DLP HW1

309553008

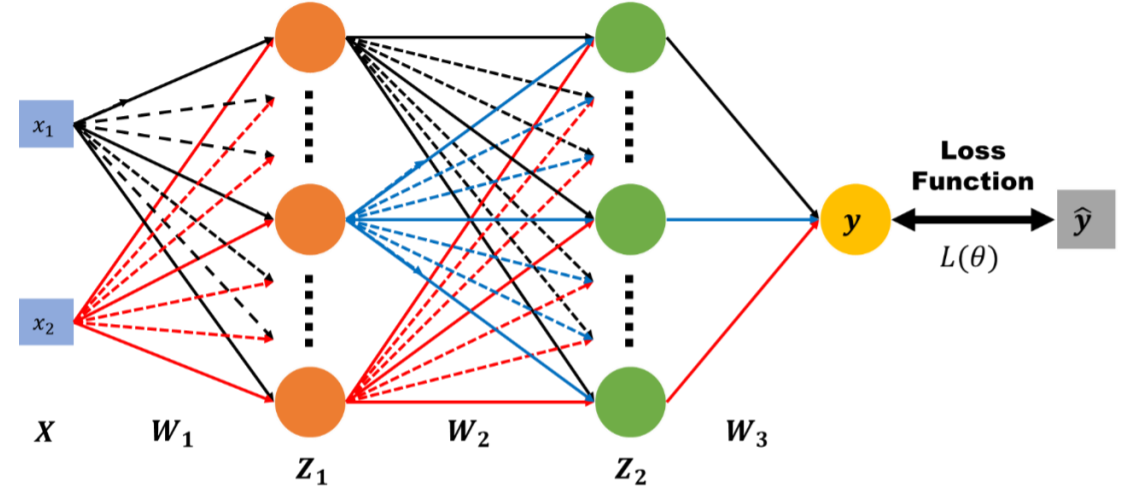
1. Introduction

這次作業是要實作一個有兩層hidden layer、一層input layer、一層output layer的neural network，首先input經過forward propagation得到一個預測值y，將此預測值與ground truth比對得到Loss，再將此Loss分別對不同的weight做偏微分得到該weight的斜率，然後根據斜率更新weight (backward propagation)，最後依據更新後的weight做下一輪迭代直到fit。

1. Experiment setups
2. Sigmoid functions :

Sigmoid functions是一種值域在[0, 1]的一個函數。他可以把值壓在[0, 1]之間，算是一種normalize，經過Sigmoid functions後，就不會因為原本的值差異太大而影響network，而且因為這是一個非線性函數，這使得network可以fit非線性的資料。

1. Neural network :



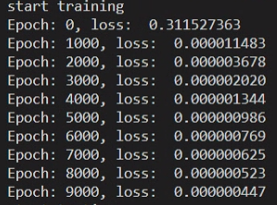
這是一個有兩層hidden layer的fully connected network，input為一個點的座標(x, y)，經過W1 = [w1, w2, ……, whidden\_unit\_num]後得到Z1，Z1經過W2得到Z2，Z2經過W3最後的到預測的y，藉由預測的y與ground truth y計算loss (我是採用MSE)，其中W1、W2、W3分別是[2, hidden\_unit\_num], [hidden\_unit\_num, hidden\_unit\_num], [hidden\_unit\_num, 1]的矩陣

1. Backpropagation :

藉由Loss (L(**θ**))對不同的W做偏微分，我們可以得到Loss對該W的Gradient。藉由這些Gradient，我們可以去更新W使Loss達到最小。

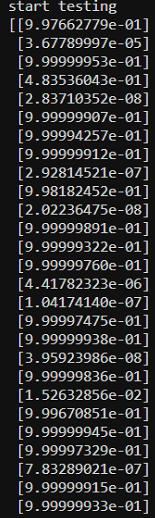
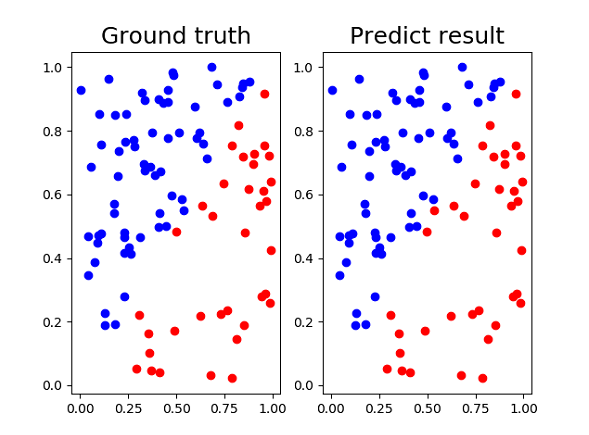
1. Results of your testing
2. Screenshot and comparison figure

Training(linear):

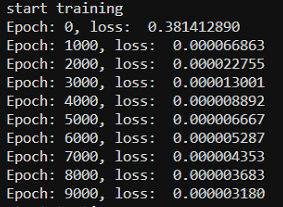


Testing(linear):

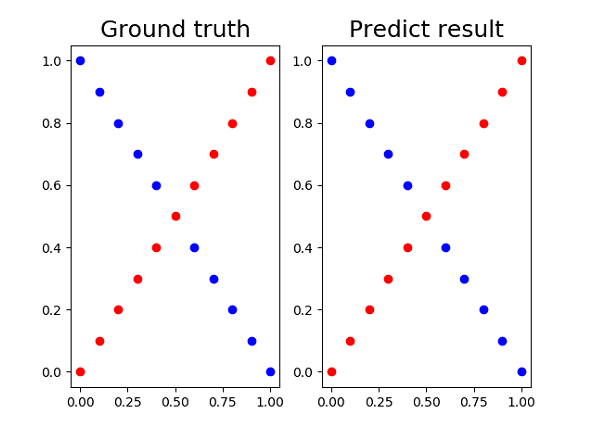
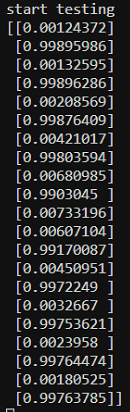
(因為有100個點，這邊貼其中幾個示意)

Training(XOR):



Testing(XOR):

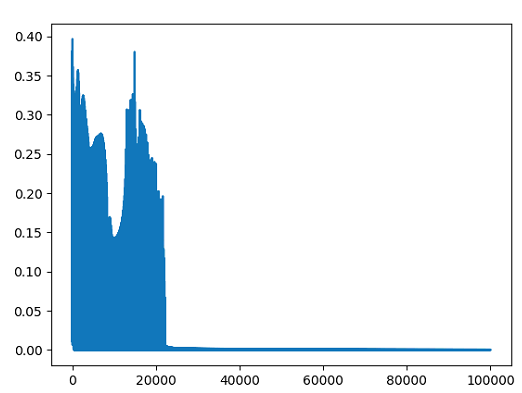


1. Show the accuracy of your prediction

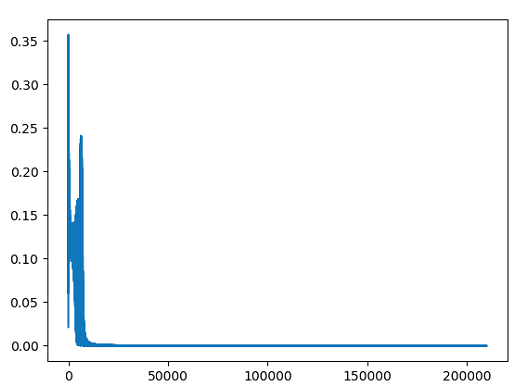
不管是linear還是XOR都是100%

1. Learning curve (loss, epoch curve)

Linear:

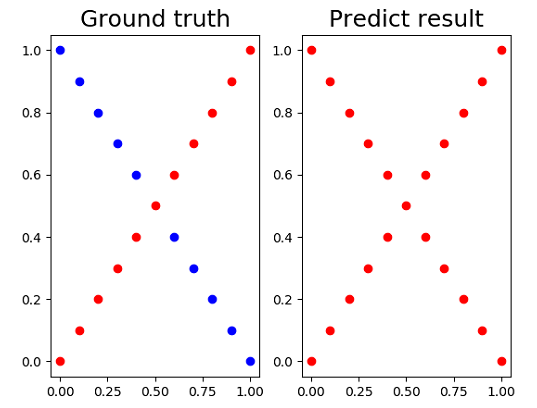


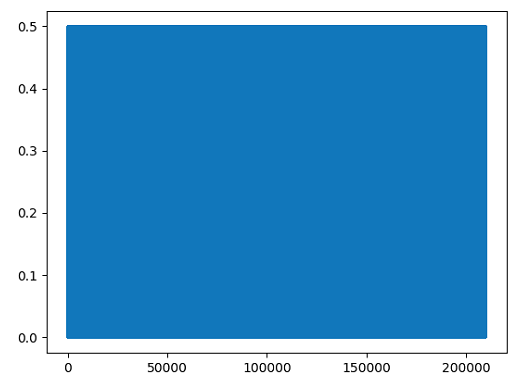
XOR:



1. Discussion
2. Try different learning rates

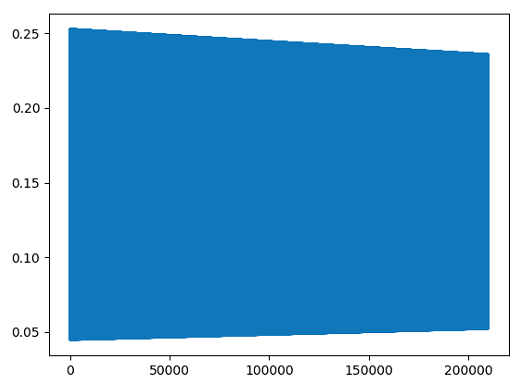
如同助教課所說的一樣，如果learning rate太大，會讓點一直震盪，無法收斂到谷底(Loss最小的點)以下面的圖來說，我把learning rate開超大，Loss只能在0.5就下不去了







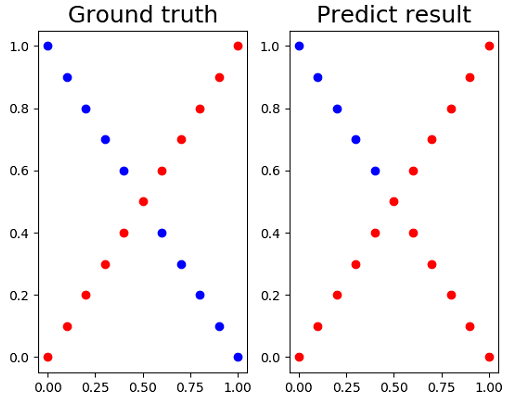
而learning rate太小，則收斂速度太慢





1. Try different numbers of hidden units

以XOR來說，如果hidden unit太少會造成結果不好，因為這樣簡單的架構無法fit這個問題(下圖為hidden units = 3)

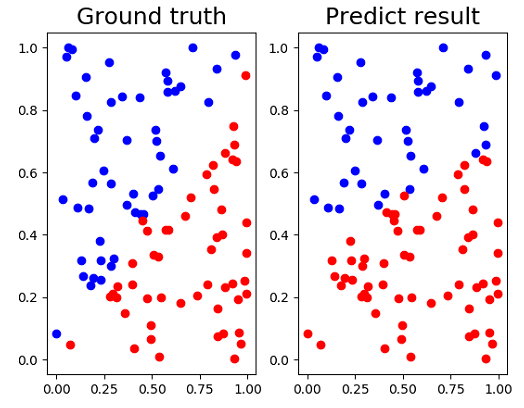




1. Try without activation functions

剛開始的時候一直爆開，後來想一想發現是因為沒有用activation function，所以值會變得比有用大很多，如果這個情況下用原本的learning rate的話就會爆開，後來把learning rate縮小後就可以跑了。

這是linear的結果，雖然沒有很好，但是正確率還是有82%，應該是因為這data本來就是線性的，所以效果還可以



至於XOR的話就因為data不是線性，所以這個network無法fit，正確率就下去了(33.33%)

