鄭筠蓉 108502545 類神經網路第三次作業-Hopfield

一、程式執行說明

點擊 dist 資料夾裡面的 exe 執行檔並且輸入訓練資料集及測試資料集檔案位置即可執行。(GUI 就是小黑窗)

二、程式簡介-Hopfield

read_data：讀訓練及測試資料

把資料上空白轉換成-1、1 轉換成 1，因為一筆資料是 9*12，所以當他讀入 12 行之後，就把他存起來，並且計算 P 值(維度)

```
# 讀檔


def read_data(address):
    Read = open(address)
    data = Read.readlines()
    DATAs = []
    input = []
    count = 0
    P = 0
    for line in data:
        if line == '\n':
            continue

        for i in range(0, len(line)):
            if line[i] == ' ':
                input.append(-1.)
            if line[i] == '1':
                input.append(1.)
        count += 1

        if count == 12:
            P = len(input)
            input_arr = np.array([input])
            DATAs.append(input_arr)
            count = 0
            input.clear()

    return P, DATAs
```

cal_W：計算 W 陣列

N 代表有幾筆 input 資料。計算 train data 和他的 transpose 的內積 (108*108)，共三筆，再扣掉(N/P)*I

```
# 計算 W


def cal_W():
    # default:'C:\\Users\\angela_cheng\\Downloads\\Hopfield_dataset\\Basic_Training.txt'
    Add1 = input('training data address:')
    P, training_data = read_data(Add1)
    N = len(training_data)
    res = np.zeros((P, P), dtype=float)
    I = np.identity(P)

    for i in range(0, N):
        tmp = np.transpose(training_data[i]).dot(training_data[i])
        res = res + tmp

    W = (res*1-I*N)/P

    return W
```

cal_theta：計算閥值

把 W 的每個 row 取出來，算每個 row 總和並存起來

```python
# 計算閾值


def cal_theta():
    W = cal_W()
    theta = []
    for i in range(0, W.shape[0]):
        element = 0
        for j in range(0, W.shape[0]):
            element += W[i][j]

        theta.append(element)

    return W, theta
```

sign：計算輸出

uj> θ j 回傳 1，uj< θ j 回傳-1，uj= θ j 維持原狀態

```python
# sign function


def sign(num, orign):
    if num > 0:
        return 1
    elif num < 0:
        return -1
    else:
        return orign
```

train：會去 call 前面的所有 function

```python
def train():
    W, theta = cal_theta()
    return W, theta
```

draw：把訓練出的結果用比較好看的方式畫出來

```python
# 畫結果

def draw(arr):
    arr = arr.flatten('C')
    L = arr.tolist()
    for i in range(0, len(L)):
        if L[i] == 1.0:
            print('■', end='')
        if L[i] == -1.0:
            print('□', end='')
        if i % 9 == 8:
            print('\n', end='')
```

test：非同步的更新

把 input 和 θ (W 的第 n 個 row)做同一個位置的相乘，再扣掉第 n 個 theta，把它丟到 sign function 裡面。

當 108 個 element 都 run 過一次，就把上一個狀態的 input 和更新後的 input 做比較，如果一樣就可以終止訓練，如果不一樣就再 loop 一次。

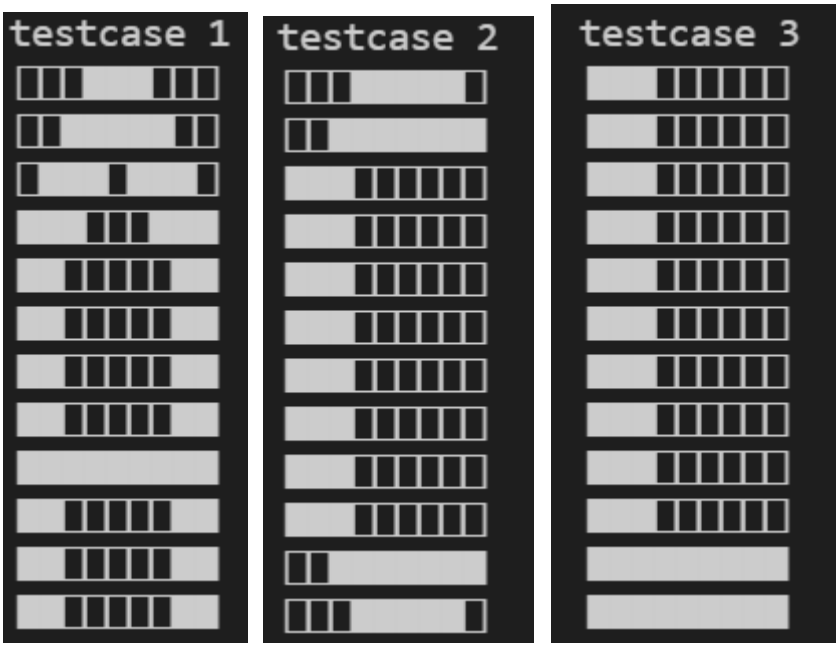
```python
# 測試

def test():
    # default:'C:\\Users\\angela_cheng\\Downloads\\Hopfield_dataset\\Basic_Testing.txt'
    Add2 = input('testing data address:')
    P, testing_data = read_data(Add2)
    W, get_theta = train()
    theta = np.array([get_theta])

    before = 0
    after = 0
    for test_index in range(0, len(testing_data)):
        stop = False
        while stop == False:
            cal = 0
            before = testing_data[test_index]
            for i in range(0, testing_data[0].shape[1]):
                re_row = W[i].reshape(-1, 108)
                tmp = np.multiply(re_row, testing_data[test_index])
                orign = testing_data[test_index][0][i]
                for j in range(0, testing_data[0].shape[1]):
                    cal += tmp[0][j]
                res = sign(cal-theta[0][i], orign)
                testing_data[test_index][0][i] = res
                cal = 0
            after = testing_data[test_index]
            stop = (before == after).all()
        print('testcase', test_index+1)
        draw(after)
```

三、實驗結果(所有資料集都須有實驗結果集說明)

根據觀察，testing data 應該是 training data 的殘缺版，想測試最後是否有成功回想，以實驗結果的截圖來看，是有的！



四、實驗結果分析及討論

- 如果類神經元輸出的更新是採用非同步模式，則網路必定會收斂至某一穩定狀態(他最後一定會收斂)
- 利用能量函數的局部極小特性來儲存資料
- 離散 Hopfield 網路的記憶容量有其上限，若類神經元的數目是 p，在記憶提取有 99% 正確率的情況下，則可儲存的資料筆數不會超過 $\frac{p}{4\ln p}$
- 實驗結果看起來蠻好的，都有進入正確的局部極小值