

# 人工智慧-hw01 實作類神經網路

40823117L 資工系 方國丞

1. 首先，你要說明你針對這個成績計算的Data set，輸入是甚麼?有幾個?輸出是什麼?有幾個?你想用多少組當訓練資料?用多少組當測試資料?

注意:請詳細說明你所使用之機器軟、硬體規格、作業系統等相關資訊以及你為何選擇這樣的規格。另外請提供你的連絡電話，以便不時之需。

輸入為: 抽問分數, 作業平均, 測驗平均

因為每次小考分數最後都以靜計算至平均分數了，所以拿三個獨立的變項來做輸入

輸出為: 總成績

訓練資料: 10筆

訓練資料: ~80%

測試資料: 4筆

測試資料: ~20%

環境

兩個環境都測試過

OS: macOS 13.0  
CPU/GPU: Apple M2

以及

OS: Windows 10  
CPU: AMD Ryzen 5 3600  
GPU: RTX3080

手機號碼: 0978271809

2. 請先用個一個神經元，有activation function，寫python程式，用來訓練你這個資料集，看花了不同的時間: 訓練資料的mean square error的結果為何? 測試資料的mean square error結果為何?也請說明如何執行你的程式。

測試資料總成績: 76 73 44 57

使用activation functions: Sigmoid 並且 learning\_rate設置為0.1

- 訓練100000次
  - 測試資料預測結果: 84 80 74 67
  - 訓練資料的mean square error: 19.7
  - 測試資料的mean square error: 278.25
- 訓練10000次
  - 測試資料預測結果: 83 82 70 74

- 訓練資料的mean square error: 24.9
  - 測試資料的mean square error: 273.75
- 訓練1000次
  - 測試資料預測結果: 83 83 69 82
  - 訓練資料的mean square error: 31.3
  - 測試資料的mean square error: 349.75
- 訓練100次
  - 測試資料預測結果: 83 83 69 83
  - 訓練資料的mean square error: 31.0
  - 測試資料的mean square error: 362.5

訓練100000次似乎出現過擬和的狀況

執行程式: `python hw0102.py`

### 3. 請使用不同的activation functions來做上一項任務，同樣地做說明，並比較有沒有比較好的activation functions?

測試資料總成績: 76 73 44 57

**使用activation functions: Tanh 並且 learning\_rate設置為0.1**

- 訓練100000次
  - 測試資料預測結果: 76 72 37 39
  - 訓練資料的mean square error: 3.7
  - 測試資料的mean square error: 93.5
- 訓練10000次
  - 測試資料預測結果: 76 72 37 39
  - 訓練資料的mean square error: 3.7
  - 測試資料的mean square error: 93.5
- 訓練1000次
  - 測試資料預測結果: 75 73 33 55
  - 訓練資料的mean square error: 8.2
  - 測試資料的mean square error: 31.5
- 訓練100次
  - 測試資料預測結果: 77 81 47 87
  - 訓練資料的mean square error: 34.4
  - 測試資料的mean square error: 243.5

使用Tanh作為activation function，訓練資料的mean square error以及測試資料的mean square error明顯較Sigmoid低許多，尤其是訓練次數較多的情況下，但Tanh再訓練次數超過10000次後迭代速度開始變慢，最後權重也無再更新，出現梯度消失的狀況

執行程式: `python hw0103.py`

#### 4. 請設法用兩個神經元(如講義上的，有**bias**，有**activation function**，來建構類神經網路。請推導**backproagation**公式出來。

1. 因為有兩顆神經元，因此會有兩個誤差，需要先計算輸出層的誤差，再計算隱藏層(1層)的誤差

```
# 先將訓練的總成績資料轉置成10*1的陣列再減去預測的總成績資料
output_error = y_train.reshape((-1,1)) - output_layer_output
# 將output_error乘上w2的轉置矩陣，再乘上sigmoid函數對output_layer_input的導數，最後再與w2(輸出層的權重) 做矩陣乘積
hidden_error = np.dot(output_error * output_layer_output * (1 -
output_layer_output), w2.T)
```

#### 2. 反向傳播

```
# weight使用反向傳播公式 新w = 舊w + input * (與下一層的誤差 * activation function對輸出結果的導數) * learning_rate
# bias使用反向傳播公式 新b = 舊b + sum(與下一層的誤差 * activation function對輸出結果的導數) * learning_rate
# 使用sum因為為bias在所有樣本中都是相同的，所以只需要對所有樣本的梯度進行求和
w2 += learning_rate * np.dot(hidden_layer_output.T, output_error *
output_layer_output * (1 - output_layer_output))
b2 += learning_rate * np.sum(output_error * output_layer_output * (1 -
output_layer_output), axis=0, keepdims=True)
w1 += learning_rate * np.dot(X_train.T, hidden_error * hidden_layer_output * (1 -
hidden_layer_output))
b1 += learning_rate * np.sum(hidden_error * hidden_layer_output * (1 -
hidden_layer_output), axis=0, keepdims=True)
```

#### 5. 將上一項結論，寫出python程式，來訓練你這個資料集，看花了不同的時間:訓練資料的mean square error結果為何?測試資料的mean square error結果為何?

測試資料總成績: 76 73 44 57

**使用activation functions: Sigmoid 並且 learning\_rate設置為0.1**

- 訓練100000次
  - 測試資料預測結果: 76 72 44 51
  - 訓練資料的mean square error: 0.6
  - 測試資料的mean square error: 9.25
- 訓練10000次
  - 測試資料預測結果: 79 78 53 68
  - 訓練資料的mean square error: 13.2
  - 測試資料的mean square error: 59.0
- 訓練1000次
  - 測試資料預測結果: 86 86 85 87
  - 訓練資料的mean square error: 53.6
  - 測試資料的mean square error: 712.5

- 訓練100次
  - 測試資料預測結果: 86 86 86 87
  - 訓練資料的mean square error: 57.2
  - 測試資料的mean square error: 733.25

執行程式: `python hw0105.py`

**6. 將上一項結論，使用不同的activation functions，寫出python程式來做上一項任務，同樣地做說明，並比較有沒有比較好的activation functions?**

測試資料總成績: 76 73 44 57

**使用activation functions: Tanh 並且 learning\_rate設置為0.1**

- 訓練100000次
  - 測試資料預測結果: 77 73 50 60
  - 訓練資料的mean square error: 0.0
  - 測試資料的mean square error: 11.5
- 訓練10000次
  - 測試資料預測結果: 77 74 51 59
  - 訓練資料的mean square error: 0.8
  - 測試資料的mean square error: 13.75
- 訓練1000次
  - 測試資料預測結果: 76 75 50 67
  - 訓練資料的mean square error: 5.2
  - 測試資料的mean square error: 35.0
- 訓練100次
  - 測試資料預測結果: 90 91 81 96
  - 訓練資料的mean square error: 84.0
  - 測試資料的mean square error: 852.5

與之前使用一顆nureon一樣，梯度下降的比sigmoid還快，也沒有出現與一顆nureon時出現梯度消失的問題，甚至到100000次時已經能準確預無誤差的預測訓練資料了

執行程式: `python hw0106.py`

**7. 請設法用三個神經元(如講義上的，有bias，有activation function)，來建構類神經網路。請推導backpropagation公式出來。**

將原本的2層神經元改成3層，需要在初始化時新增一組weight和bias，並在前向傳播和反向傳播加入第二層和第三層的運算

**8. 將上一項結論，寫出python程式，來訓練你這個資料集，看花了不同的時間:訓練資料的mean square error結果為何?測試資料的mean square error結果為何?**

測試資料總成績: 76 73 44 57

**使用activation functions: Sigmoid 並且 learning\_rate設置為0.1**

- 訓練100000次
  - 測試資料預測結果: 77 72 43 49
  - 訓練資料的mean square error: 0.9
  - 測試資料的mean square error: 16.5
- 訓練10000次
  - 測試資料預測結果: 86 86 86 86
  - 訓練資料的mean square error: 55.3
  - 測試資料的mean square error: 718.5
- 訓練1000次
  - 測試資料預測結果: 86 86 87 87
  - 訓練資料的mean square error: 57.2
  - 測試資料的mean square error: 754.5
- 訓練100次
  - 測試資料預測結果: 86 86 87 87
  - 訓練資料的mean square error: 57.2
  - 測試資料的mean square error: 754.5

多了一顆Nureon，訓練次數少時梯度下降將很不明顯，但多的時候下降很快，但似乎沒有2顆Nueron的還好，可能訓練次數還不夠多

執行程式: `python hw0108.py`

**9.自由申論及發揮:ChatGPT這個聊天機器人程式，請你試著用看看，請記錄問答記錄。A.你問它十個問題，它回答讓你滿意的有多少?B.你有辦法讓它都答不出正確的答案嗎?C.你問它最近的新聞事件，它會亂答嗎?有人說可用它來寫作業，請你將最近老師們出的作業給它做做看，有讓你驚嘆嗎?有人說可用它來算數學題目，它能解決多困難的數學題目?**

- A. 取決與問題的取向，像是如果問他邏輯與數學的問題，他的回答可能就不太好，還有像是歷史的問題，他可能會給出錯誤的回答，但如果是創意的問題，他給出的答案是很讓人驚艷，我現在大部分都會問他翻譯以及一些寫作的問題，滿意度可以到90分，翻譯正確率可以說是100%，但數學、歷史這種有固定答案的，可能只有50%
- B. 可以，你問他2021之後發生的事情，大概率都是不正確的，或是像是一些不屬於英文文化圈的迷因、文化、歷史，他都很大機率會答錯，我想原因是因為他的訓練資料98%都是使用英文資料，其他語言他會說，但是準確率就不太高
- C. 一定亂答，畢竟他現在無法連網，但如果使用Bing的Bing Chat就可以得到正確答案，因為他就是可以連網的ChatGPT
- D. 可以拿來寫創作的作業，像是作文、書面報告、翻譯那類，另外其實他的程式也寫的不錯，但需要有一些先驗知識，你必須debug後才能用，因為他會寫出1個80%對的程式，如果你什麼都不懂，就一定運行不起來

**10.請說明你做此作業所碰到的一些狀況及困難。**

1. weight與bias初始化只能靠隨機的？對這部分還是有點不太確定，因為查到像是Relu似乎可以較公式化的去初始化
2. 中間再算誤差、反向傳播遇到比較多困難，數學很重要
3. 不確定輸入這樣設定是不是最好，也許全部資料丟進去會有更好的效果？

11.請列出你的參考文獻(含網站)來源，並請說明參考了那些部份用結果於作業中。

- <https://zhuanlan.zhihu.com/p/71892752>
- <https://numpy.org/doc/stable/reference/index.html>
- <https://www.twblogs.net/a/5ef361b70cb8aa77788364a7>
- <https://chat.openai.com/chat> ChatGPT for Question 9