* Learning Problems

1. 自我監督式學習舉例：

如同老師上課提到的應用，在自然語言處理中自我監督學習幾乎是第一個階段；會先在目前所擁有的資料文本中挖空格，讓機器去根據上下文學會某個空格應該要填入甚麼單詞；或是去學會如果出現某個句子，之後應該要生成出甚麼詞或句子。經由這樣的「熱身動作」，機器好像對這些文本資料有了某種「背景知識」，從這樣的訓練中學會了詞與句子之間的某種關聯，讓機器在做下一階段的訓練時可以有更好的表現。

1. ML for shortest path of a maze

我認為如果是對於那種有著固定牆壁和終點的迷宮，可以用已知的演算法找出最佳路徑，像是使用廣度優先搜尋可以找到最短的路徑；或者有可能路徑具有權重，可能可以使用Dijkstra's algorithm來找到抵達終點的最短路徑，不需要使用機器學習中的技巧，像是ChatGPT在題目中所提到的強化學習Q-Learning或是深度強化學習Deep Q-Networks來去學會找到最佳路徑。但如果是像移動迷宮或其他奇奇怪怪的迷宮，可能牆壁依照某種人類很難觀察出來的規則移動，或是地板以某種神奇難以推斷出的規則產生斷路陷阱無法前進，可能真的會如同ChatGPT所說的，機器學習可能可以帶給我們一些驚喜。

1. Machine Learning speed up any off-the-shelf algorithm

我覺得ChatGPT會做出這樣的回答，除了因為他的資料是基於2021年9月之前，所以不會知道最近發生的事情之外，另一個原因是因為他所獲得的知識根據大量的文本而來的，也就是說如果某個知識的文本量越多，他就學的越精確；相對的，對於文本量少的知識，他對該知識的理解就可能非常少或不正確。Google DeepMind 所找到的新的更好的算法，除了時間是在近期這個因素之外，我覺得另一個原因是ChatGPT根據大文本量所學到的知識是在高階層語言的層面，而不是在組合語言的層面。

* Perceptron Learning Algorithm

1. 根據PLA的更新公式：

由於 一開始是0，而經過了 跟 次分別發生在 跟 的修正後，由於都定為，所以可以根據更新公式得知：

1. 首先根據老師的講義中最後推導出的關係式：

其中，，是修正次數。

題目有說明只有當垃圾字(spam-like words)的出現次數大於正常字(less spam-like words)的時候，才會判斷是垃圾郵件，讓輸出，而且透過這個閾值項，當垃圾字跟正常字的數量一樣時，sign函數裡面的值不會是0而是；並且從那個上天知道的神祕當中，我們可以發現他給垃圾字的權重都是，正常字的權重都是。

下面我們來嘗試建構出可以完美分類的權重：

1. 首先我們可以發現不可以為0。如果為0的話對於完全沒有字出現的郵件，算出來會是0，代表該點落在該超平面上；但是題目有說我們可以完美的分開這些資料，所以不可以為0。
2. 令所有垃圾字的權重都是，所有正常字的權重都是， 是一個大於1的正整數。
   * 也就是說權重的絕對值要一樣，因為我們要計算的是字出現的「個數」，每種字的影響力是一樣的。
   * 是因為這個閾值項的權重只能是負整數，跟神祕的不一樣。
3. 是一個大於的負整數
   * 因為要讓沒有任何字、或是垃圾字數量等於正常字的時候sign函數裡面要小於0，不會判定是垃圾郵件。
   * 同時若郵件中就只有一個字，且是垃圾字的時候、或者垃圾字比正常字數量多一個的時候，sign函數要大於0，因此不可以等於。
   * 下面令。

統整後可以發現，PLA得到的最佳形式如下：

其中代表可能是兩者中的其中一個：總共會有個，總共會有個；是我們的值；只要最後的權重符合這個形式，他就跟一樣可以完美正確的分出各種郵件到底是垃圾郵件還是一般郵件。

有了這些代號，根據開頭的關係式還有題目的定義，可以推得：

其中最大的因為裡面最多就個字，所以會有個1，長度平方就是；最小的就是信裡面最多0個字，所以的值就只剩下，而長度平方根據上面的結論可以知道有個和一個，整合起來後就變成了上面的公式。因此如果是以上面的方式建構，得到的Bound會是：

至於題目的Bound仔細一看可以發現，其實就是代入。

所以可以知道不是真正的Bound；然而上面的建構方式也未必是最general的，因為權重或許可以有點花樣，建構方式也會不同，得到的Bound要考慮的因素就更多了。

1. 一樣先列出那個厲害的關係式：

令跟得到的關係式如下：

由於跟只差在一個是一個是，可以額外發現：

但重點是，根據公式，對這兩種資料跑的PLA，只要資料是線性可分的，最終都一定會停下來；也就是說最終停下來後得到的跟都是可以完美正確的分開「訓練資料」的權重，他們兩個在「訓練資料」內做的事情是一樣的，他們是等價的equivalent；但是跟很有可能終究還是兩個不一樣的權重，所以到了測試資料的環境，這兩個權重非常有可能會對一個資料的意見產生分歧，而做出不一樣的判斷，此時他們就不是等價的not equivalent。

1. 一樣先列出那個超好用的關係式：

這是尚未normalizing時的長相，此時，。

然後現在將資料做normalizing，全部的長度都是1，根據推導的過程，上面的關係式會修改成：

所以如果令 ，可知

因此上面的式子就可以改寫成：

這就是經過normalizing後的Bound。

1. PAM的特色是將更新的時候改成了當下面發生的時候：

也就是說，對於一個完美可以分開全部點並滿足「邊界寬度要求」的 可以知道：

所以可以知道：

也就是跟PLA一樣，每次的更新都好像讓更靠近。

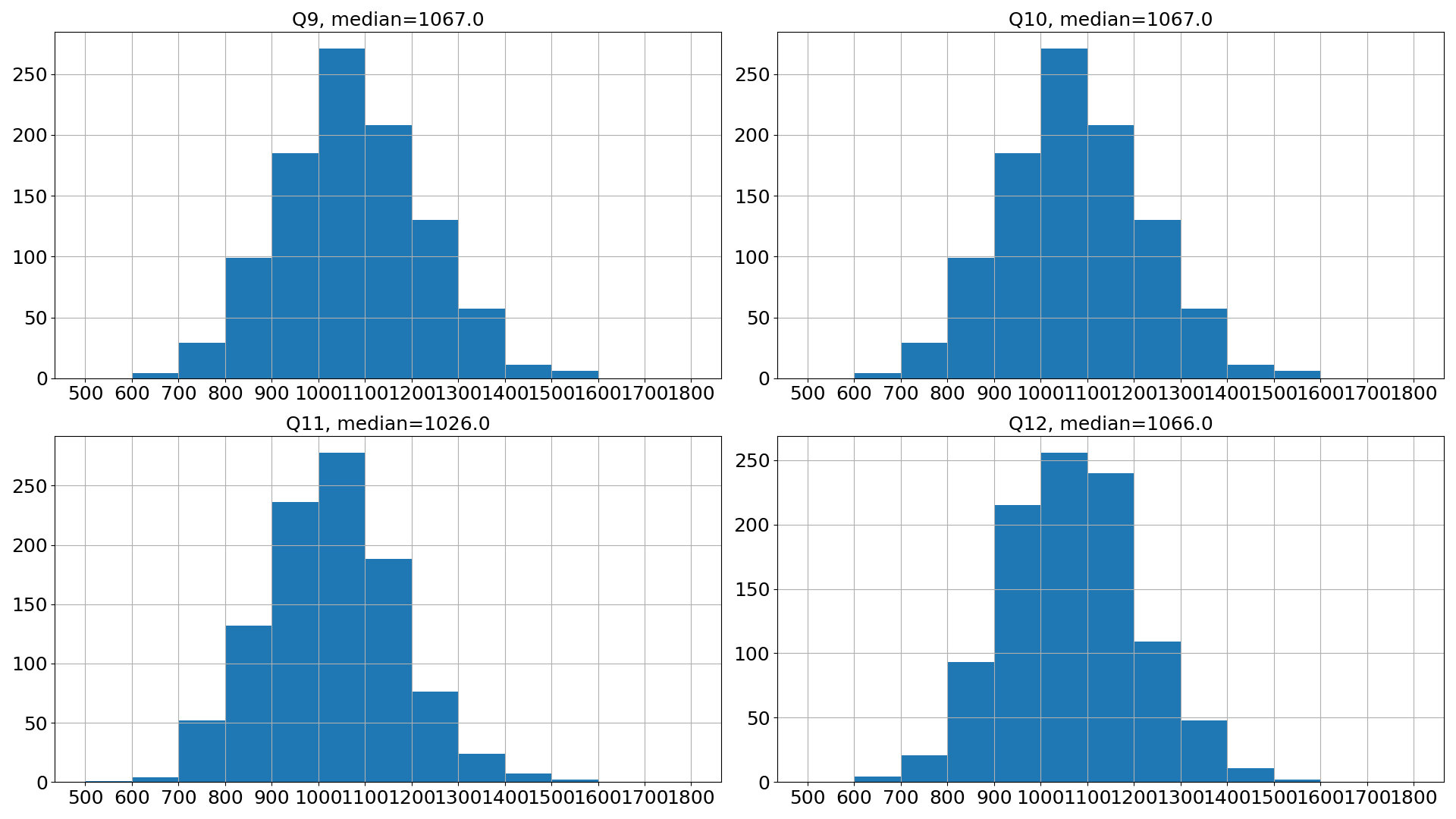
所以接著是嘗試弄出單位向量的形式，先推導出下列：

所以這部分也跟PLA一樣，每次的更新，長度的成長是有上限的。

所以只要把這兩個結論合併起來：

所以我們就跟PLA的過程一樣，成功說明了PMA也會停下來。

* Experiments with Perceptron Learning Algorithm



1. 中位數是1067。
2. 可以發現，中位數還有形狀分佈跟第9題的一模一樣。
3. 中位數是1026。實驗結果中可以發現，修正次數超過1100的數量比第9題的低，低於1100的則均比第9題的高。
4. 中位數是1066。跟第9題的圖形十分接近，但是可以發現變得更集中了，實驗結果中，位於900到1200區間內的實驗結果數量均高於第9題，反之以外的均低於第9題。

* Bonus

1. 在這裡我將speed up定義為，「最差的情形」能不能加速，所謂最差的情形是指，更新次數跟理論上限相差不會太多，上限的高與低會造成影響。

先列出原版PLA的最後推論結果：

接著是第7題的最後推論結果：

上面兩行當中的 是跟內積最小值的點的夾角。

可以發現，兩者主要差別在於未經過normalization的上限會受最大和最小的的長度所影響。

但是要注意的是，兩者的並沒有保證會一樣，也就是說，就算透過normalization將資料點長度的影響消除掉了，但是資料跟的關係是另一個影響因素，如果的值太小，也就是說跟最小的夾角近乎90度，那麼更新次數的上限就會非常大；反之，如果跟最小的夾角近乎0度，那麼上限就會非常小。

如果換個角度來看，分母的其實就是距離最佳超平面的最近距離，所以一樣只能知道normalization將分子的最大資料點長度消除掉，但並不能知道normalization會把離超平面最近的點的距離造成何種影響。

所以我不認為第7題的normalization可以加速「最差的情形」。