

# 類神經網路作業二

資工三乙 406262319 黃育皓

## 統計結果

每個資料皆測試10次，此為平均結果

- 測試總時間56300s
  - min: 1657s
  - max: 10466s
- 終止條件設定為RMSE<0.15 或是 到達epoch上限100000

Epoch	NeuronNum 5	NeuronNum 10	NeuronNum 15	NeuronNum 30
LearningRate 1.0	431.5	606.6	975.2	2833.5
LearningRate 0.5	70205.8	1139.4	2018.7	3931.3
LearningRate 0.1	28778.6	2045.7	3315.7	46890.8

TrainAccuracies	NeuronNum 5	NeuronNum 10	NeuronNum 15	NeuronNum 30
LearningRate 1.0	98.14%	98.3%	98.3%	98.3%
LearningRate 0.5	93.972%	98.3%	98.22%	98.3%
LearningRate 0.1	98.139%	98.3%	98.3%	85.812%

TestAccuracies	NeuronNum 5	NeuronNum 10	NeuronNum 15	NeuronNum 30
LearningRate 1.0	97.28%	96.6%	96.6%	97.28%
LearningRate 0.5	93.95%	96.94%	96.6%	97.96%
LearningRate 0.1	96.94%	96.6%	96.6%	84.97%

## 紀錄解析

- 由於每次init weight皆為random出來的，紀錄10次以觀察平均結果
- 檔案Record中紀錄每次每個條件下的error number
- 在 NeuronNumber=5 , LearningRate=0.5 這筆資料中
  - 發現大部分的結果都會將error卡在 0.4 ~ 0.5 之間
  - 表示其已經找到local min且這個LRate並不能使其跳脫去尋找更小的誤差
  - 也導致其train出來的神經網路其結果比其他略低一些
- 隨著NeuronNum的數量上升，其所需的Epoch數量也會跟著上升
- 隨著LearningRate的下降，其所需的Epoch數量上升
  - LearningRate=0.1 時可能會因為下降得太慢導致跳脫不了一個local min
    - Record1中 NeuronNum=30 , LearningRate=0.1 就卡在 0.7
- 以最後所有的準確率來看， NeuronNum=30 , LearningRate=0.5 是測試的最好的一筆資料
  - 而NeuronNum=30時，在Record檔案中可以觀察到
    - 其error數一開始大都停在 1.2 左右
    - 過一段時間後會開始快速下降
    - 找到的local min約略有 1.2 , 1.0 , 0.6 , 0.16
    - 在到達 0.16 之後下降速度變慢但依舊慢慢接近0.15並低於
  - LearningRate 為 0.5 是平均起來準確率最高的一個

## README

- neuron.py 為整個神經網路架構的class

- `Backpropagation.py` 為主要執行程式
  - 在第88行可進行更改
  - `main(NeuronNum, LearningRate)` 可進行測試並記錄在Record.txt與Output.out
- 第90行開始為總測試時用的，跑出來的檔案為 `Record1.txt ~ Record10.txt` 與 `Output1.out ~ Output10.out`
  - 將其分別放置於Record與Output資料夾中
  - 費時15hr