

## Assighment #0

### 1. 編譯結果

```
Sloth@LAPTOP-V9DEG1U3 /cygdrive/d/tmp
$ gcc assighment0.c -o assighment0
```

### 2. 執行結果

```
Sloth@LAPTOP-V9DEG1U3 /cygdrive/d/tmp
$ ./assighment0

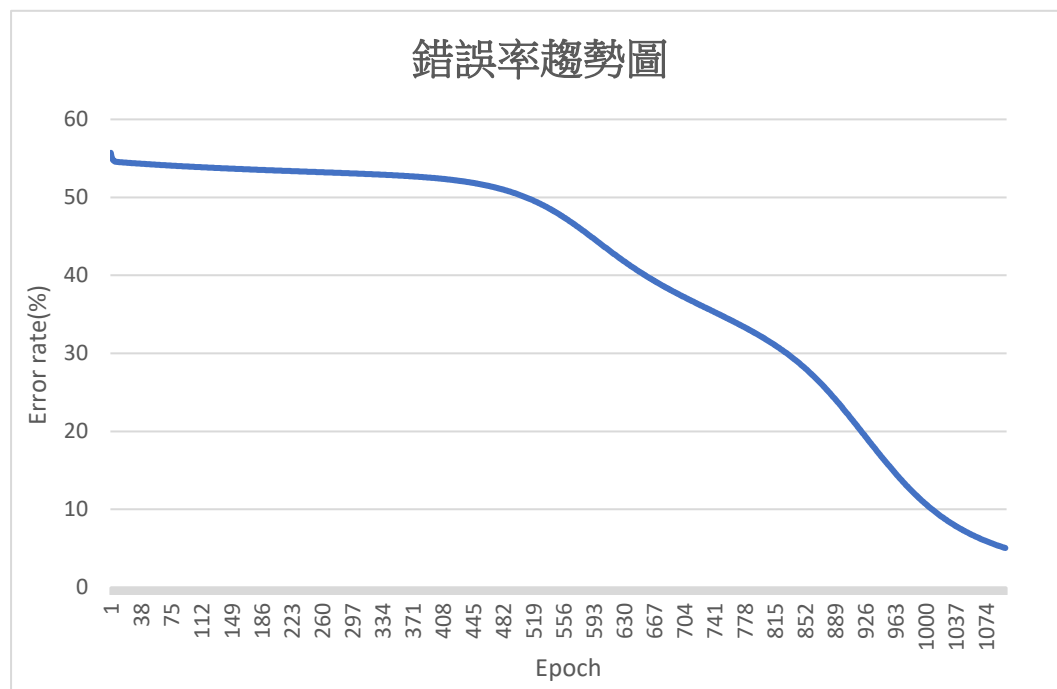
      Network Results

Pat   Input1      Input2      Targets      Outputs
0      0.000000    0.000000    0.000000    0.115932
1      0.000000    1.000000    1.000000    0.850441
2      1.000000    0.000000    1.000000    0.852348
3      1.000000    1.000000    0.000000    0.205880

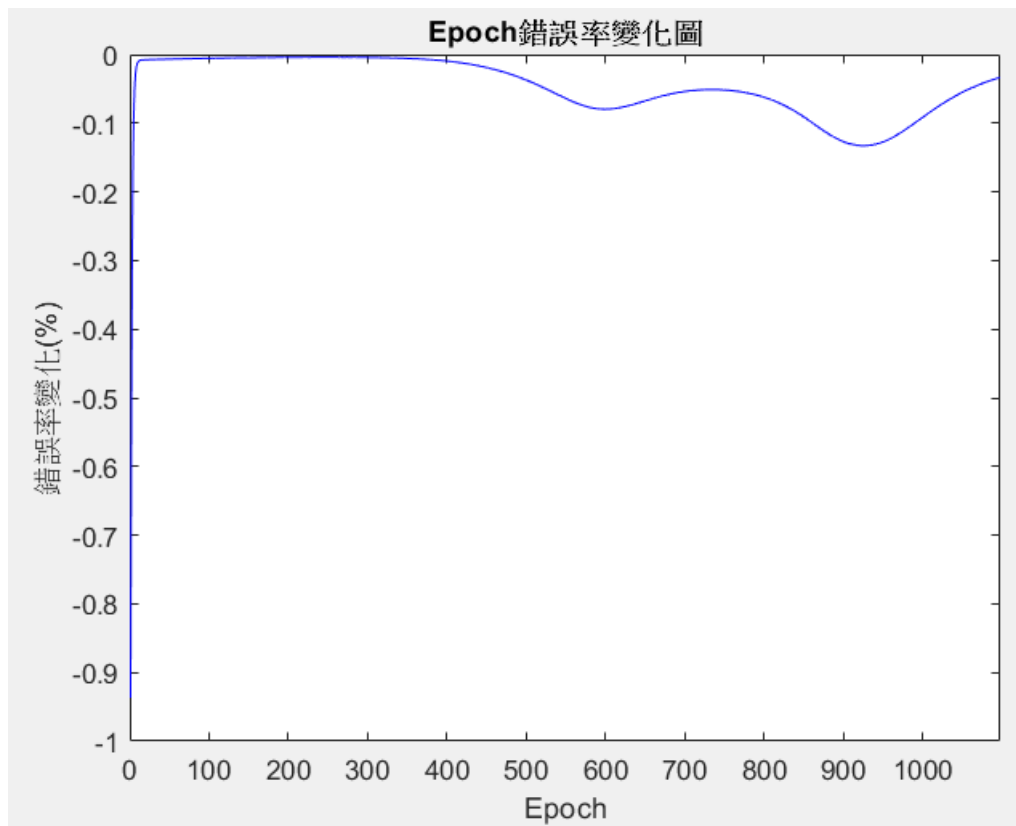
Accuracy = 100.000000%
```

### 3. 分析

利用這個類神經網路模型，可以完成 100%精確率的 XOR 運算。將 1099 個 Epoch(因為程式設定錯誤率低於 5%即停止訓練)完成後所對應的錯誤率利用 Excel 繪圖而成，發現經過越多次 Epoch，錯誤率越低，使機器達到學習的效果。



另外，我又將後面一個 epoch 完的錯誤率減掉前一次 epoch 完的錯誤率，透過 matlab 繪製成下圖，發現機器學習的過程最大的錯誤率變化在第一次到第二次之間。



#### 心得

起初我真的有點搞不懂 XOR 跟類神經網路的關係，後來查了很多很多資料才搞懂老師的作業意思，原本我是打算自己寫一個，但寫著寫著，發現自己有點沒辦法寫出權重的程式碼，結果就一直卡在研究權重設定，也搞不太清楚隱藏層的運算要使用哪種數學函式比較好，接著就發現自己沒時間了，迫於作業繳交時間，最後決定先看懂並引用網上已完成的程式碼，但我還是有點沒搞懂所謂的梯度下降法該如何分析，決定在下次再完成更完整的分析結果。