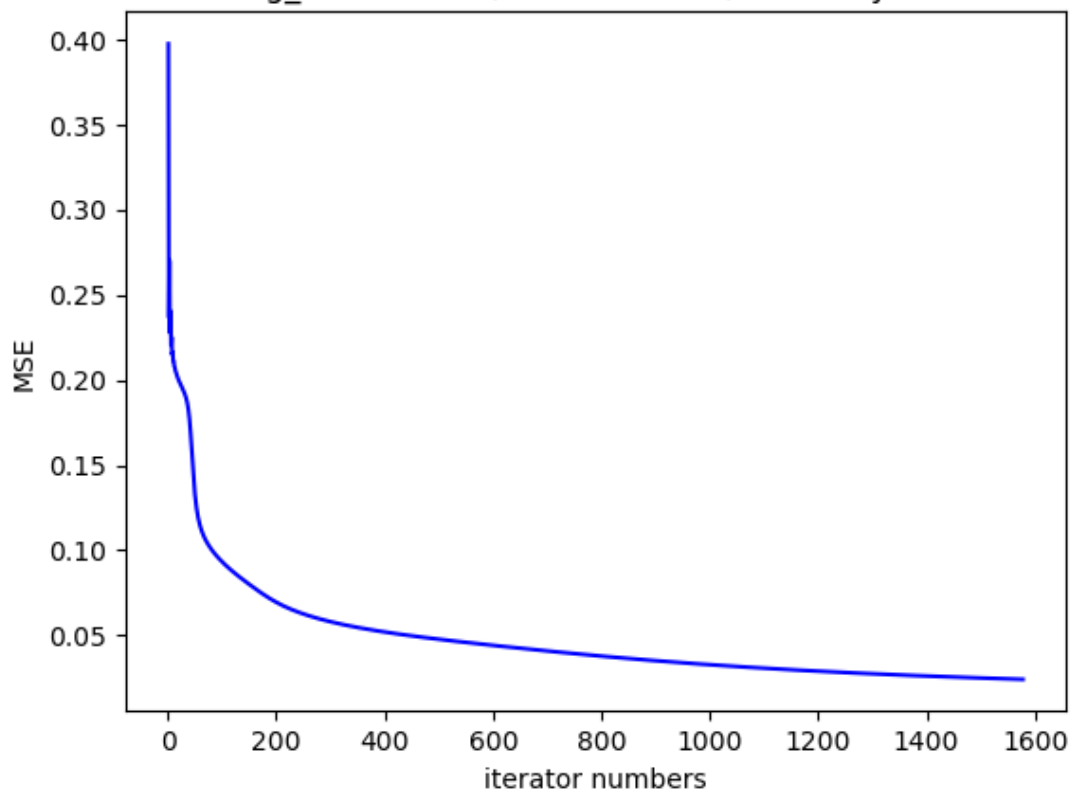


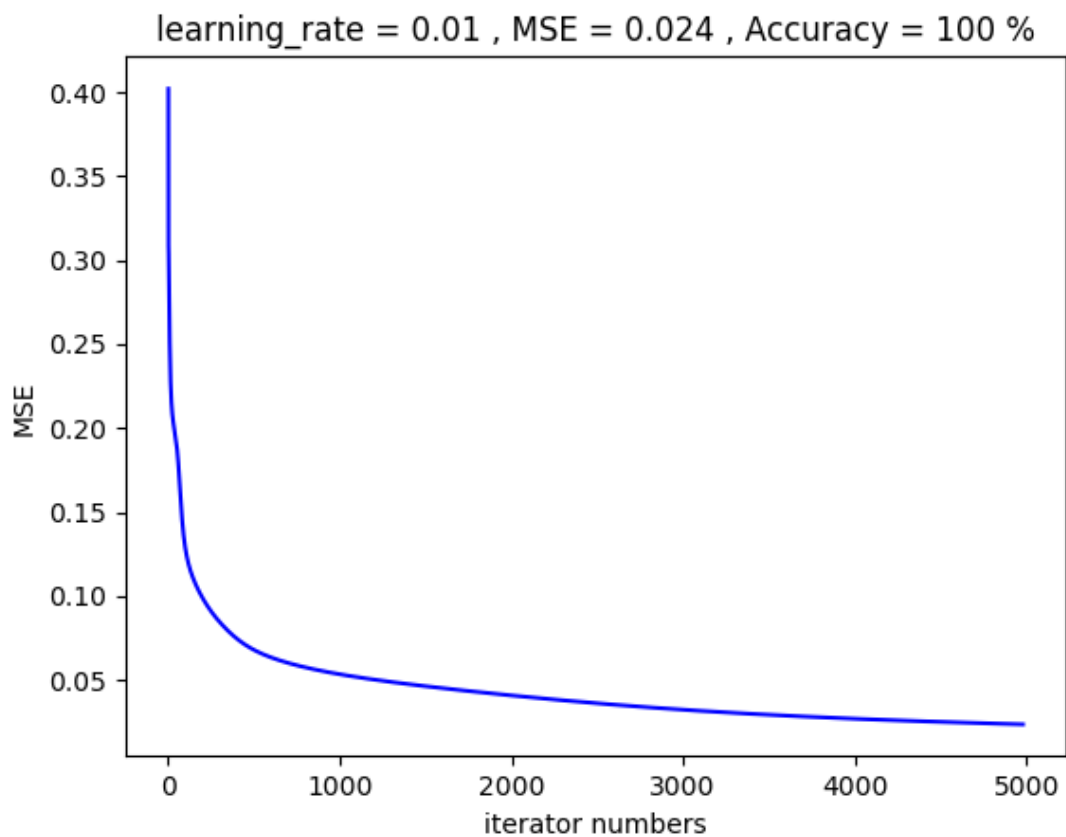


learning_rate = 0.05 , MSE = 0.016 , Accuracy = 100 %



learning_rate = 0.03 , MSE = 0.024 , Accuracy = 100 %





安裝所需套件

1. 開啟 **terminal** 並且進入專案所在的資料夾
2. 輸入下列指令來安裝所需套件

```
pip install -r requirement.txt
```

使用

1. 開啟 **terminal** 並且進入專案所在的資料夾
2. 輸入下列指令來執行程式

```
python main.py
```