1. Introduction

McCulloch & Pitts 於 1943 年發表的論文(“A logical calculus of the ideas immanent in nervous activity”)中設計了一個簡單的類神經數學模型，該模型包含了輸入連結(input)、偏離權重(weight)、輸入函數、激勵函數(e.g., sigmoid) 、輸出、輸出連結 (output) ，當輸入的線性組合超過一定臨界值，他會激發 (啟動的概念) 。類神經網絡正是連結所有蒐集的單元一起，至於網絡的特性，決定了他的拓樸性質跟神經元的特性。而從 1943 年之後，很多學者開發許多更精細與實際的模型，用於類比人腦中的類神經和更大的系統，另一方面，人工智慧跟統計學研究相關人員開始對類神經網絡更抽象的屬性感興趣。類神經網絡結構是由單元構成的，他們透過有向連結連接在一起，從單元 a 到單元 b 的連結作用是把激勵 X 從 a 傳到 b，每條連結還有一個數值的權重 w 與之相關聯，該權重決定了連接的強度和符號，如同一個線性回歸模型一樣。如果今天是做一個多層的網路學習(本實驗)，首先需要思考網路需要以向量的函數建構，而非一個純量，設計主要的複雜問題在於網路的隱層的附加，雖然輸出層的誤差很清楚，但因為訓練資料無法告訴我們隱節點應該具有什麼價值，所以我們才需要從輸出層向隱層逆向傳播(backpropagation)誤差(由 vapnik 提出)，逆向

傳播過程會從總體誤差梯度的導數(微分)中求得。

2. Experiment setups

A. Sigmoid functions

為了取得 hidden layer 的最終 value，這邊加入常用的 sigmoid 作為激勵函數，讓其輸出從 0-1 的值，讓我們更輕鬆的改變權重:

$$f(x) = \frac{1}{1 + e^{-x}}$$

```
def sigmoid(self, x):
    # 激勵函數
    return 1/(1+np.exp(-x))
```

B. Neural network

實驗設計 NN 框架為兩層 hidden layer、input size =2、output size =1、hidden size = 3，其餘設置如下:

```python
#2 layer Neural Network
class Neural_Network(object):
    def __init__(self):
        #各layer大小設定
        self.inputSize = 2
        self.outputSize = 1
        self.hiddenSize = 3

        #初始權重
        self.W1 = np.random.randn(self.inputSize, self.hiddenSize)
        self.W2 = np.random.randn(self.hiddenSize, self.outputSize)

    def forward(self, x):

        self.z1 = np.dot(x, self.W1)
        self.z2 = self.sigmoid(self.z1)
        self.z = np.dot(self.z2, self.W2)
        output = self.sigmoid(self.z)
        return output
```

C. Backpropagation

逆向傳播設置如下:

```python
def backward(self, x, y, output):

    self.error_output = y - output
    self.delta_output = self.error_output*self.sigmoidPrime(output)

    self.z2_error = self.delta_output.dot(self.W2.T)
    self.z2_delta = self.z2_error*self.sigmoidPrime(self.z2)

    self.W1 += x.T.dot(self.z2_delta)
    self.W2 += self.z2.T.dot(self.delta_output)
```

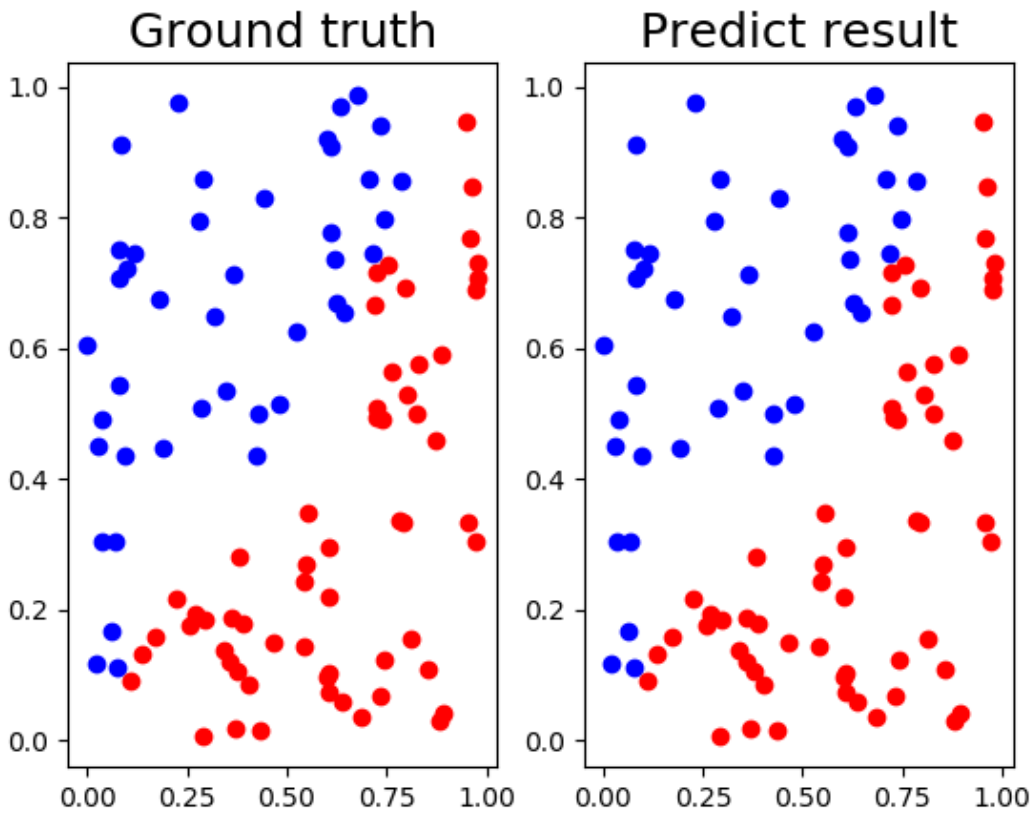
3. Results of your testing

A. Screenshot and comparison figure

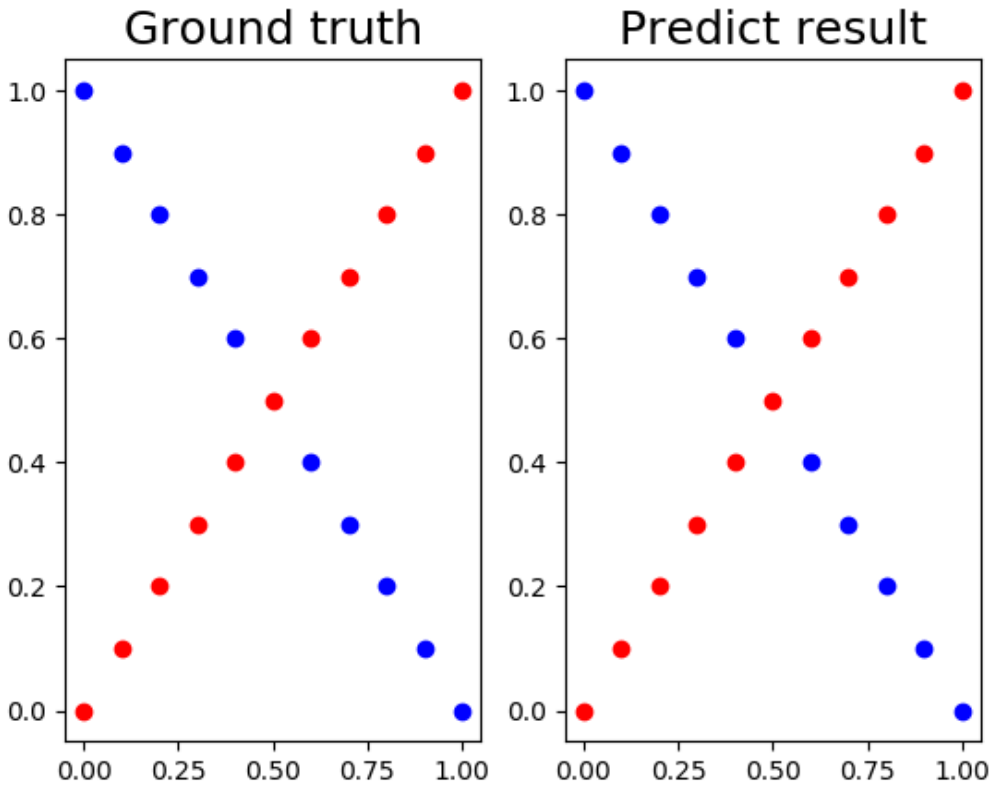
1. Linear data & XOR data 不同次數 epoch 下的 loss(MSE):

```
PS C:\Users\user\Desktop\67-0A50450\dl_course_lab\DLP_LAB1_0786027_郭權瑋> python '.\Source code.py'
linear train - epoch 0 loss : 0.34023805675413415
linear train - epoch 1000 loss : 0.0002466175654911449
linear train - epoch 2000 loss : 0.00010874839462161969
linear train - epoch 3000 loss : 6.701820932676293e-05
linear train - epoch 4000 loss : 4.75650757609277e-05
linear train - epoch 5000 loss : 3.6496037807250065e-05
linear train - epoch 6000 loss : 2.9420482476434574e-05
linear train - epoch 7000 loss : 2.4537906742844094e-05
linear train - epoch 8000 loss : 2.09808465525897e-05
linear train - epoch 9000 loss : 1.8282459463934678e-05
linear train - epoch 10000 loss : 1.6170288984881993e-05
XOR train - epoch 0 loss : 0.25999461716030875
XOR train - epoch 1000 loss : 0.006414332941561198
XOR train - epoch 2000 loss : 0.0019721862962669926
XOR train - epoch 3000 loss : 0.0011240786363397167
XOR train - epoch 4000 loss : 0.0007787542667327176
XOR train - epoch 5000 loss : 0.0005934208290907598
XOR train - epoch 6000 loss : 0.0004783705365903724
XOR train - epoch 7000 loss : 0.00040020414691285973
XOR train - epoch 8000 loss : 0.0003437267598204488
XOR train - epoch 9000 loss : 0.0003010567394364879
XOR train - epoch 10000 loss : 0.00026770721775690913
```

2. Linear data 的比較圖(predict result and ground truth):



3. XOR data 的比較圖(predict result and ground truth):



B. Accuracy rate:

```
The accuracy rate of linear dataset is: 0.9991587586187342
The accuracy rate of XOR dataset is: 0.9974586231583845
```

```python
#AC rate
def accuracy(x, y, n):
    total = 0
    for i in range(len(x)):
        bias = x[i] - y[i]
        total += abs(bias)
    return 1 - float(total / n)
```

4. Discussion

這邊滿有趣的是邊做實驗途中出現大大小小的 bug..像是

1. 一開始覺得很奇怪怎麼顯示結果(show_result fuction)

那邊都一直變不了顏色，後來才發現要用 round(float()) 去設計。

2. hidden size 也很重要，一開始設定 2，怎麼都跑不出來，後來慢慢測試(5、8、13…)最後試到 3 可以跑得很漂亮。

3. epoch 也是慢慢的從 95000->80000->40000->10000。

4. 後來有測試 n=200、300、500、1000，真的 n 越大越容易值觀看出問題在哪。

整體來說，自己 built 這個 NN 滿充實，更深刻的了解這個架構的運作過程