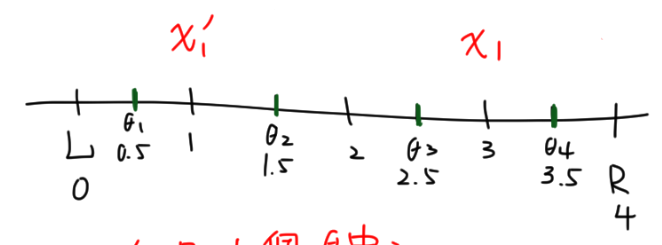
* More about Support Vector Machines
  1. kernel on decision stump

首先的部分因為都是同號，所以可以不用看：

所以就是要判別各個是還是。

由於的值只會是介於L跟R之間的整數，則是介於這些整數之間差0.5的值，所以我們可以觀察以下的例子：



假設這是某個在第1個維度的時候的情形，可以知道可能的有四個，並且根據此題目的特性，的個數可以由算出來。

在上面的模擬中，可以知道只有當比小的時候，以及比跟 大的時候，的結果才會是；介於 跟 之間時 的結果會是。一樣根據題目的特性，我們可以知道有 個比 跟 大，有 個 比 跟 小，有 個 介於 跟 之間，因此我們可以列出加起來的值為：

但是上面的例子一旦當在的左邊，也就是的時候，公式就會失效，必須要將 跟 在公式中的位置對調，或者說 的值取一個負號才會正確，所以公式如果要更為普遍，要改成絕對值：

此時再考慮有兩種方向，並且總共有個維度，所以要再乘以跟：

最後公式可以改成用一範數(one norm)來表示：

* 1. shift and scale the kernel function

根據題目所述：

列出「縮放的SVM」的對偶問題：

解完QP問題後可以得到這個問題的**。**這時我令，並且將上面的問題做一些修正：

並且將限制改為：

部分可以透過消除掉，所以最終得到：

這個新的SVM問題，我將他稱為「還原的SVM」問題。

而且我令「原本的SVM」問題為：

我們可以知道「還原的SVM」的解(的部分)，跟「原本的SVM」的解是一樣的，因為只差了這個正的常數倍。

這時我們來從還原出以及，根據講義的公式，還原b只要找其中一個自由支持向量(free SV) ：

當我們透過這個關係就可以得到「還原的SVM」的以及：

依舊是自由支持向量，因為

而「還原的SVM」的以及，跟「原本的SVM」的以及是一樣的，所以我們就可以知道「縮放的SVM」的以及，跟「原本的SVM」的以及是一樣的。

* Blending and Bagging

1. Upper bound of Error

對於每個的錯誤率，假設總共有筆資料，那麼犯錯的筆數就是。而在17個的投票過程當中，一筆資料要被分類錯誤，必須要17個中一半的人都犯錯，該筆資料才會是投票後犯錯的，也就是要個都在該筆資料犯錯。

而總共犯錯的筆數是：

可以知道我們最多可以弄出：

這麼多筆資料是投票之後犯錯的，所以可以知道最大可以是：

因此：

1. OOB Probability

這裡直接用講義(隨機森林第8頁)推導的結果：

所以可知：

* Adaptive Boosting and Gradient Boosting

1. AdaBoost can deal with “imbalanced” data immediately

首先我們可以知道：

並且在第一回合，由於AdaBoost algorithm吐了回來，所以可以知道有2%的資料是犯錯的，它們的；98%的資料是預測對的，所以首先可以算出錯誤率：

接著可以算出縮放係數：

所以我們可以將更新成：

所以我們可以算出：

跟原本的值相比：

減少了許多。

1. Some result AdaBoost

首先我們知道：

所以可以巧妙的發現：

所以只要一直往回推到，就可以知道：

1. Some result of gradient boosted decision tree

首先列出更新公式：

所以可以知道：

其實就是題目說的更新後的。這時候回顧計算的方式：

代進去：

Red correction：跟改後的題目為，而這點可以從上面的結論清楚的推論出來

* Neural Networks

1. gradient components

下面的表記法採用老師講義的格式：

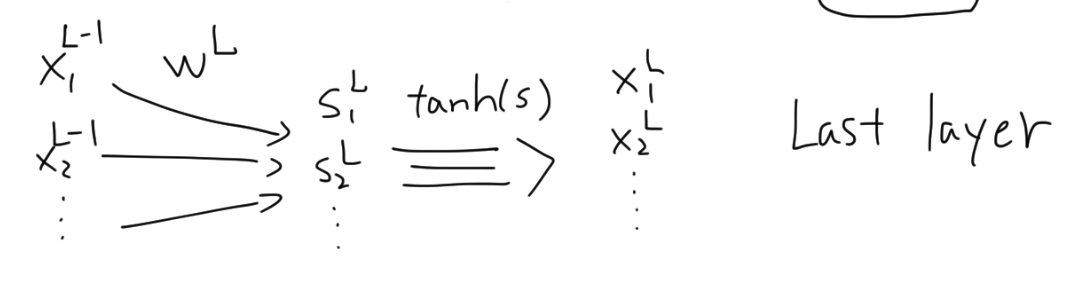
，配上第層的送往第層的的權重

，第層的第個*x*

，第層的第個*s*

先列出通往最後一層的權重的偏導數

圖示的部分如下：



因為題目有說output neuron 也有經過 tanh，所以才會有，並且這裡考慮的是更普遍的情況，輸出不只一個而是很多個。

會根據error function的不同而不同，例如講義是用均方誤差。

接著列出中間層的的權重的偏導數：

由於初始權重都是0，所以可以知道任何算出來的都會是0。

因此根據公式可以推得：

所以中間層的偏導數都會是0。

而最後一層的情形為：

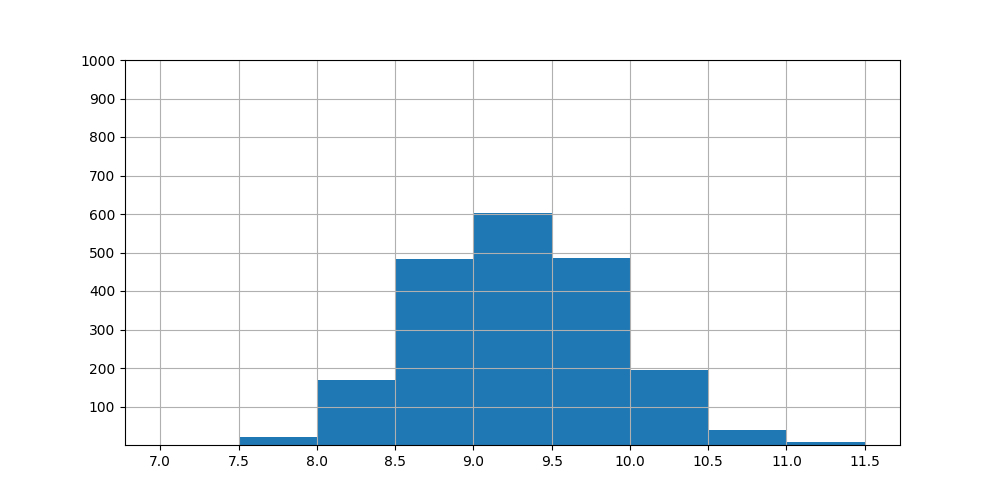
這時候情況會有些不一樣：對於來說，他是我們補上去的常數項，如果他是0的話，那麼就會是0。

對於其他的來說，他們是從算出來的，而

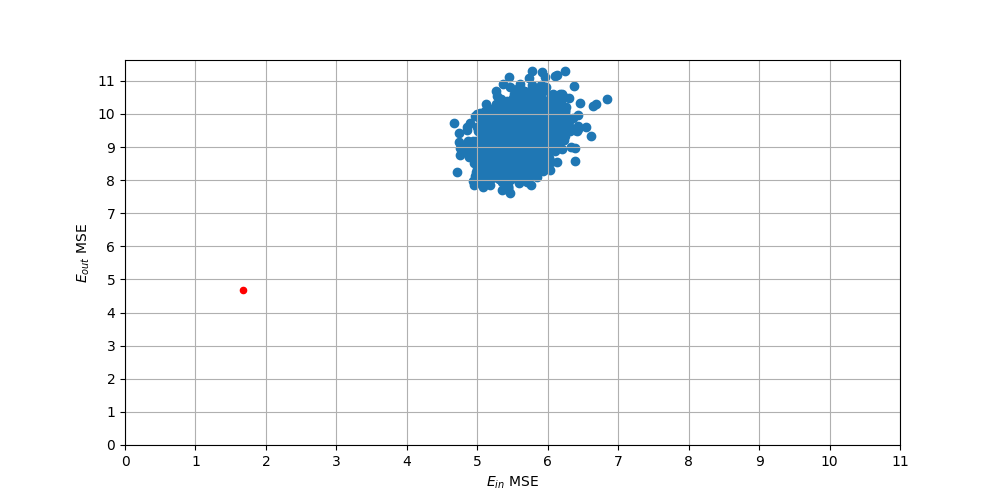
所以可以知道必定為0。因此可以知道只有是有可能不會是0的，其他都必定為0。

* Experiments with Decision Trees and Random Forests

1. of unpruned decision tree
2. of 2000 unpruned decision trees

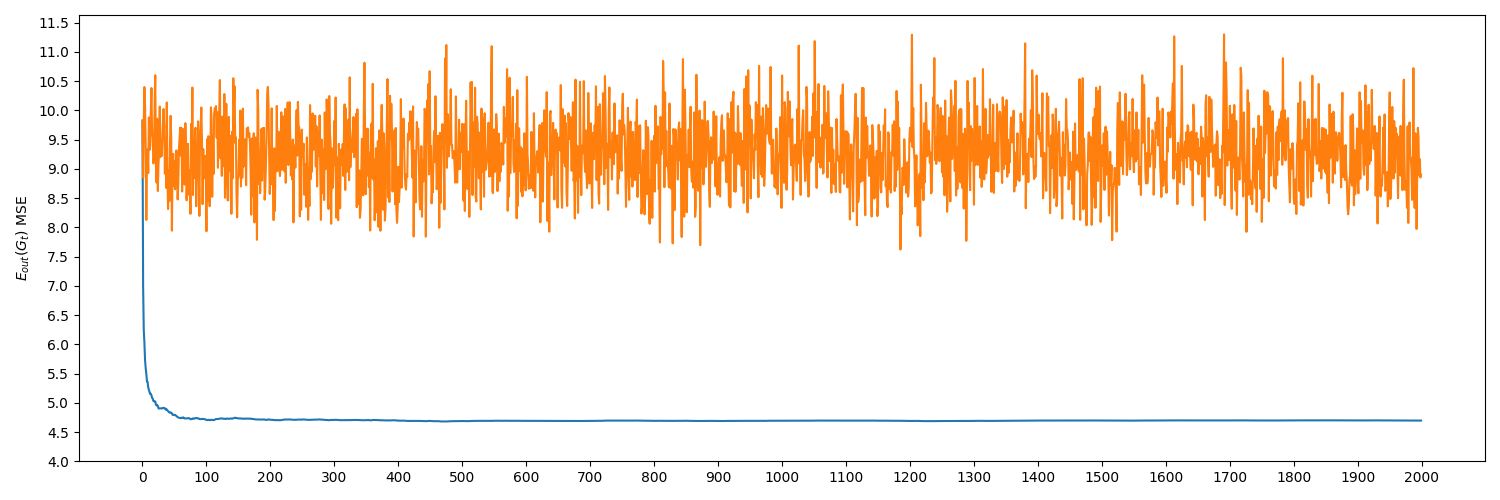


1. and scatter plot



紅色點是的所在位置，可以看到他離資料分布的位置有一段距離，並且是 Error 小的區域，這很好的顯示了voting所達到的降低錯誤效果，並且也不易受離群值的影響。

1. function



隨著參與投票的人越來越多，除了結果越趨於穩定，不易受離群值影響，錯誤率也跟著降低。不過可以發現錯誤率在大約100棵樹的時候就不再下降了，而那些有誤差的地方我認為應該是「尚未學到」的部分，因為都經過這麼多人投票決定了卻依舊有誤差。

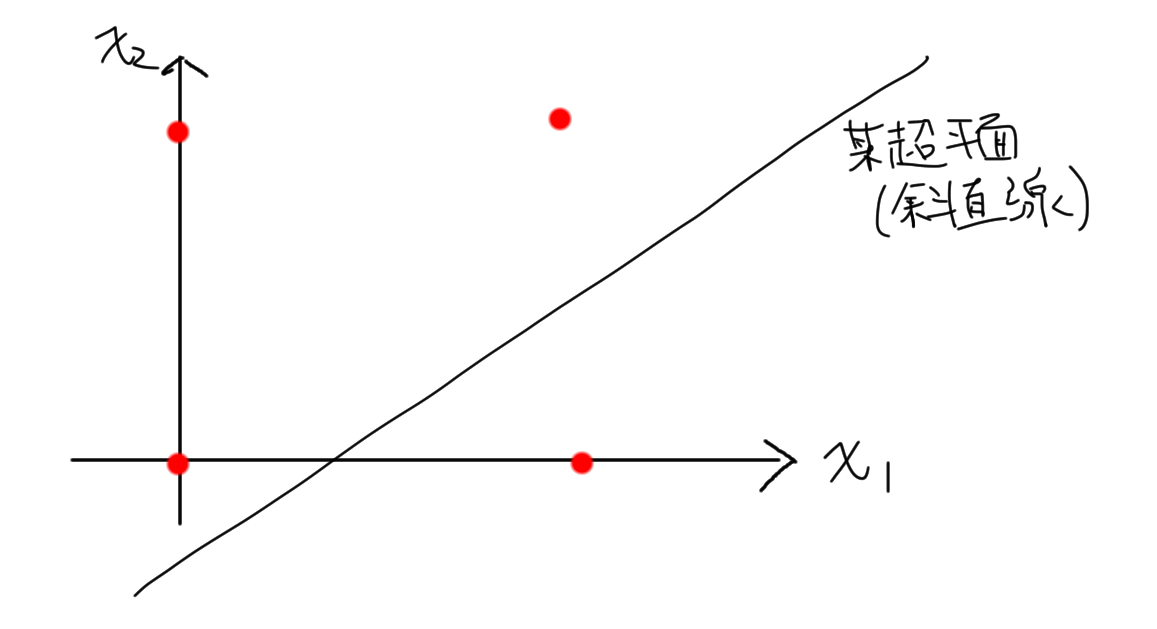
* Bonus

1. Crazy XOR

從這份作業發佈的那一天我就把前12題弄完了，全力對付這題，但是就算如此，經過這麼多天，甚至我使用了6張金牌延長3天的時限，同時伴隨著其他科期末的壓力，我依舊想不到這題要如何證明對所有的d是不可能的。

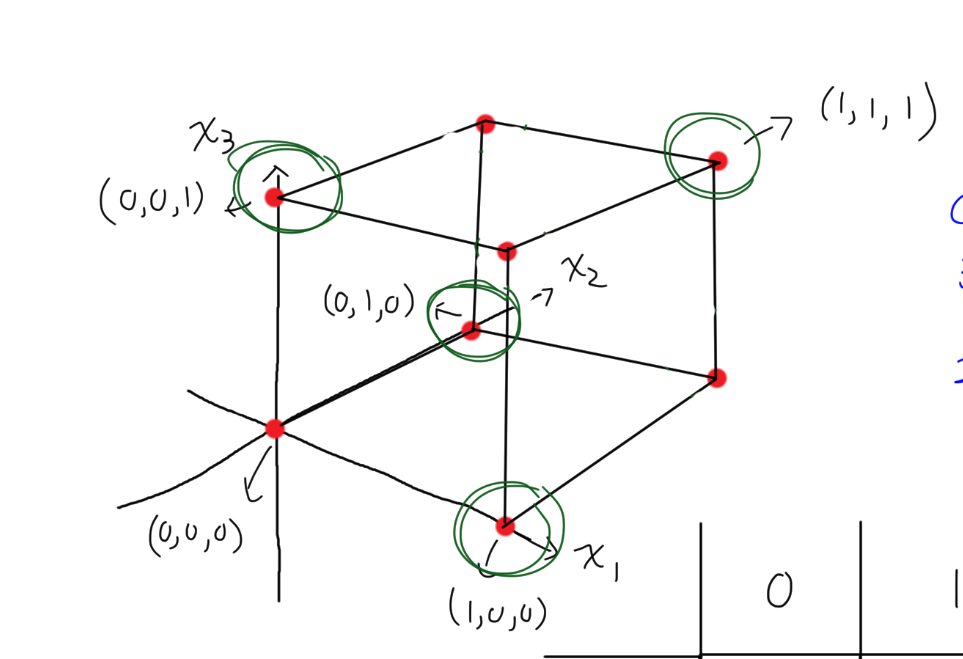
這裡我的設定是x只會是0或1。

目前我只能證明出的時候是不可能的，因為的時候可以看做是二維平面的4個點：



然而無論選了哪條斜直線，(1,0)或(0,1)這至少有一點就是無法被正確判定是。因此的情形可以用清楚的方式說明。

的時候其實也是可以用類似的手法進行說明：



但是也就僅止於了，再上去就超越人類極限了，必須要用更普遍的方法。

下面列舉我曾經的思路。

思路一：歸納法

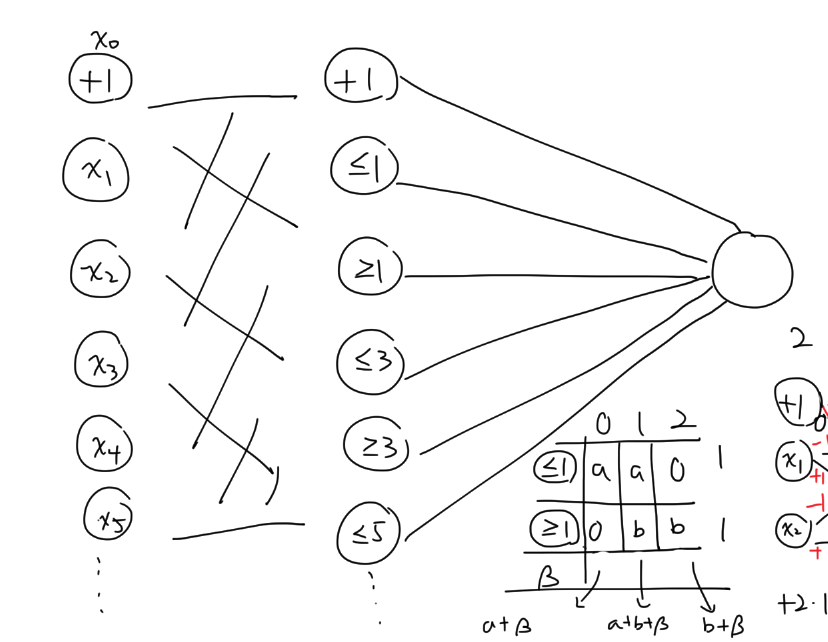
由於有上面的明確例子可以使用，所以我曾經想使用歸納法做證明，我的induction hypothesis是「網路不能正確的達成XOR的要求，也就是某些情形的輸出是錯誤的」。

但是會遇到許多問題：

就算我有了的情形，我很難推導出的時候也可以成立，很難從小的網路變成大的網路，因為會有新的輸入權重接上舊的神經元，舊的神經元會接上新的中間層神經元；亦或是這樣的方法，因為並沒有確定權重是多少，可能就會違反歸納法的要求了。

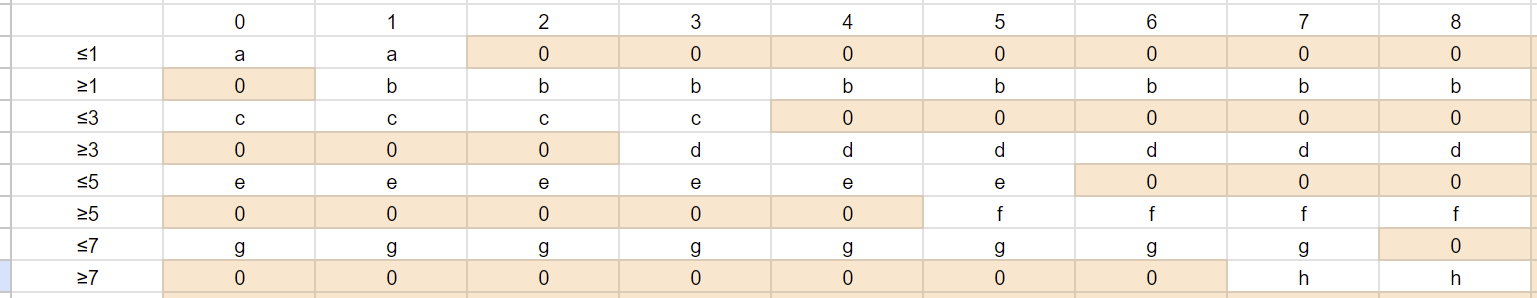
思路二：證明「解結構」的唯一性

由於我有已知確定形式的的「解結構」：



也就是中間層的各個神經元，或者說超平面，用途是判別當前1的個數是「小於等於1」、「大於等於1」、「小於等於3」、「大於等於3」…以此類推，如果是偶數的話最後一個就是「大於等於」，但如果是奇數的話，最後一個要是「大於等於」。

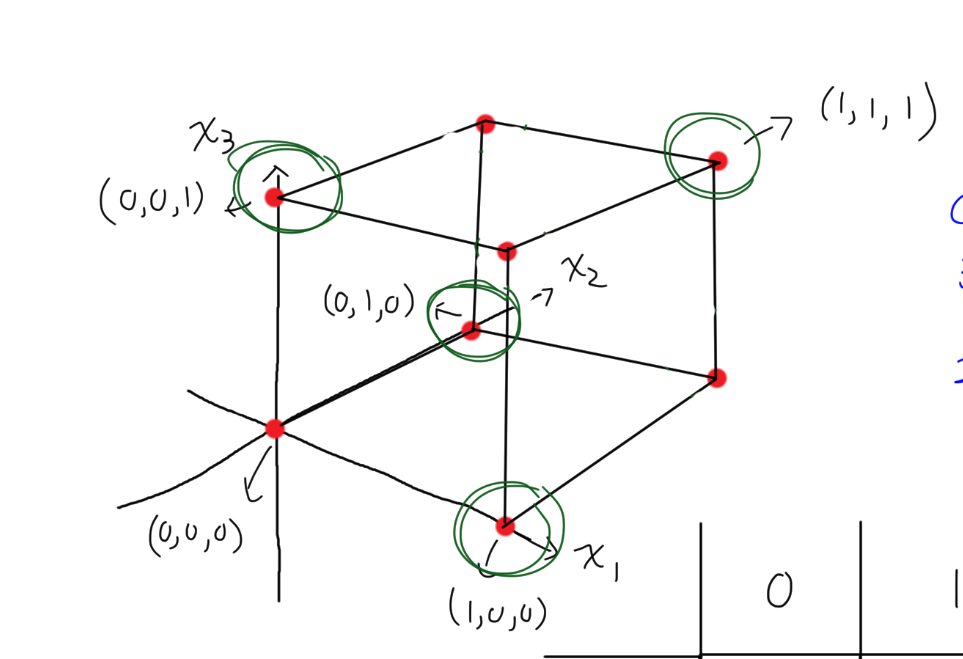
這樣的形式我有準確的配置權重方式，讓輸出滿足XOR的要求：



中間的字母a,b,…代表左方column的中間層神經元，如果輸出是1的話再乘上其前往輸出層的權重的結果，這裡我令由上到下的神經元權重分別是a,b,…以此類推。最上面的row的0,1,2,…代表當前的輸入有幾個1。可以看到如果輸入是奇數個1，那麼總共一定有5個中間層神經元輸出是1，如果是偶數個1，則會是4個；這個現象可以推廣出一個結論：偶數個1會有個神經元輸出是1，奇數個1則是有個神經元輸出是1。

從這個結論可以得知只要讓每個字母的值都是1，最後bias項的權重剛好是，這樣子就是個合法的XOR神經網路。

上面是我已知的合法結構，並且我可以根據對稱性，也就是經過旋轉或鏡射得到等價的結構，例如上面3維的結構：



我可以將整個圖繞軸順時針旋轉90度，並將這些旋轉後的點當作暫時該位置的點，然後再用這個點，搭配上面的網路架構，一樣會得到合法解，再轉回去後就是一個新的合法解了。

但是我不知道怎麼證明這是個唯一形式的「解架構」。

如果我確定了這是唯一形式的解架構，那麼我就可以說因為隨便砍掉其中一個中間神經元，就無法藉由調整bias的手段使得輸出滿足XOR，因為會導致有些偶數1的輸入跟奇數1的輸入，經過中間神經元後的輸出加總是一樣的。

思路三：另一種角度

如果有個中間層神經元，所有的輸入，他的輸出都一樣，那麼它的作用其實就跟bias項是一樣的，並且因此可以刪除。所以我有嘗試去證明對於網路來說，所有中間層的神經元，對所有的輸入其輸出一定不會都一樣。但是要證明這件事我一樣想不到orz。

以上是心路歷程，如果有個苦勞分我會感到很欣慰：\_ )。這題的解法還懇請助教到時標記在gradescope一下，真的很好奇是怎麼證明的。