

3. 分析

現象一分析：隨著 training 次數越做越多，error 值愈來愈小！表示透過 propagation 運算，理論值與實驗值愈來愈接近，程式逐漸找到合適的 weighting 來得到正確的 XOR 結果。

【現象一補充解釋】：epoch 是 forward propagation 運作的次數，每做一百次，記錄一次 Target 與該次 Output 的 error 誤差值。當 error 小於 0.0004 時，運作停止，所得的 weighting 就是機器學習得到的結果。（如圖所示，此次學習運作次數=1620 次，0.0004 為自訂值）

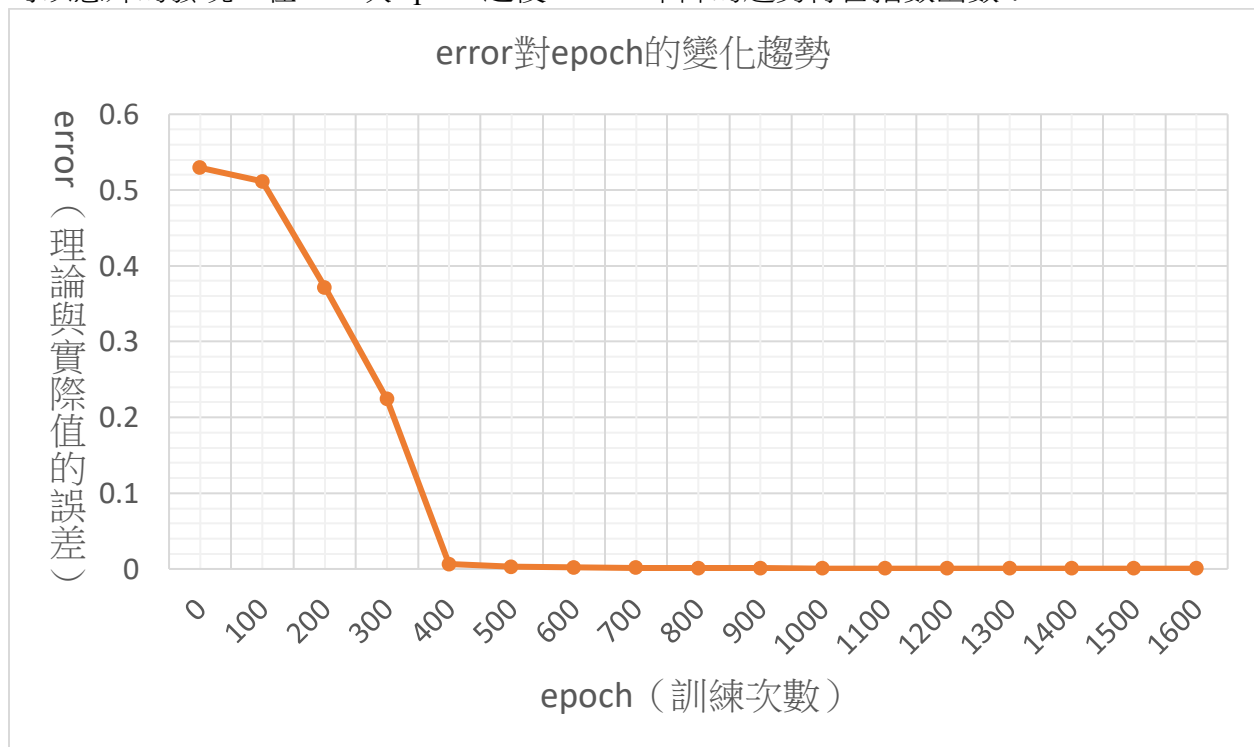
現象二分析：藉由最終學習到的 weighting 計算，對於系統用於 training 的四組輸入，可得到四個相對應的 Output1，分別和 Target1 相差（1.268%、1.342%、1.342%、1.668%）
(= (Output1-Target1) * 100)

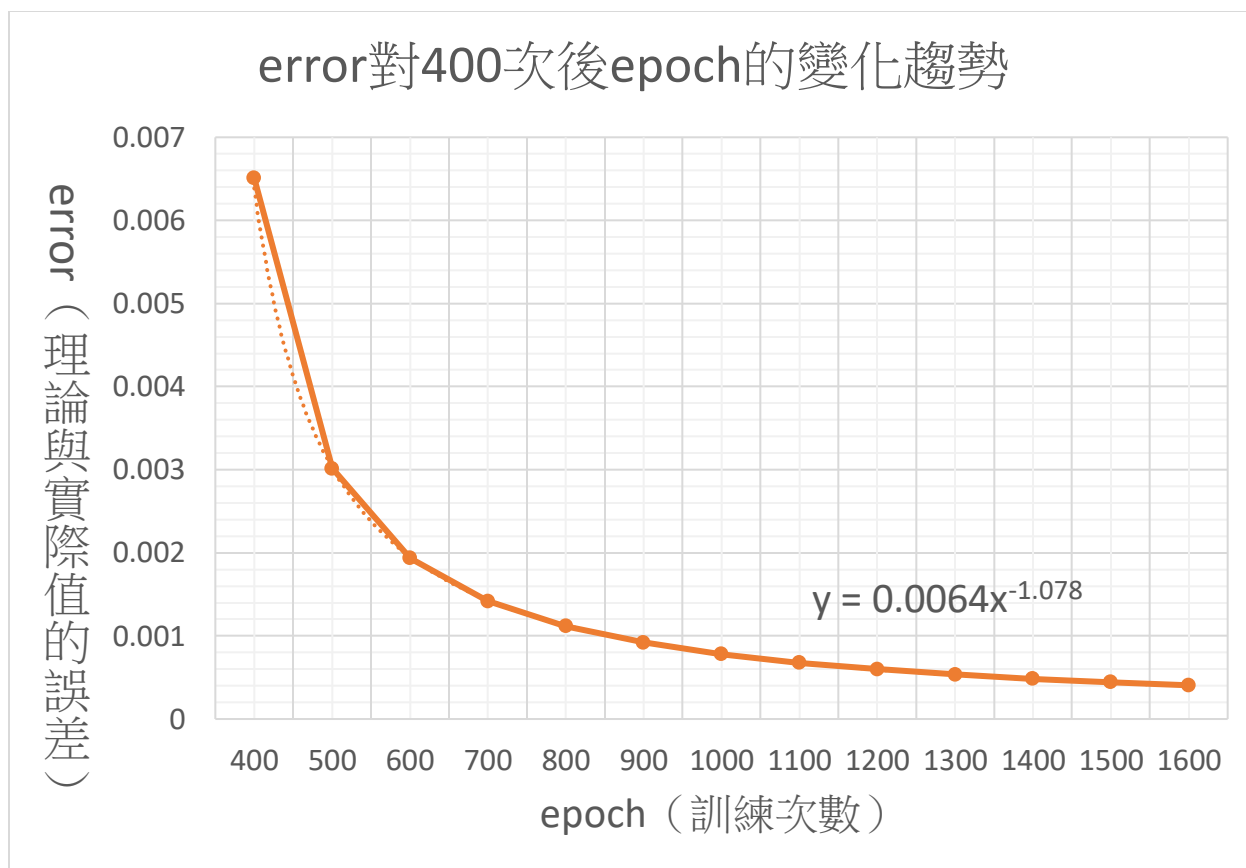
【現象二補充解釋】：Pat 表示第幾組 training data，Target1 表示四組理論值

現象三分析：接著，對於使用者輸入的任意四筆隨機數據（1,0 / 1,1 / 1,1 / 0,1），產生的 Output 值，與理論值（1 / 0 / 0 / 1）分別相差（1.341% / 1.668% / 1.668% / 1.3416%）

作圖：

【作圖補充解釋】：整體來說，error 持續下降。因為 error 值在 epoch 300 時，有一個明顯的 slope，因此，我將第 400 epoch 到 1600 epoch 的 error 拉出來做一個圖表。可以意外的發現，在 400 次 epoch 之後，error 下降的趨勢符合指數函數！





心得與遇到的問題：做完這次的作業覺得非常感動！雖然一開始看到 NN 題目時很害怕，躊躇了好幾天一直不敢去面對它，但是做完仍舊成就感滿滿！（因為我居然成功完成了簡易的機器學習！！）還好那時候沒有因為作業而退選～～不然就沒有這種成就感了！

這次作業碰到的最大問題，是理解 NN 的運作原理，weighting、propagation、sigmoid and loss function 在運作中扮演的角色為何，有了這些背景知識，我就能夠看懂老師提供的開源程式碼在幹嘛～

除此以外，次要的問題是，主程式與副程式間怎麼用指標傳遞矩陣，為此，我反覆上網觀看 Youtube 教學影片，才終於理解原理，並寫出自己的傳遞 code！

希望 10/3 的動態記憶體我也能像這次這樣順利解出～