

# Deep Learning (Homework 1)

0753420 郭家瑄

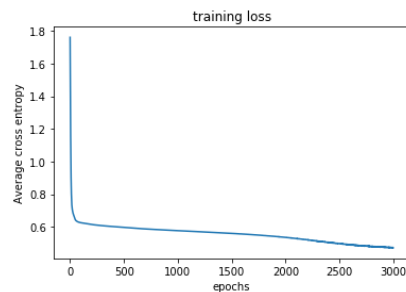
## 1. Variables setting :

- network architecture :
  - input layer dimension: 6
  - hidden layer 1 : 16 neurons
  - hidden layer 2 : 16 neurons
  - outputs layer dimension : 2
- learning rate : 0.001
- mini-batch size : 32
- number of iterations : 3000
- explanation :

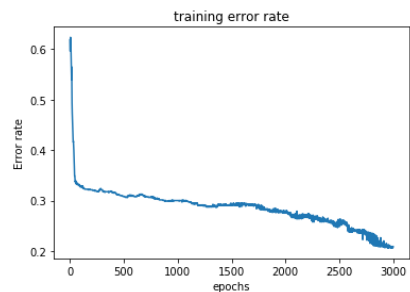
training data 的筆數共有 800 筆，其實沒有很多，所以我覺得 hidden layer 層數不用太多，參數量也不宜過大，所以只有設定兩層 hidden layer，並將 neuron 數量設成 16。

## Result :

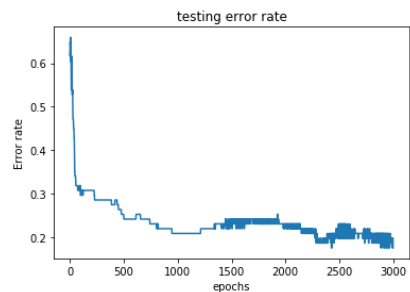
### ● Training loss



### ● Train error rate



### ● Test error rate

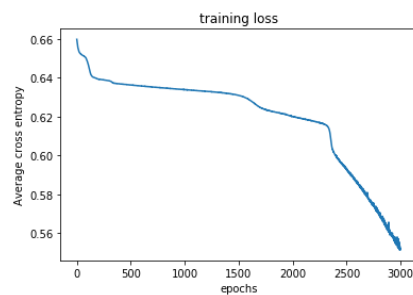


## 2. Variables setting :

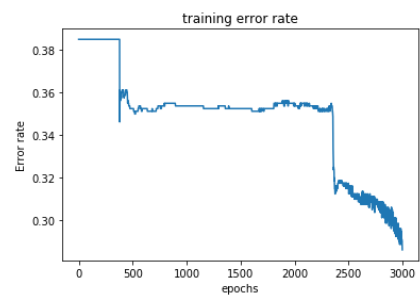
- network architecture : (fixed)
  - input layer dimension: 6
  - hidden layer 1 : 3 neurons
  - hidden layer 2 : 3 neurons
  - outputs layer dimension : 2
- learning rate : 0.01
- mini-batch size : 8
- number of iterations : 3000

## Result :

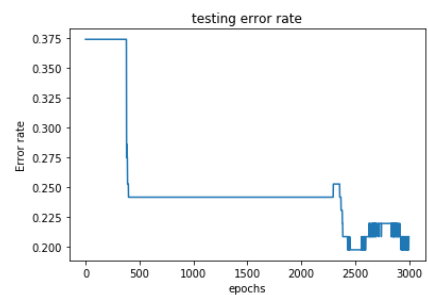
- Training loss



- Train error rate



- Test error rate



3. 我希望將 Fare 特徵壓縮至 0 到 1 之間，這樣跟其他特徵的量級才不會相差太多，因此這邊選擇用 Min-max normalization：

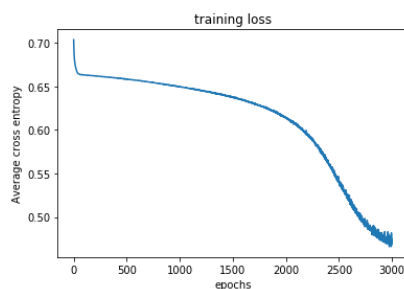
$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

#### Variables setting :

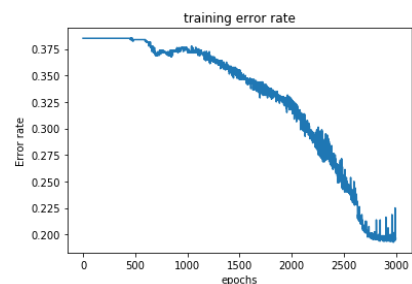
- network architecture : (fixed)
  - input layer dimension: 6
  - hidden layer 1 : 3 neurons
  - hidden layer 2 : 3 neurons
  - outputs layer dimension : 2
- learning rate : 0.01
- mini-batch size : 8
- number of iterations : 3000

#### Result :

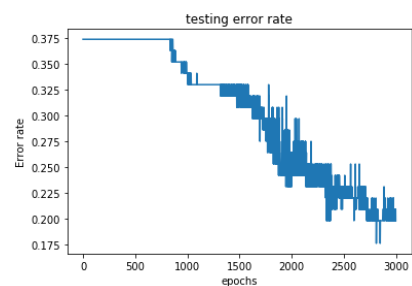
- Training loss



- Train error rate

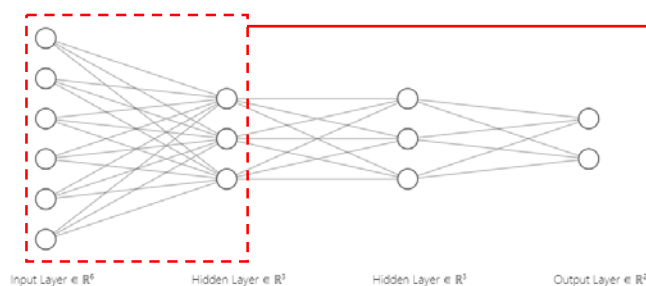


- Test error rate



除了 Fare 需要正規化之外，我認為還可以對 Age 這項特徵做正規化，因為 Age 的最大值是 80、最小值是 0，差距甚大，可能會造成 NN 在訓練時，對這項特徵的較敏感。

4. 我使用第 1 題 train 出來的 model，進行這一題的實驗，為了找出哪個特徵對預測的效果影響最大，我依序將每個 input feature 對應的第一層 hidden layer 的權重設為 0，檢視哪個 feature 權重變成 0 時，error rate 飆升最多者，就是影響預測結果大的特徵，這個概念的示意圖如下：



Original weight :

	0	1	2
0	0.257085	0.384541	0.193886
1	0.161624	-1.42941	-1.83907
2	-0.0778812	-0.994366	-0.532843
3	0.0790154	-1.00099	-0.306061
4	2.05439	-0.592606	-1.50022
5	-1.05934	0.105811	-0.340329

Set original weight to:

	0	1	2
0	0	0	0
1	0.161624	-1.42941	-1.83907
2	-0.0778812	-0.994366	-0.532843
3	0.0790154	-1.00099	-0.306061
4	2.05439	-0.592606	-1.50022
5	-1.05934	0.105811	-0.340329

每次只把一個 row 的權重設為 0，也就是一次只把一種特徵遮起來，看特徵有幾種就迭代幾次，用這種方式測試哪個特徵對預測結果影響較大。

### Result:

```
把Pclass對應的第一層model weight設為0:
train data error rate = 0.19374999999999998
, test data error rate = 0.20879120879120883
----
把Sex對應的第一層model weight設為0:
train data error rate = 0.49750000000000005
, test data error rate = 0.4285714285714286
----
把Age對應的第一層model weight設為0:
train data error rate = 0.1925
, test data error rate = 0.19780219780219777
----
把SibSp對應的第一層model weight設為0:
train data error rate = 0.22624999999999995
, test data error rate = 0.21978021978021978
----
把Parch對應的第一層model weight設為0:
train data error rate = 0.22124999999999995
, test data error rate = 0.19780219780219777
----
把Fare對應的第一層model weight設為0:
train data error rate = 0.385
, test data error rate = 0.37362637362637363
----
```

從結果分析得出 **Sex** 這個特徵是影響 model 預測結果最重要的一項特徵，因為若是 model 在轉換 Sex 的權重都變成 0 時，預測 train data 的 error rate 升高到 0.4975、test data 的 error rate 也升高到 0.4285。

除了 Sex 之外，第二重要的就是 **Fare** 這項特徵，因為若是轉換 Fare 的權重都變成 0 時，預測 train data 的 error rate 升高到 0.385、test data 的 error rate 也升高到 0.3736。

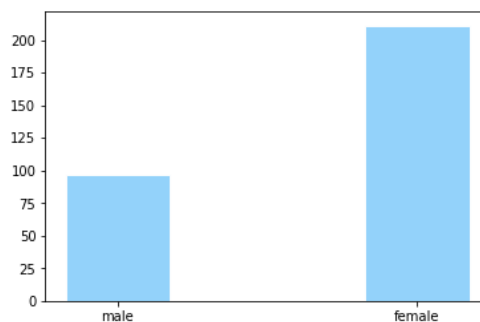
這邊用的 model，是沒有對 Fare 做正規化的 model。

5. 作業給的資料集中的 Pclass 是用數字 0~3 標示，這次的範例不管數字或是 one-hot 我覺得都可以，one-hot 的合理性在於，視每個類別為獨立、互相無交互關係的轉換，所以 pclass 進行 one-hot 的轉換未嘗不可。但我認為這邊就算直接用數字來當作特徵值也沒關係，因為若是 Ticket class 有分層級，有相對的優劣、等第關係，那數字的相對大小就會剛好呼應到，那不轉換成 one-hot 也沒關係。

6. 設計的兩筆 sample 如下：

#	pclass	sex	Age	sibsp	parch	fare	Expected survived	Model prediction
1	2	0	25	2	2	700	1	1
2	2	1	25	2	2	10	0	0

這邊利用第 4 題分析出的結果，也就是 Sex 跟 Fare 為變因設計，其他都固定相同，存活者的性別比例如下圖：



我們可以知道，若性別為女性，則存活機率會大幅提升，而 Fare 則是價錢越高者，其存活機率越高。所以設定一個樣本，為 female(0)、Fare(700)，猜測 model 應該會將這筆資料判斷 survived(1)，也就是會存活；另一個樣本設定成 male(1)、Fare(10)，猜測 model 應該會將這筆資料判斷 survived(0)，也就是無法存活，實際將樣本丟進 model 後，得到的答案如我所預期。