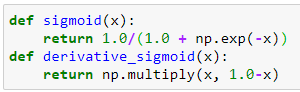
Report Spec

1. Introduction

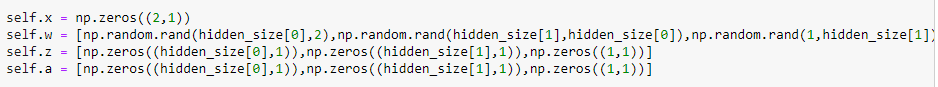
在Lab1實作中，主要為練習與了解當output出現類似XOR這種無法用linear function 來準確分別結果時，我們會需要使用 nonlinear 的 activation function 例如 : sigmoid,relu,softmax…

但當使用這種nonlinear function時，gradient descent會變得難尋找到其中參數的最佳解，而本次實驗將使用backpropagation來快速尋找各個weight對應的gradient descent，進而在每次的step中更新weight值

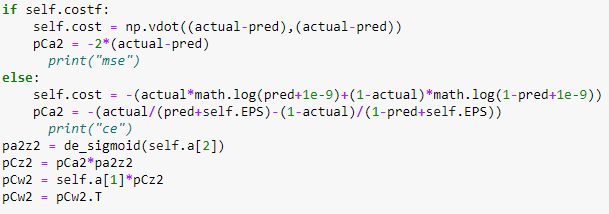
1. Experiment setups
   * 1. Sigmoid function

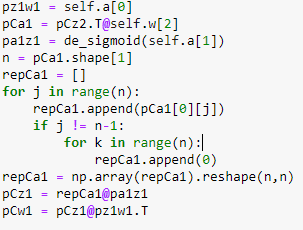
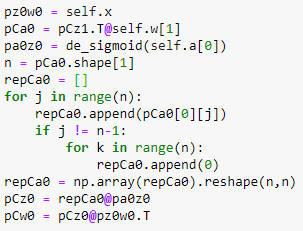


* + 1. Neural network (hidden layer可自己設定)



* + 1. Backpropagation



1. Results of your testing

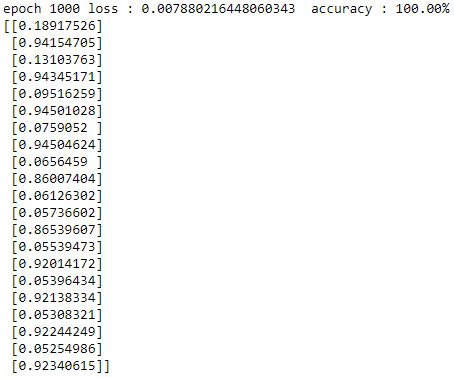
A. Screenshot and comparison & B. figureShow the accuracy of your prediction

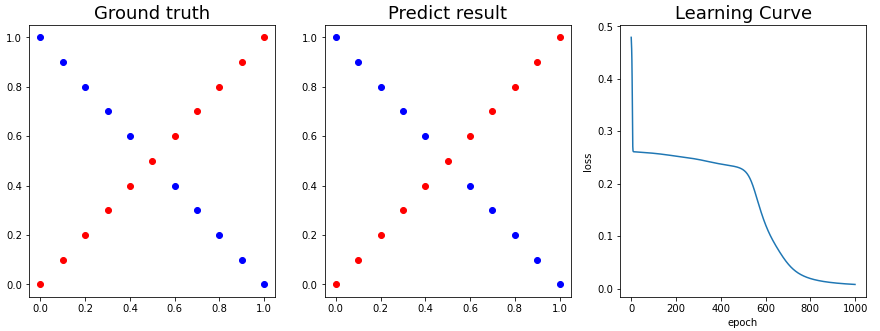
|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

C. Learning curve(loss, epoch curve)

hyperparmeter set : hidden\_size = (8,8) / learning\_rate = 0.05 / epoch = 1000 /

activation function( hide1 / hide2 / output\_layer) = ( no / sigmoid / sigmoid )

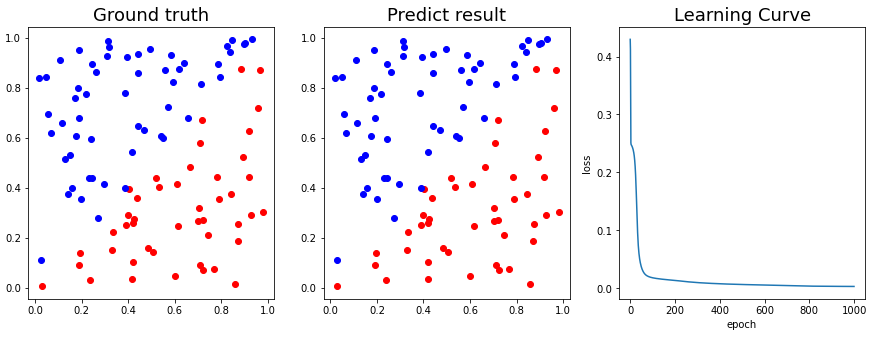




D. anything you want to present

the result of linear case

在hyperparmeter不變為前提之下，loss降的速度快nonlinear很多



1. **Discussion**

A. Try different learning rates

hyperparameter :

epoch 1000 , hide\_layer(8,8 ), cross entropy , activation funt : (hide1 : no ,hide2 : relu, hide3 : sigmoid)

|  |  |
| --- | --- |
| 1. lr = 5 | 1. lr = 1 |
| 1. 0.5 | 1. 0.1 |
| 1. 0.05 | 1. 0.01 |

總結 :

learning\_rate 過小 > 要train較久 , 右下lr=0.01要等到800+才開始降loss

learning\_rate 過大 > 找不到optimal的地方

1. Try different numbers of hidden units

hyperparameter : epoch 3000, learning\_rate 0.05, cross entropy

activation funt : (hide1 : no ,hide2 : relu, hide3 : sigmoid)

|  |  |
| --- | --- |
| 1. hidden\_unit = (2,2) | 1. hidden\_unit = (4,4) |
| 1. (8,8) | 1. (32,32) |
| 1. (16,2) | 1. (2,16) |

總結 : hidden\_unit

過多 > code runtime時間變長 ,適當的數量可以增加訓練速度(loss降的快)

過少 > loss卡住 train不起來，適當的數量減少code runtime

另外有發現後面的hidden\_unit較多效果比較好 (左下、右下)

1. Try without activation functions lazy to use

hyperparameter : epoch 10000, learning\_rate 0.05, , hidden\_unit(4,4), mean square error

|  |  |
| --- | --- |
| 1. ( no , no , no )     loss會有overflow問題 擴散 | 1. ( no , no ,sigmoid)     雖然沒有擴散的問題，但沒有辦法有效的收斂 |
| 1. (relu, no , sigmoid ) (穿插)     穿插使用 無法收斂 | 1. ( no , relu, sigmoid )     減少第一層的使用 效果最好 |
| 1. ( sigmoid , sigmoid , sigmoid )     可以收斂但速度慢 | 1. ( relu , sigmoid , sigmoid )     可以收斂但速度較慢 |

1. Anything you want to share

output without activation function 部分討論

首先without activation function 無法使用太大的hidden unit ，若使用過多的unit，

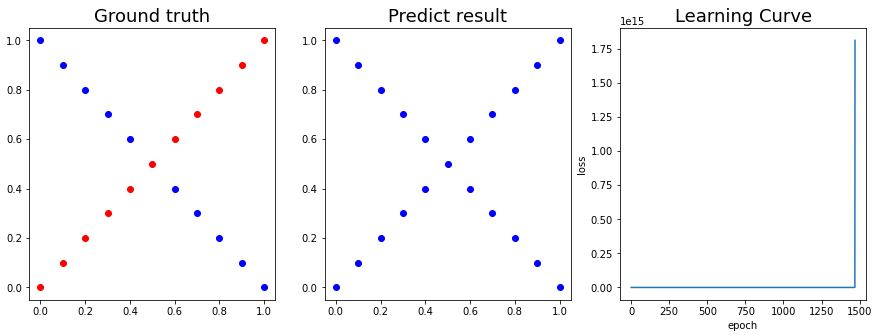
容易因為初始weight設定的關係，造成loss上的overflow

* 1. 另外會因為答案屬於linear or nonlinear分布，產生不同的結果，實驗如下

hyperparameter : epoch 10000, learning\_rate 0.05, , hidden\_unit(2,2), mean square error

activation funt : (hide1 : no ,hide2 : no, hide3 : no)

‧若答案屬於nonlinear分布 則Loss無法收斂



‧但答案屬於 linear 分布時，雖不到100%正確但至少有個大概的方向



