DLP Lab1 Backpropagation

0711529 陳冠儒

1. Introduction (20%)

建造一個具有兩層hidden layers的model，並計算forward和backward propagation，來預測Linear和XOR dataset。

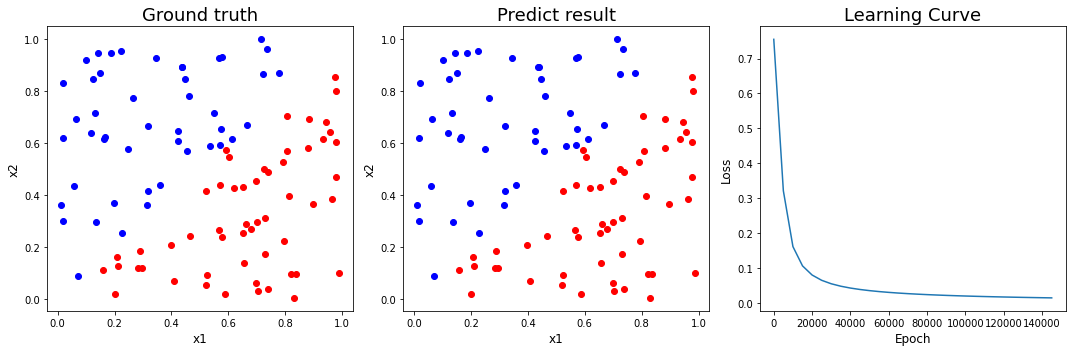
Dataset：

x: shape = (number of data, input dimension)

y: shape = (number of data, label)

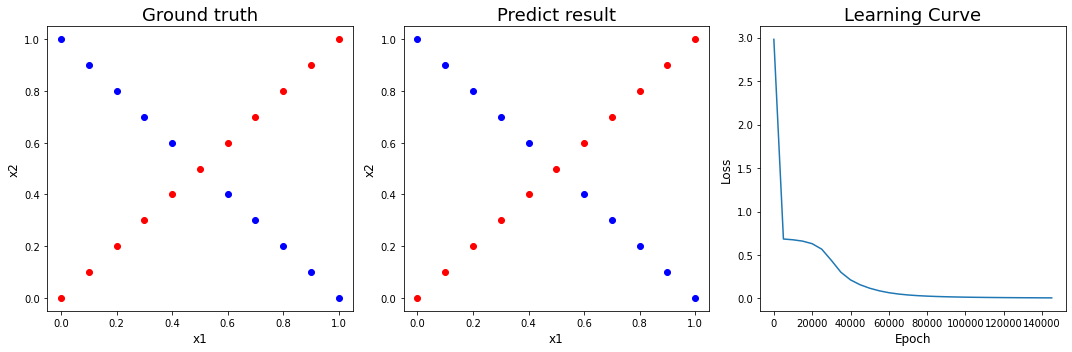
1. Linear

* label = 0, x1 > x2 which falls on the bottom-right side.
* label = 1, x1 < x2 which falls on the upper-left side.



1. XOR

* label = 0, x2 = x1
* label = 1, x2 = 1 - x1



1. Experiment setups(30%):
   1. Sigmoid functions

* 簡介

Sigmoid functions可以用來當作hidden layer的activation function來使其成non-linear，也可以用在output layer使其成Bernoulli output distributions。

* Sigmoid function在數學上的介紹

Sigmoid function是logistic function的一個特例。

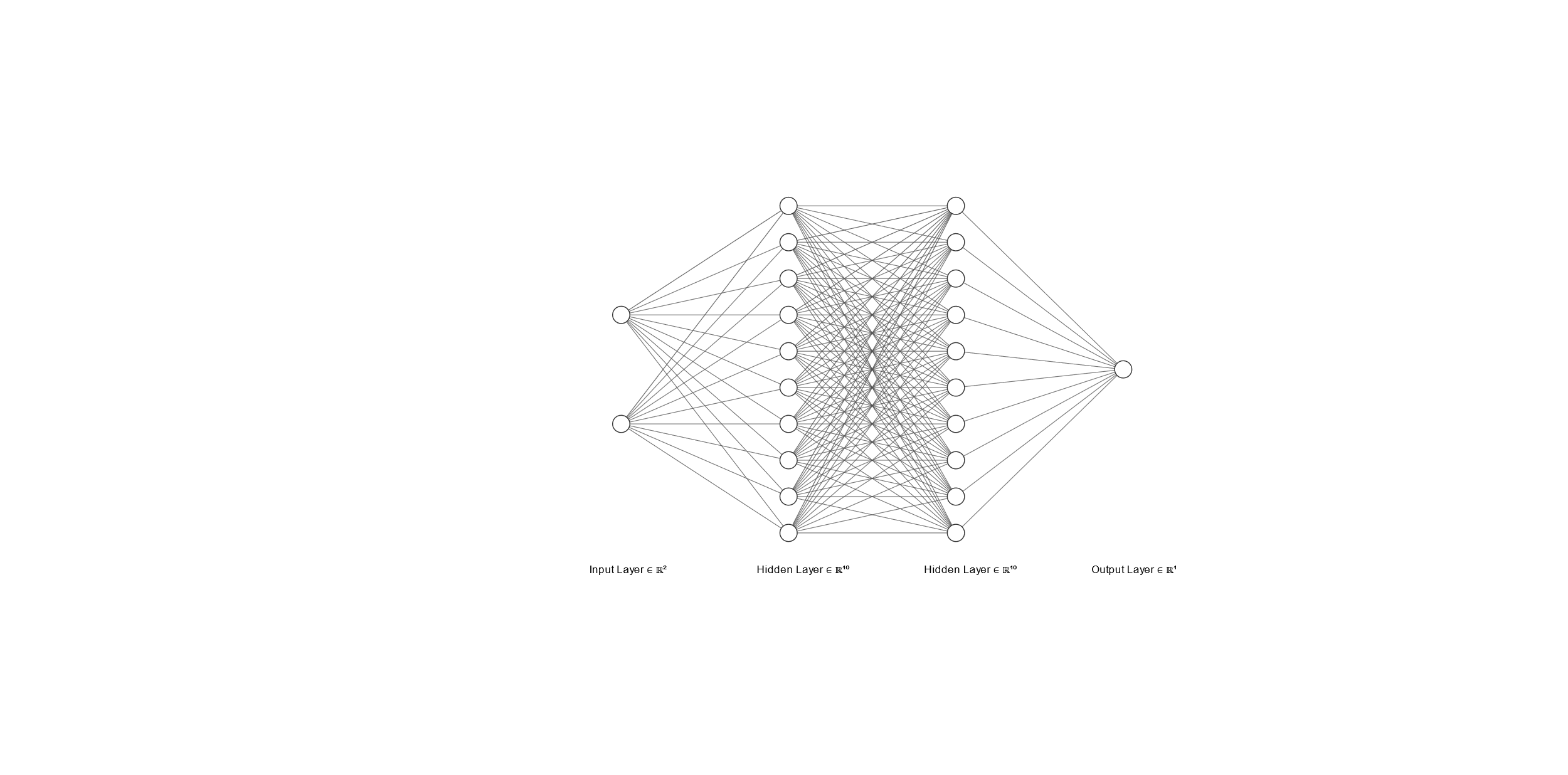
L控制最大最小值、k為最小到最大的變化速率、為x的偏移量，而sigmoid function就是當L = 1、k = 1且=0，故在output layer用sigmoid function的二分法model也被稱為logistic regression。

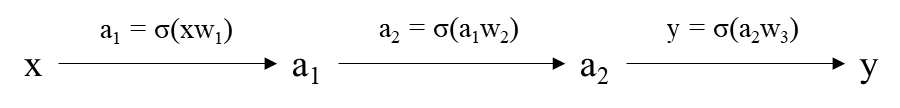
* Derivation of sigmoid function
* Sigmoid優點

中央區的信號增益較大，對兩側區的信號增益小，在信號的特徵空間映射上佳，學習時可以將重點特徵推向中央區，將非重點特徵推向兩側區。。

* Sigmoid缺點

1. 均值不為0，意味著自帶了一個bias，在計算時是額外的負擔，這會使得收斂變得更慢。
2. 在其飽和區會變化的很緩慢，微分趨近於0，容易造成梯度消失，使得訊息丟失。
   1. Neural network





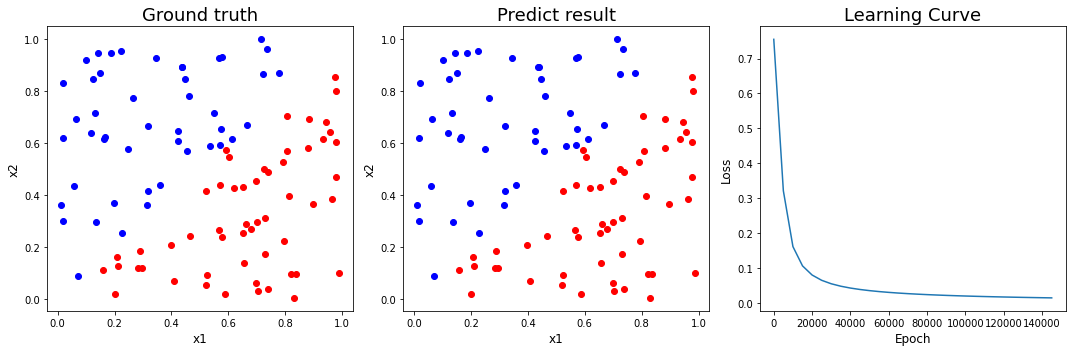
* Neural Architecture & parameters design
* hidden units = ( 10, 10 )
* learning rate = 0.01
* Epoch = 150000
* Training data number: (1.) linear: 100 (2.) XOR: 21
* Testing data number: (1.) linear: 100 (2.) XOR: 61
* Loss Function

我使用Maximum Likelihood來當作loss function，上式所代表的是兩個Bernoulli distribution的cross entropy，即算出y distribution和ypred distribution的距離，minimizing cross entropy就是希望這兩個越接近越好。

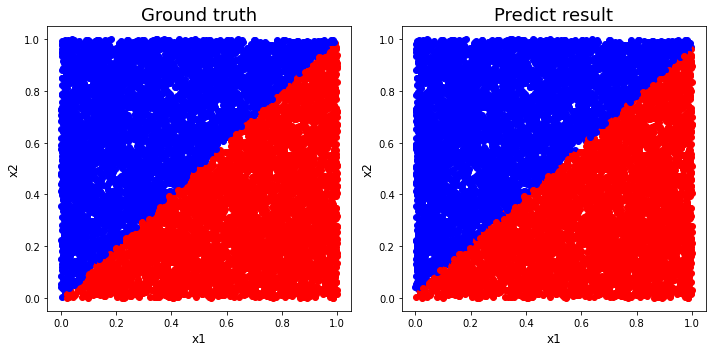
* 1. Backpropagation
* , where
* , where
* , where

1. Results of your testing (20%)
   1. Screenshot and comparison figure
      1. Linear  
         Testing data generating by changing the random seed

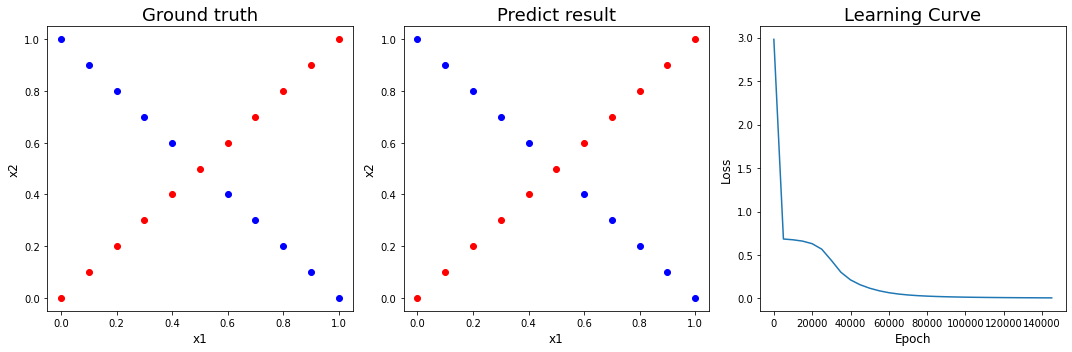
* n\_data = 100 → accuracy = 1.0



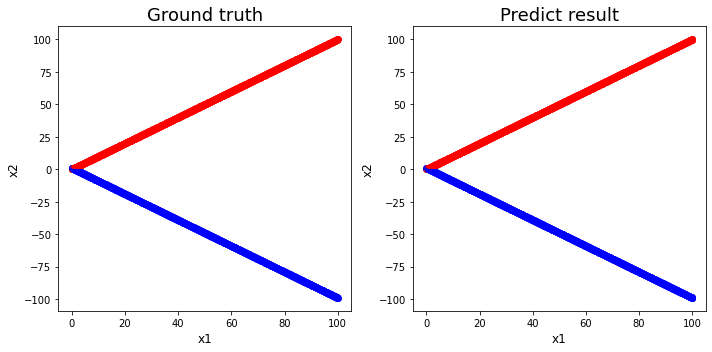
* n\_data = 10000 → accuracy = 0.984



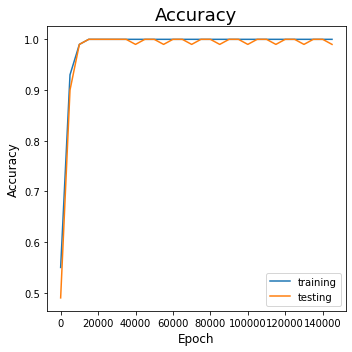
* + 1. XOR  
       Testing data generating by changing the number of points the function generated.
* n\_data = 21 → accuracy = 1.0



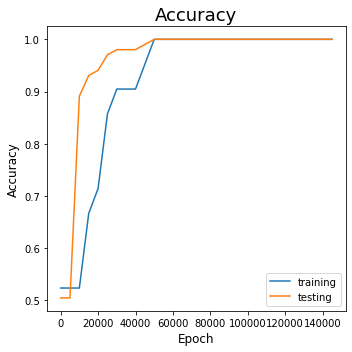
* n\_data = 2001 → accuracy = 1.0



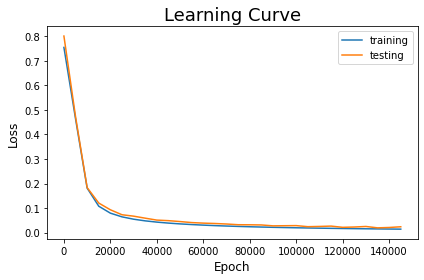
* 1. Show the accuracy of your prediction
     1. Linear



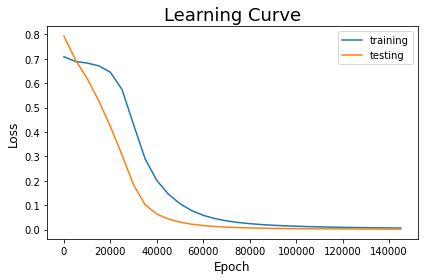
* + 1. XOR



* 1. Learning curve (loss, epoch curve)
     1. Linear



* + 1. XOR



* 1. Anything you want to present

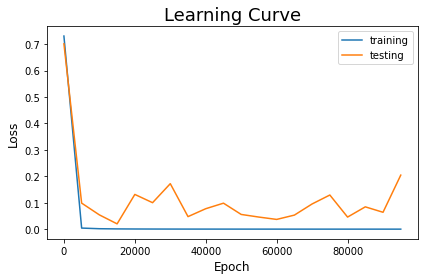
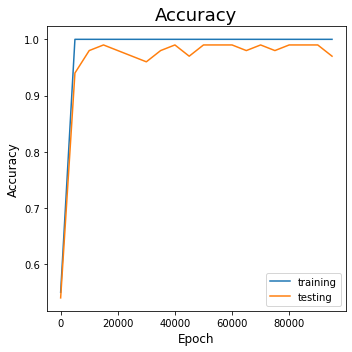
XOR function accuracy跟learning curve testing都優於training的原因，是因為其function較為簡單學習，且dataset是依照同一個線性function所產生的，所以在training時，因為只有21個資料點，會由於樣本少，而使得每一個錯誤對cost和accuracy的影響被放大，但在testing中，因為資料點多(101個資料點)，所以錯誤的影響會被稀釋。

1. Discussion (30%)
   1. Try different learning rates

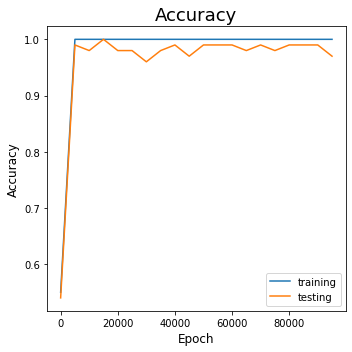
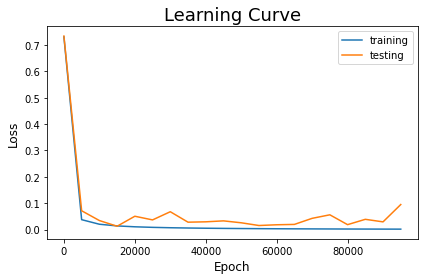
固定Epoch = 100000、hidden units = ( 5, 5 )。

1. Linear

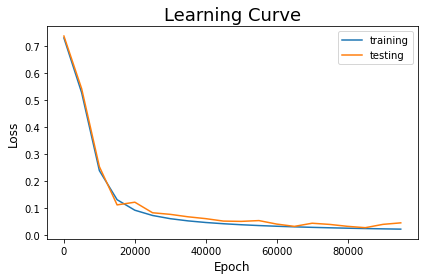
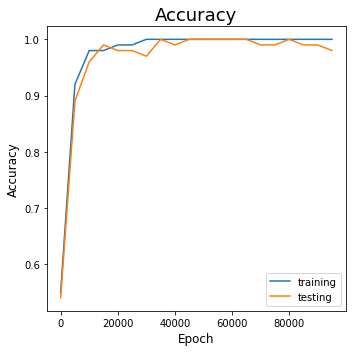
* Learning rate = 1

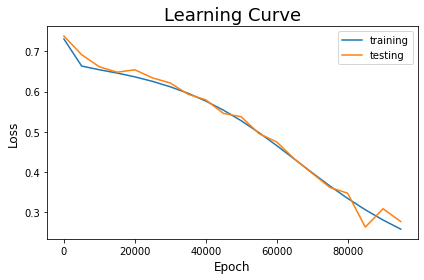
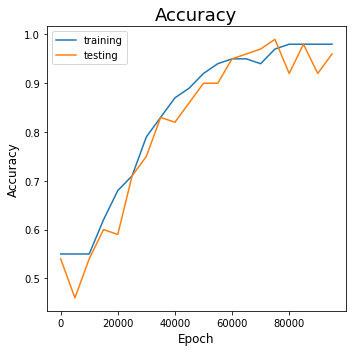
* Learning rate = 0.1



* Learning rate = 0.01

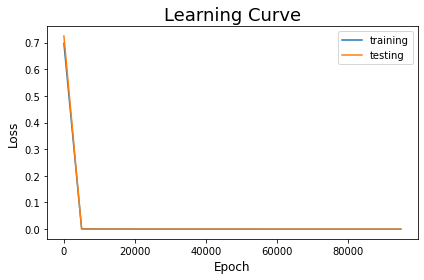
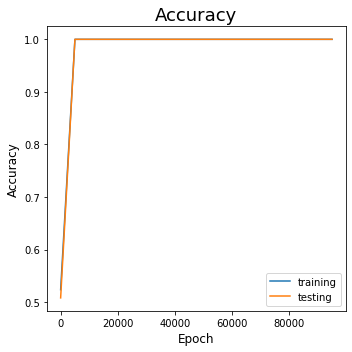
 

* Learning rate = 0.001

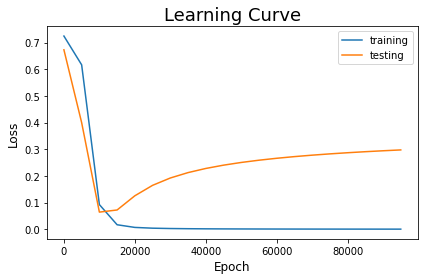
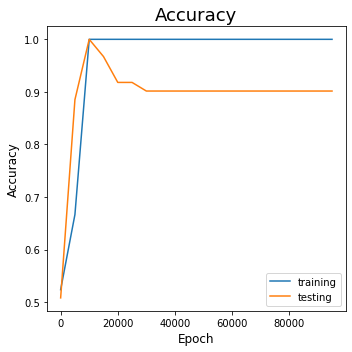
 

1. XOR

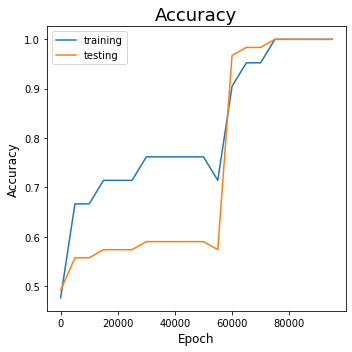
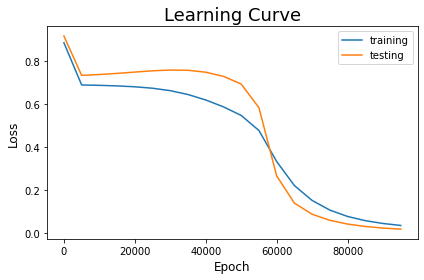
* Learning rate = 1

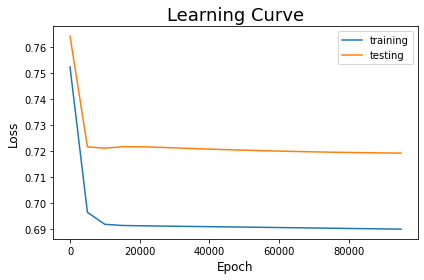
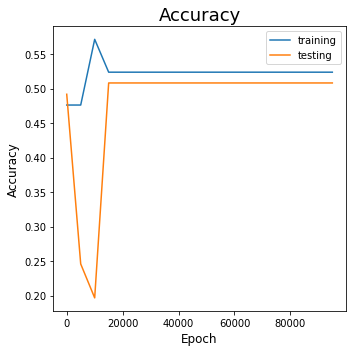
* Learning rate = 0.1

* Learning rate = 0.01



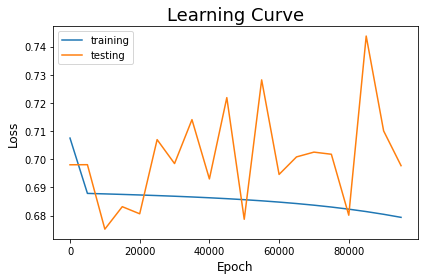
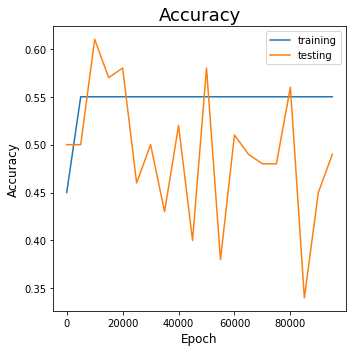
* Learning rate = 0.001

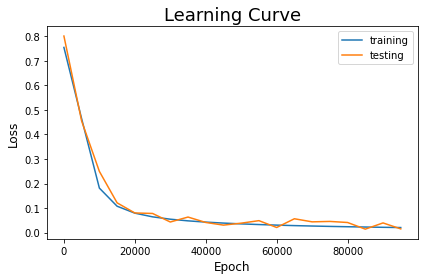
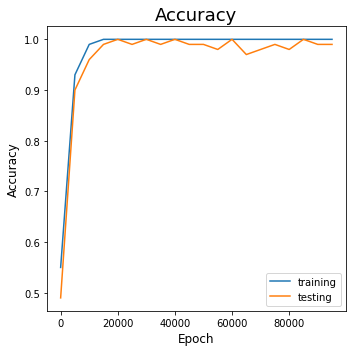
* 1. Try different numbers of hidden units

固定Epoch = 100000、Learning rate = 0.01。

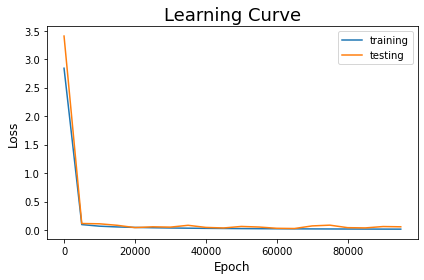
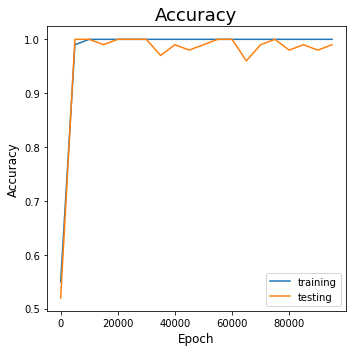
* + 1. Linear
* hidden units = ( 1, 1 )

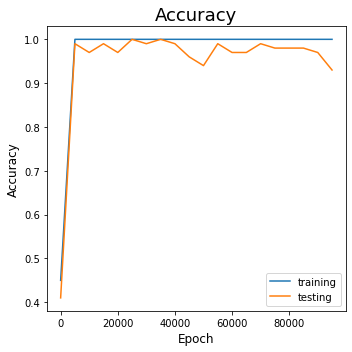
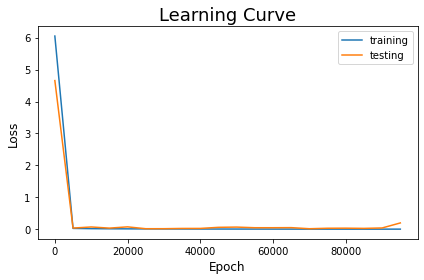
* hidden units = ( 10, 10 )

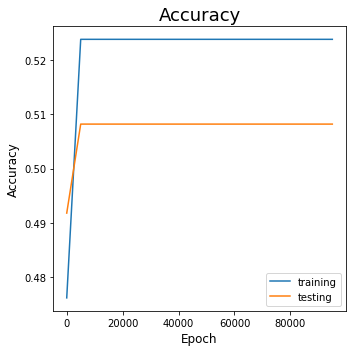
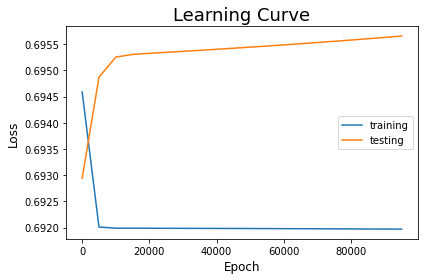
* hidden units = ( 100, 100 )

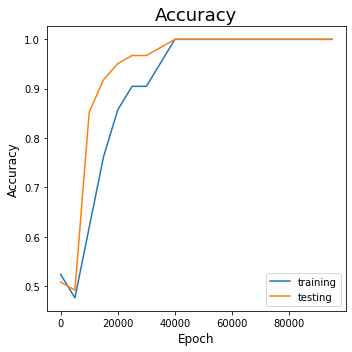
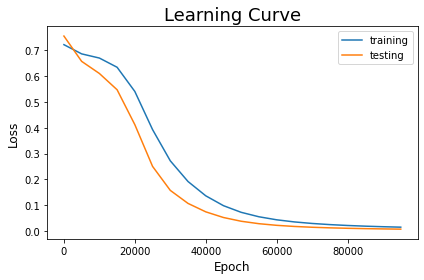
* hidden units = ( 500, 500 )



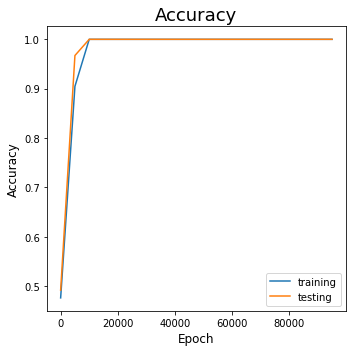
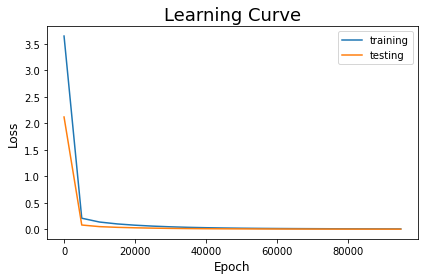
* + 1. XOR
* hidden units = ( 1, 1 )



* hidden units = ( 10, 10 )



* hidden units = ( 100, 100 )



* hidden units = ( 500, 500 )
  1. Try without activation functions
  2. Anything you want to share
* Use Different Activation Function at hidden layer
* tanh
* ReLU
* Use Different Loss Function
* MSE

1. Reference

* 也談激活函數Sigmoid, Tanh, ReLu, softplus, softmax：https://zhuanlan.zhihu.com/p/48776056