I.Introduction:

將二維的資料和一維的 0/1 ground-truth 丟進兩層 hidden-layer 且每層 4 個 node 的 model 中訓練，因為較簡單，因此 training set 和 testing set 使用相同的 dataset。

II.Experiment setups

A. Sigmoid functions

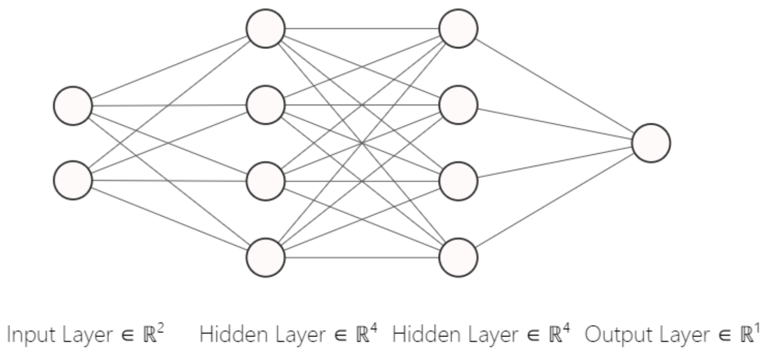
Sigmoid 函數是一種邏輯函數，又稱 S 函數，其 output 在 0~1 之間。

```
def sigmoid(x):
    return 1/(1+np.exp(-x))

def sigmoid_derivative(x):
    return x * (1 - x)
```

B. Neural Network

1.Structure:



input node: 2 個

hidden layer: 2 層

hidden node: 每層 4 個

output node: 1 個

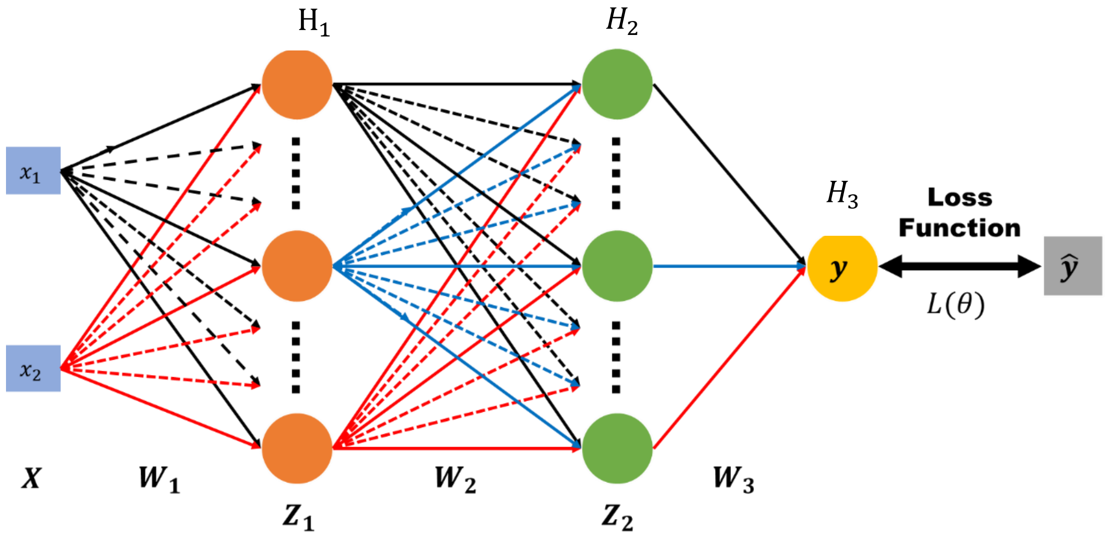
2.Neural Nerwork 之運作:

將 weight 做隨機初始化，透過 forward propogation pass value 得出 output 後，續做 back propogation，計算 gradient 得出 weight 更新的方向，再用適當的 learning rate 對每次更新的幅度作調控，以對 weight1, weight2, weight3 做更新，如此反覆迭代後得出與 ground-truth 相同的 output。

3.Class Neural Network 之函式說明:

1. Init: 隨機初始化權重，以及傳入 training data set 和 ground-truth vector
2. forward: 做向前傳播的動作，每層 layer 中的 hidden unit 存的是 weight 乘上 input 後取 sigmoid 的值
3. backprop: 詳見 C.
4. train: 每次先做 forward propogation (call function forward)，再做 backward propogation(call function backprop)， 如此反覆跌代 10000 次。
5. plot_loss: 做出 loss 和 epoch 間的關係圖

C.Backpropogation:



H_1=XW_1 H_2=Z_1W_2 H_3=Z_2W_3

Z_1=σ (H_1) Z_2=σ (H_2) y=σ (H_3)

由後往前，依序用 Loss fuction 對每個 weight 做偏微分以計算 gradient。

W_3:

W_3:

$$\frac{\partial L}{\partial W_3} = \frac{\partial L}{\partial y} \times \frac{\partial y}{\partial H_3} \times \frac{\partial H_3}{\partial W_3}$$

$$\frac{\partial L}{\partial y} = \frac{\partial (y-\hat{y})^2}{\partial y} = -2(y-\hat{y})$$

$$\frac{\partial y}{\partial H_3} = \frac{\partial \sigma(H_3)}{\partial H_3}$$

$$\frac{\partial H_3}{\partial W_3} = \frac{\partial Z_2 \times W_3}{\partial W_3} = Z_2$$

$$\therefore \frac{\partial L}{\partial W_3} = -2(y-\hat{y}) \times \frac{\partial \sigma(H_3)}{\partial H_3} \times Z_2$$

W_2:

W2:

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial y} \times \frac{\partial y}{\partial H_3} \times \frac{\partial H_3}{\partial Z_2} \times \frac{\partial Z_2}{\partial H_2} \times \frac{\partial H_2}{\partial W_2}$$

$$\frac{\partial H_3}{\partial Z_2} = \frac{\partial Z_2 \times W_3}{\partial Z_2} = W_3$$

$$\frac{\partial Z_2}{\partial H_2} = \frac{\partial \sigma(H_2)}{\partial H_2}$$

$$\frac{\partial H_2}{\partial W_2} = \frac{\partial Z_1 \times W_2}{\partial W_2} = Z_1$$

$$\therefore \frac{\partial L}{\partial W_2} = -2(y-\hat{y}) \times \frac{\partial \sigma(H_3)}{\partial (H_3)} \times W_3 \times \frac{\partial \sigma(H_2)}{\partial H_2} \times Z_1$$

W_1:

W1:

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial y} \times \frac{\partial y}{\partial H_3} \times \frac{\partial H_3}{\partial Z_2} \times \frac{\partial Z_2}{\partial H_2} \times \frac{\partial H_2}{\partial Z_1} \times \frac{\partial Z_1}{\partial H_1} \times \frac{\partial H_1}{\partial W_1}$$

$$\frac{\partial H_2}{\partial Z_1} = \frac{\partial Z_1 \times W_2}{\partial Z_1} = W_2$$

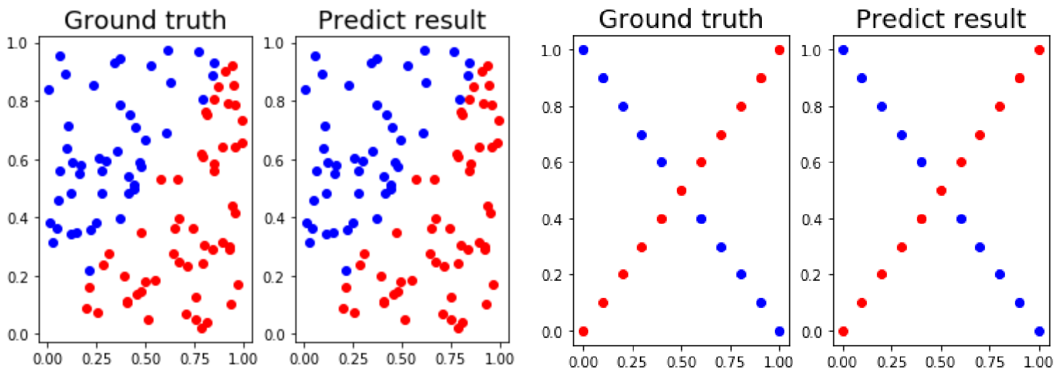
$$\frac{\partial Z_1}{\partial H_1} = \frac{\partial \sigma(H_1)}{\partial H_1}$$

$$\frac{\partial H_1}{\partial W_1} = \frac{\partial X \times W_1}{\partial W_1} = X$$

$$\therefore \frac{\partial L}{\partial W_1} = -2(y-\hat{y}) \times \frac{\partial \sigma(H_3)}{\partial H_3} \times W_3 \times \frac{\partial \sigma(H_2)}{\partial H_2} \times W_2 \times \frac{\partial \sigma(H_1)}{\partial H_1} \times X$$

III.Results of testing (model parameter: lr=0.1, Epoch=10000)

A. Result

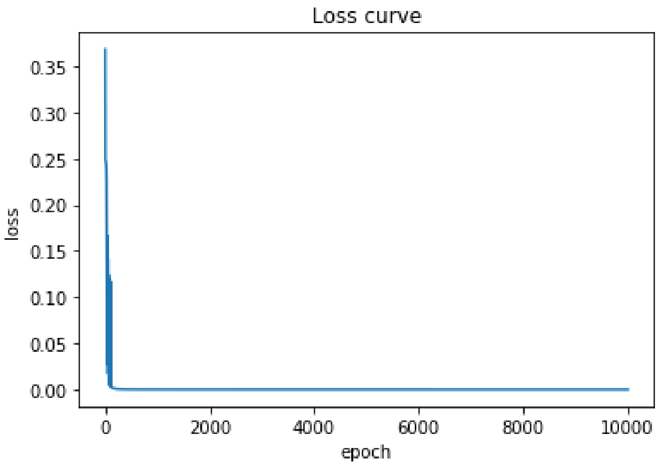


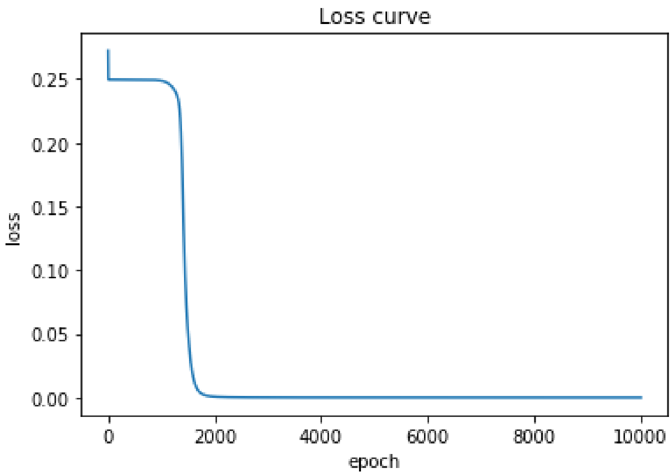
B.Accuracy

```
Console 1/A
Accuracy : 100.0 %
Epoch:7000 Loss: 5.76343587607012e-06
Accuracy : 100.0 %
Epoch:7500 Loss: 5.4056547739229685e-06
Accuracy : 100.0 %
Epoch:8000 Loss: 5.092301376825656e-06
Accuracy : 100.0 %
Epoch:8500 Loss: 4.815369277828499e-06
Accuracy : 100.0 %
Epoch:9000 Loss: 4.568688489781288e-06
Accuracy : 100.0 %
Epoch:9500 Loss: 4.347425044018901e-06
Accuracy : 100.0 %
Epoch:10000 Loss: 4.1477360221935484e-06
Accuracy : 100.0 %

In [780]:
```

C.Loss-Epoch curve(linear/XOR)



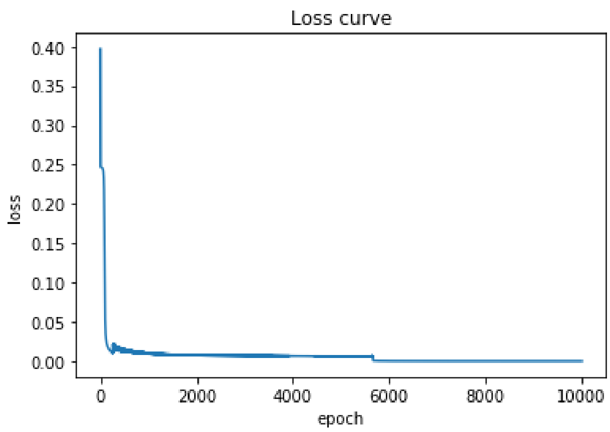


IV. Discussion

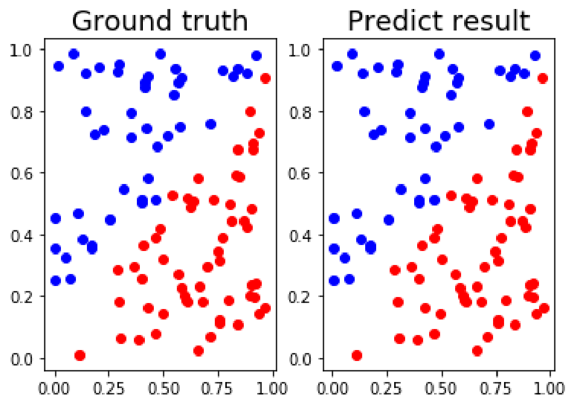
A. Adjust learning rate(lr):

<table>
<tr><td>lr=0.05</td><td>lr=0.4</td></tr>
<tr><td>

Loss: 5.418680150047533e-06:



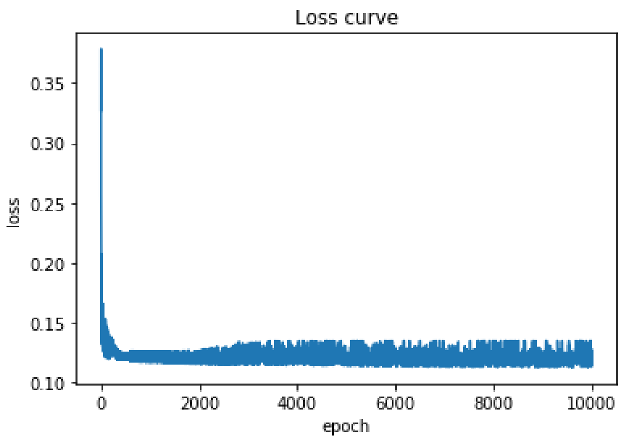
Acc: 100%



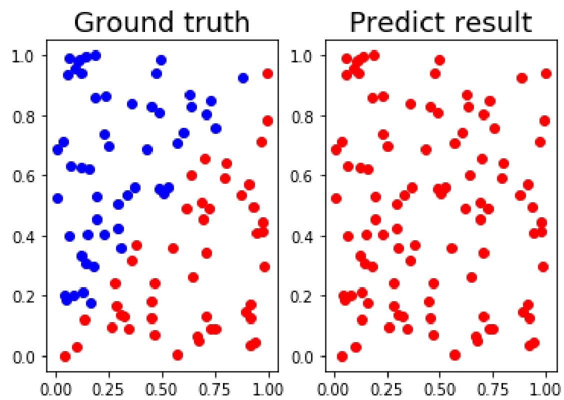
linear

</td><td>

Loss: 0.13002231505468648



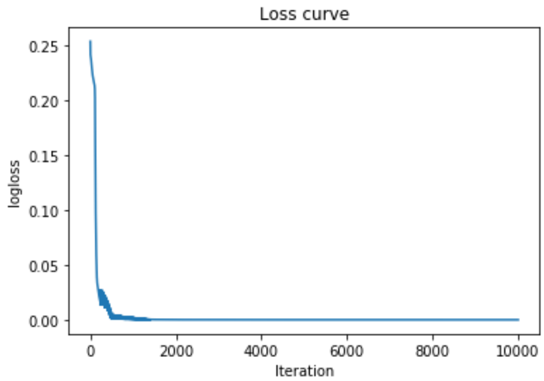
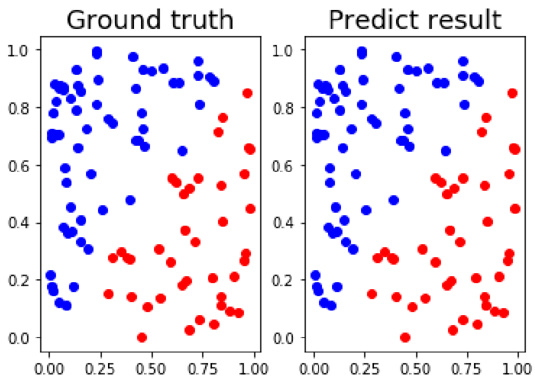
Acc: 55%



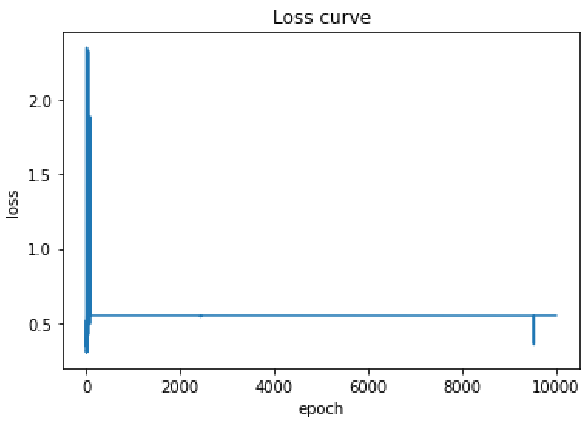
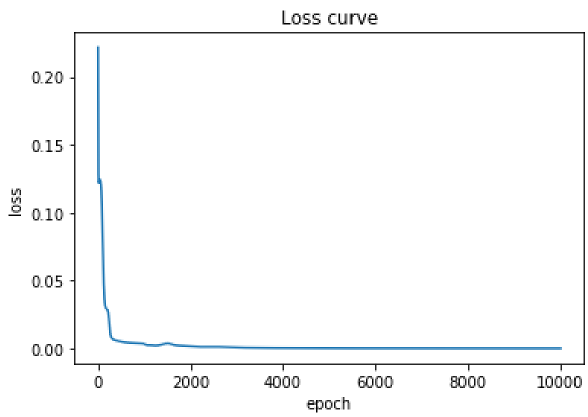
linear

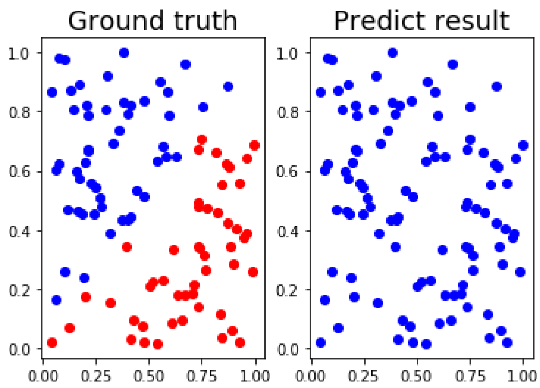
</td></tr>
</table>

B. Adjust hidden nodes:

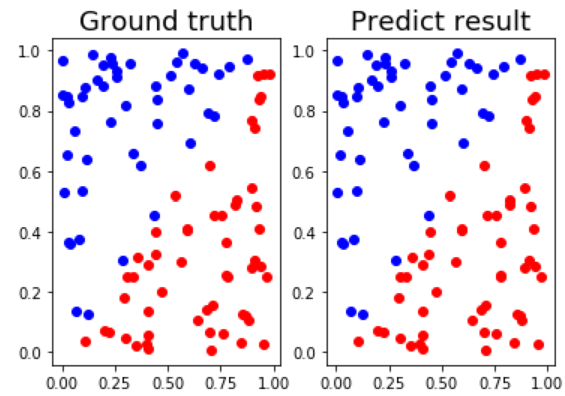
2 Hidden nodes for each layer
 Loss: 6.65110056020496e-06 Acc: 100%  linear

C.Without sigmoid(use tanh):

參數與原先 model 相同	Lr=0.01，其餘參數不變
	
Loss: 0.5499999999990769 Accuracy: 45%	Loss: 4.7283110830964585e-05 Accuracy: 100%



linear



linear