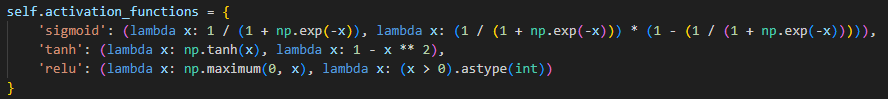
1. Introduction

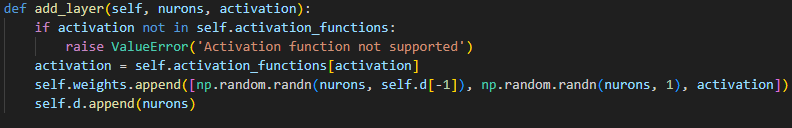
深度學習在各項領域中表現優秀，我們要去了解其背後的原理以及概念，本次LAB透過練習不使用現成函式庫，手刻深度學習模型，並且去實作所有細節

1. Implementation Details
   1. Sigmoid function

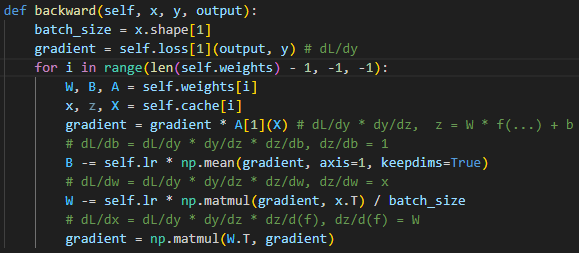
由於是activation function，最好是可微分的函式，實作上建立了一個字典，儲存各種activation function，字典的value為一個tuple，儲存函數和導函數。

* 1. Neural network architecture

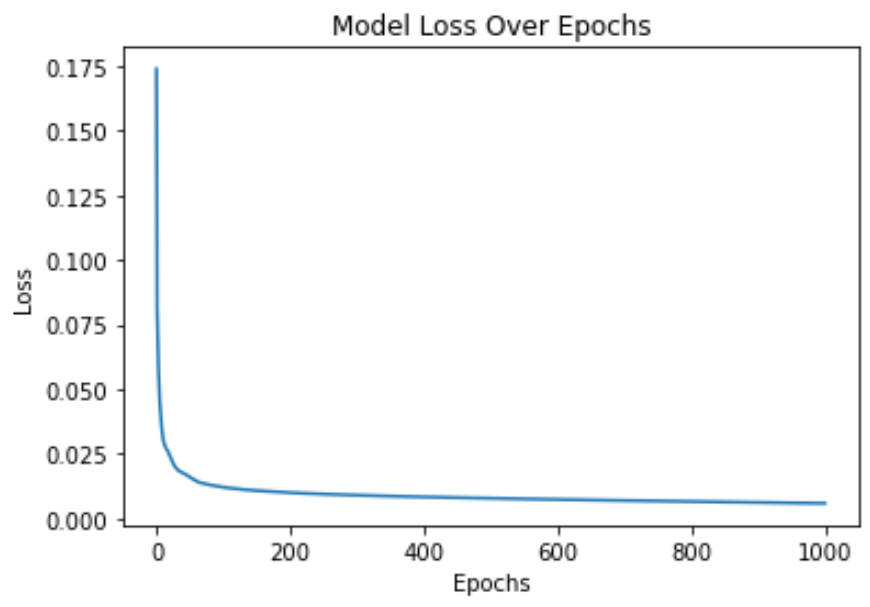
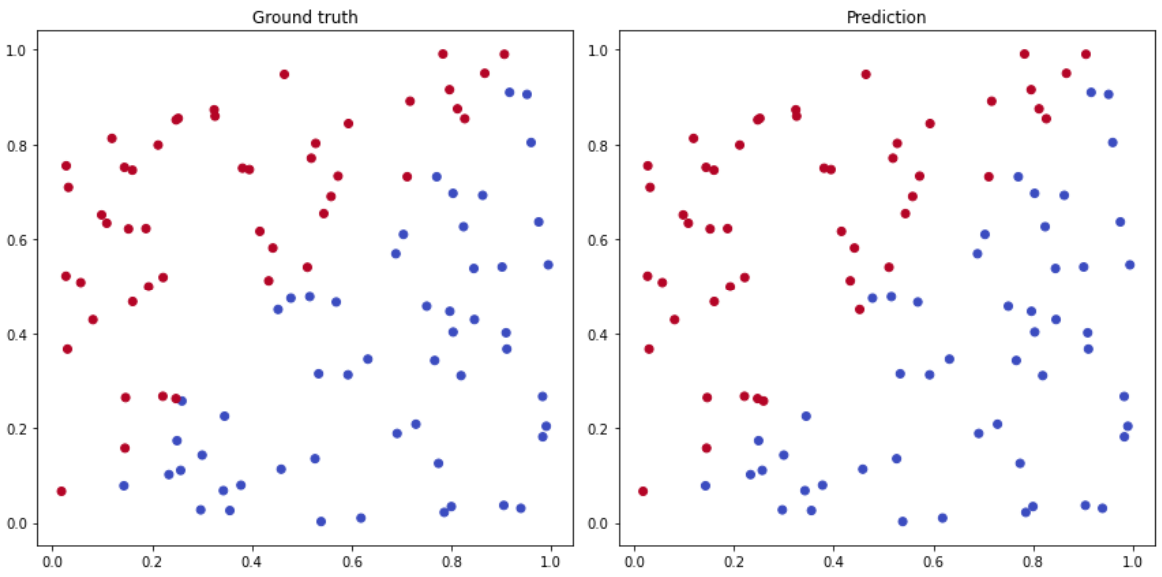
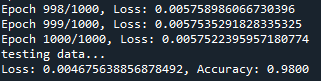
使用了add\_layer函式，能夠方便動態新增不同樣式的layer，weights裡面儲存的是每一層的參數，分別是W、bias、activation function。

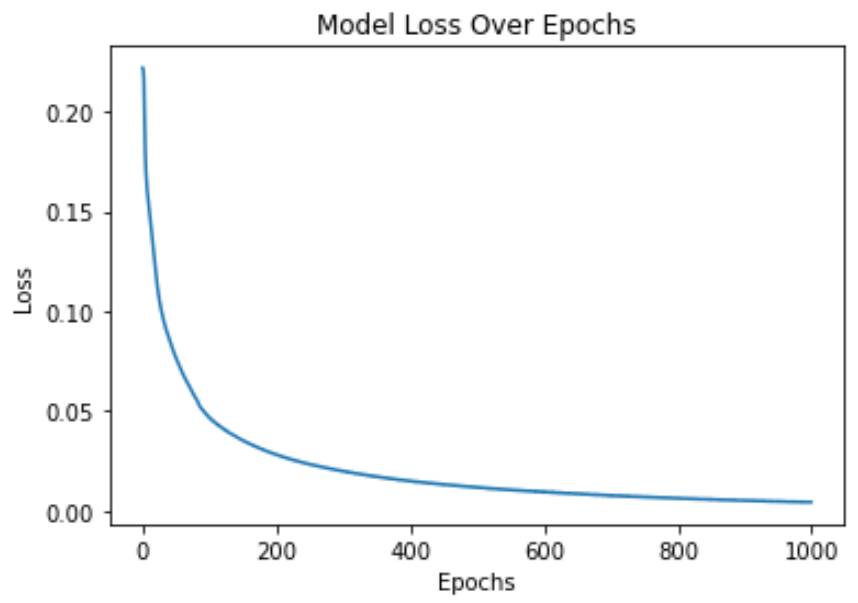
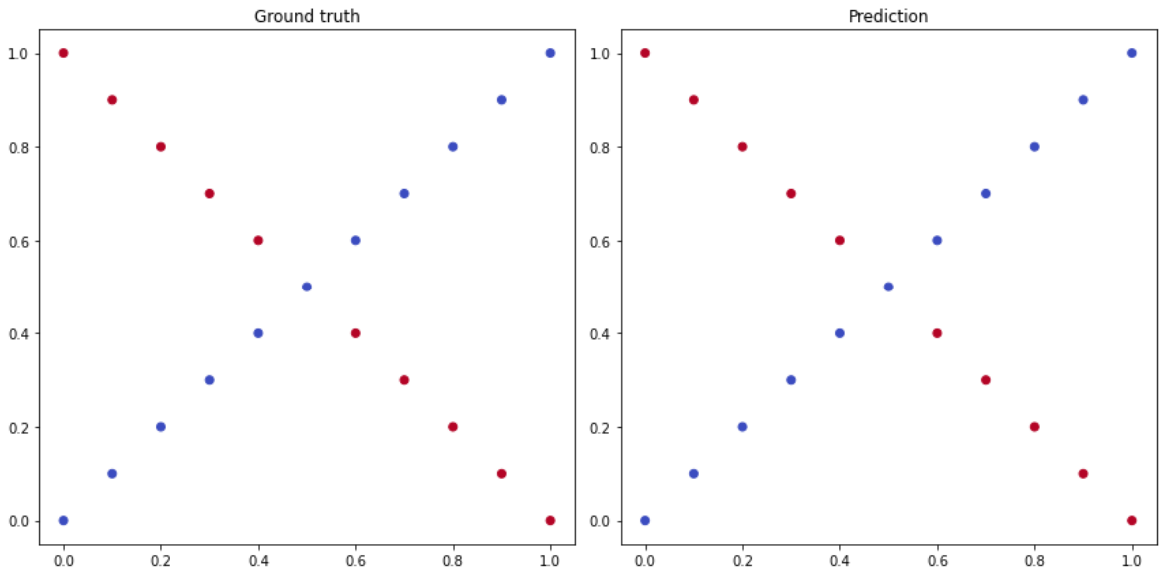
* 1. Back-propagation

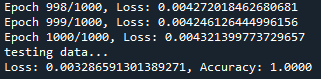
根據loss function，去計算他對模型參數的梯度是多少，因為可以使用chain rule，所以從output的地方往回更新回去比較快，cache[i]裡儲存的是第i層layer中，input x在forward中的轉換變化，而其中X=activation(Wx+b)，z=Wx+b。

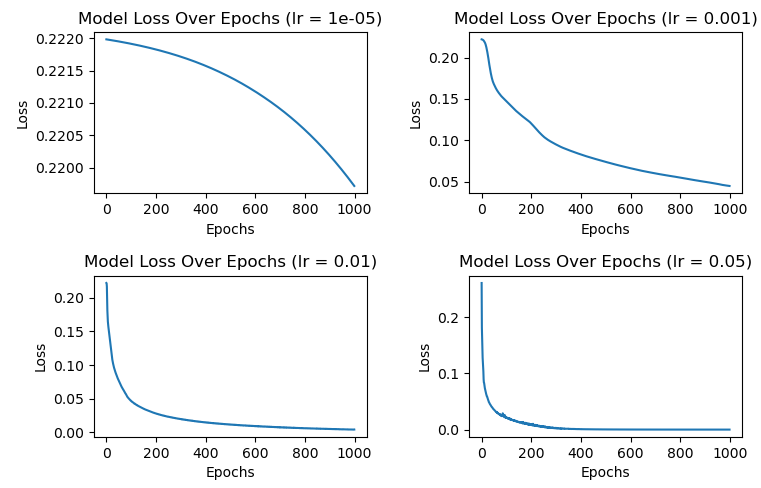
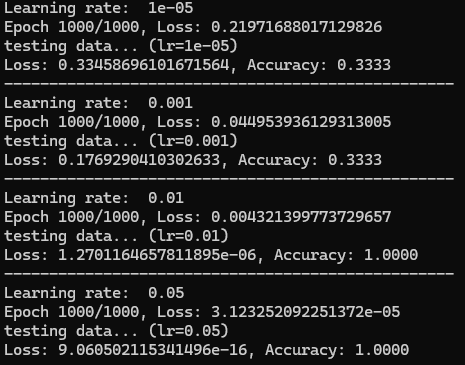
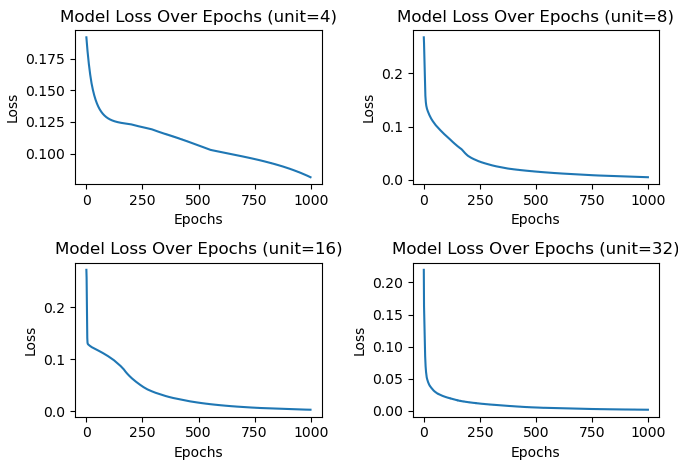
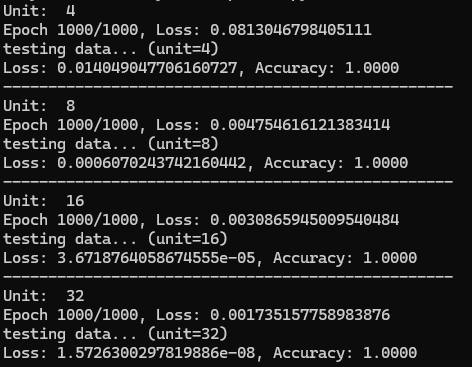


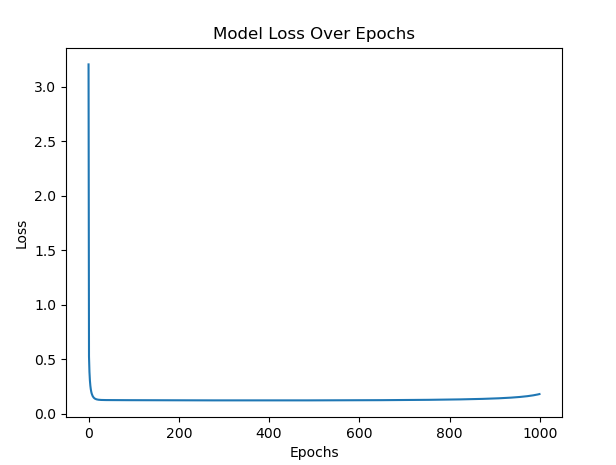
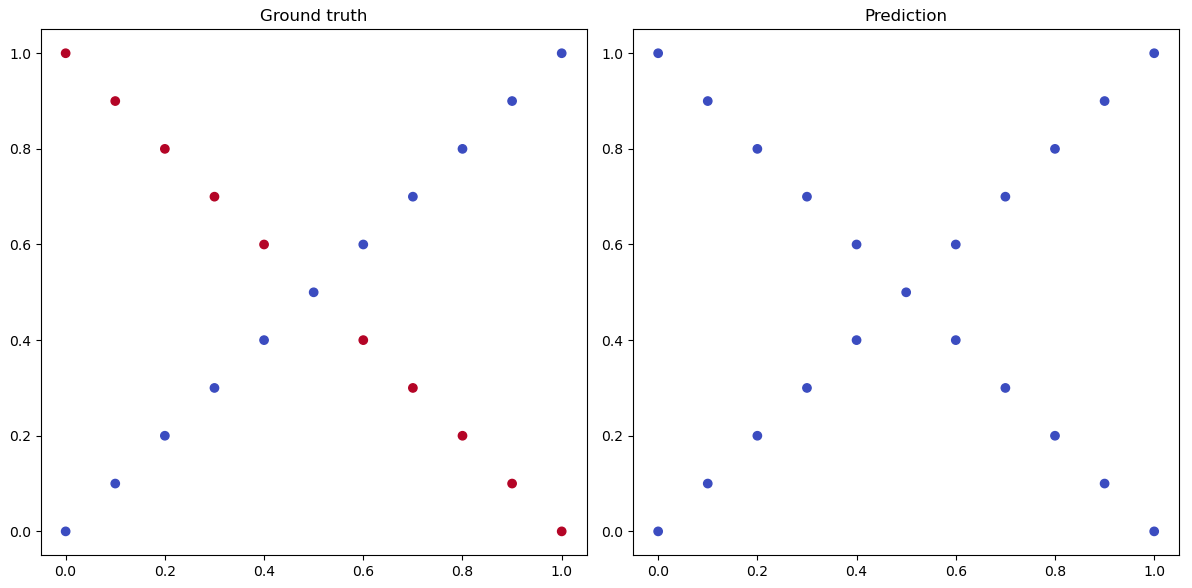
1. Experimental result
   1. Linear



* 1. xor



1. Discussion
   1. different learning rate
   2. different hidden units
   3. without activation function





1. Questions
   1. 為了要讓模型能夠做非線性對應。如果沒有activation function，不管模型架了幾層layer，都和只架一層layer的模型相同。也因此我們常用的activation function都是非線性函數。
   2. Learning rate太大的話會讓模型參數無法更新收斂到loss function的低點(對資料及而言)，有可能會在低點附近震盪。而learning rate太小的話雖然可以讓模型更新收斂到loss function的低點，但模型參數更新速度緩慢，訓練時間需要很久。
   3. Weight和bias是可以讓input經過線性空間對應後(或者再經過activation function做非線性對應後)，使得資料能夠以簡單的線性分類器(lab 中以 y>0.5 當作分類器)做分類。