2022/10/02

清理一下 Colab 程式碼中的雜物；加上一些到專題過程中小簡報投影片的連結以供參考；另外也將最終報告中特別枝微末節的內容去除。
此文件原為共編文件，作為專題過程中記錄進度用，故順序為近期之紀錄置於前面。

01/20

佳勳：
結合第一次預測資料以及第二種資料格式訓練模型，結果比原先預期的好很多，詳情請見本周簡報

浩軒：
使用佳勳上周的模型，並做些修改試著降低誤差

允誠：
https://colab.research.google.com/drive/1rJ15P4locbVVmQJgsXI7B6cjxq8ncBzl?usp=sharing

1.
先把教授上週交代的完成：加入「successes#」feature，看看對應有牆壁阻擋的資料效果如何。
首先簡單檢查資料集內容，發現 successes#在一道牆壁的資料集也全部都是 7，只有在兩道牆壁時，整個資料集中有少數一些資料降低而已。加上前面的資料集太小，又都只有單一 AP 且單一角度，資料集本身鑑別度也低。
但仔細檢查，加入 successes#確實決定係數還是有提高一點，想必就是少數那些 successes#降低的資料因為考慮了該 feature 校正得更好。
不過不管如何決定係數都超過 0.99，明顯不合理。

2.
接著整理 Gradient Descent 程式碼過程中，突發奇想順便改寫成 Python。結果改寫成 Python 之後執行時間大幅拉長，本來 Java 跑一秒內，Python 要跑幾十秒；本來 Java 跑幾秒鐘的測資，Python 不知道要跑到甚麼時候。明顯無法使用。
繼續突發奇想，開始找資料尋找解決方案，將 Java 直接打包成.jar 執行檔，接著用 Python 的「subprocess」library 直接 call Java。
最後直接完成教授放棄的原始目標：按照最初討論的架構，將校正結果串接到 Maximal Likelihood，完成整個定位系統，接著以 12/30 當周收集的最終大型資料集模擬定位系統的運作效果。
此部分在 colab 最後，最終定位的平均距離誤差大概在 2 公尺之內。
從收集資料到訓練模型，過程中還有許多可以改進的地方，前幾周紀錄皆有提及。
此部分為了模擬實際定位情況，且每一點到各 AP 都有大量資料，定位時只取 1 筆(其實是用到 9 筆，見 01/13 第 2.點)，因此使用隨機抽樣進行模擬，每次執行結果或有出入，但大致如此。
當然，此部分的抽樣僅限前一階段訓練校正模型中，分割資料集後的測試資料集部分。

3.
發現 colab 需要掛載相關資料集等等檔案，任何他人包含教授其實是無法直接執行的。將所有相關

檔案上傳或捷徑連結至 01/20(本週)資料夾中的「root」資料夾。
任何人只要能將 root 資料夾建立捷徑至自己的雲端硬碟，並執行 colab 中第一區塊，掛載自己的雲端硬碟，接下來便能正常執行。
即使遇到困難，只要能將 root 資料夾整個上傳到 colab，並修改第一區塊程式碼中的「root=」字串至相應位置，仍舊能執行。但此方法可能需要將其中的捷徑也全部取得檔案或資料夾並補上。

01/13

佳勳：
測試研究生以及老師建議的改善 LSTM 模型建議，詳情請見簡報

浩軒：
研究使用 RNN 建立模型，但因為其他課的期末都擠在這一週，花了比較多時間準備，還沒有取得很好的結論。

允誠：
https://colab.research.google.com/drive/1rJ15P4locbVVmQJgsXI7B6cjxq8ncBzl?usp=sharing

1.
跟佳勳討論他的 LSTM，雖然我不懂細節，但討論過程發現有些地方好像沒有完全善用 LSTM 的優勢(就我聽說的大概優勢)，給了他一些模糊的建議，但他說最近較忙碌沒時間試。

2.
我自己選擇以先前工控資安題目的方式，一樣是使用 Python 搭配 sklearn，這次選擇一樣是 ExtraTreesRegressor (之前是用 Classifier 但模型本身的原理是一樣的)，這次的問題主要的重點是 features 非常少，所以我選擇用所有可用的 features，但不包含目標 AP 的 ID。

甚至還另外自己生 feature：
從這次題目(校正 WiFi RTT 距離)的角度來說，之前收的資料每個檔案代表的是一個 AP 到一個固定點，也就是固定角度固定距離，持續連收一堆資料，有時序性。對於每一筆資料，往前往後連自己一共考慮 9 筆資料，將自己的 rssi 減去 9 筆資料 rssi 平均值，得到一個新的 feature「rssiD」。實際應用時，我需要連收 9 筆資料，耗時最多不到 3 秒。需要的資料跟 LSTM 有點像，只是用法不同：我是前後資料的 rssi 來從這筆資料的 rssi 獲取更多參考價值。
這個 feature 的意義是知道：這筆資料的 rssi 跟其他和它一樣真實距離的資料的 rssi 相比，到底是更嚴重還是更好。教授說 rssi 很不穩定參考價值低，那就加上這個，讓模型自己看看到底是甚麼樣的不穩定法。
另外把這個考慮的 window 從前後共 9 筆往外擴張的話，每次要定位的時候就要收更久的資料，但也能校正得更準。我一開始是以同個檔案，也就是持續收 30 秒共 100 多筆資料的 rssi 平均值來計算 rssiD，決定係數飆到 0.96 多，但後來想想不符合實際應用，通俗得說就是有點作弊。

關於我資料前處理的結果可以看這週資料夾的 out.csv。
至於資料集分割，之前工控資安題目時教授指示過希望每次分割的方式固定下來較有討論意義，所

以我是採用「按資料收集時序、每五筆資料前四筆 training 最後一筆 validation」的方式，分割成 4:1。

這麼多 features(當然也包含估計距離)全部丟進去 regressor 訓練，regressing 的目標就是真實距離，決定係數達到 0.936 以上之外，平均誤差也在 0.576 公尺以內。
少了任何一個 feature 決定係數都有降低，所以確實都有用。(雖然只靠估計距離一個 feature 也還在 0.9 以上)

3.
把散布圖畫出來也發現其中有些資料本身就收得很差，因為實驗收資料期間曾受到嚴重干擾，詳情可以往下回顧 12/30 我的文字紀錄最後部分。

4.
重點：第 4.點這個情況不只是針對我的模型，對任何要處理這個校正問題的模型，包含佳勳的 LSTM，都是一樣的。
結論就是回到之前會議中我提出的問題，真正正確的做法應該是對一台 AP 全方位無死角的收集資料，這才是這個模型需要的訓練資料。在下面 12/30 我的文字紀錄中最後也有稍微提到。
再進一步，同樣的訓練重複好幾台不一樣的 AP 之後，相信對任何"沒看過的"新 AP，這個模型也能有非常好的校正效果，不用繼續訓練。

之前教授好像說過希望這個系統訓練一次，之後可以到各個地方使用，不用再重複訓練。教授也說過不同 AP 即使相同機型也會有個體差異。
因此我試著把資料集分割的方式改成「一台 AP 的資料全部是 validation，其他四台是 training」，一樣是 4:1，差別在於這樣子訓練時模型只看到 4 台 AP，而 validation 時卻是以另一台模型從沒看過的 AP 來當標準，以此來模擬這個模型訓練完後對於新購買 AP 的校正能力。
就現在的結果來看，是勉強堪用，分別拉五台 AP 出來當新購買的 AP，誤差仍在 1 公尺左右。但這並非理論上真正的結果！！！

首先決定係數比起單純看平均誤差更加有參考價值，因為單純看誤差會受到場地規模影響，也就是 AP 跟目標點的距離拉長本來誤差就會拉大，反之亦然，並不客觀；而決定係數的公式考量了這個問題，更接近"是否精準"的意義。
我們可以看到除了第三台 AP 被當測試集之外，決定係數都在 0.8 以上，而第三台 AP 當測試集時決定係數只有 0.5 多，但它的平均誤差卻是最小的。
究其原因，我們當時實驗架設的場景是第三台 AP 在中間，其他四台在四個角落，且「懷疑是信號最穩定的地方」都面向中間。因此這個資料集中，其他四台都是讓模型學習到類似的這個角度，只有中間那台不一樣。
當其他四台其中一台被當成新購置 AP 時，仍舊有另外三台讓模型學習到需要的"知識"，但當第三台 AP 被當成新購置 AP 時，模型從這個資料集沒辦法學習到類似的情況，當然沒辦法對第三台 AP 有太好的效果，這是決定係數小(不准)的原因。
而相比於其他在角落的 AP，第三台在中心的 AP 的資料距離都較短，這是它當測試集的時候誤差反而最小的原因。

簡而言之因為這個資料集不夠全面，只是模擬出我們架設了一個應用此系統的定位場域，可以說以這個資料集來訓練的模型，也只會對這個場域、這個設置內的應用有良好的校正效果。若想如教授所說訓練一次高枕無憂放諸四海，那就如上第 3.點開頭所述要用我提的另外那種實驗收集訓練資料。
實驗場地架設詳情可以查看 12/30 的投影片；實驗過程可以往下回顧 12/30 我的文字記錄。

01/07

佳勳：
利用浩軒整理好的三維空間距離資料訓練模型

浩軒：
整理上週蒐集到的資料以方便佳勳訓練模型

允誠：
浩軒說我對於上周(12/30)取樣的 ground truth 的筆記他看不懂，因此我重新整理得更清楚，之所以會有許多問號的原因如上周文字報告(在下方)所述：我們選擇以地面上的磁磚紋路作為依據，盡量維持採樣點在 grid 上，然後不再重複不斷測量所有採樣點的真實 X 座標跟 Y 座標，X 軸及 Y 軸部分的 grid 視為理想方格。
之後拜託他用 Python 寫程式自動計算出各採樣點到各 AP 的真實距離(當時本來也想用雷射測距，但這種任意角度的距離加上用手持誤差極大，後來浩軒忽然想到測量並紀錄了 XYZ 真實座標後可以直接算出)。

12/30

浩軒：
蒐集三維空間 data，共取樣 32 個 target point，每點約 115 筆資料。

允誠：
因為基本上我原本負責的部分已經差不多了，所以我臨時決定幫忙參與實驗的部分。
但我到場(12/29 18:00)之後才得知本周佳勳仍舊在忙，並沒有參與。就結果來說，好險我突發奇想加入，因為這天所做的實驗，我不認為任何人可以獨自完成。我的意思是，撇開時間跟肢體限制之外，我不認為一個人類獨自的精神毅力能有這麼強大。
雖然浩軒可能會有點覺得正是我多管閒事又太龜毛才害他今天這麼慘，但這不是單純我個人的想法，學長們也認同這是最後終究必要的實驗。
浩軒辛苦了。

今天的實驗是最龐大的，我們進行了"grid sampling"(我不確定這個名稱是否貼切)：
簡單地說，在預定場域架設好了五台 AP，並且使用雷射精確定義好座標系統之後，XY 軸每隔 2 公尺取一個座標。雖然先前說過是三圍系統，但出於場地跟電源限制，高度基本上只有兩種 grid，因此出於時間及精力考量，我們選擇用"梅花座"的方式，所有點有包含兩種不同高度(Z 軸)，但每個 XY 取樣座標只有一種。

然而工作量仍舊龐大：對於每個座標，除了要收集五個 AP 的資料，期間要確保手機位置盡量保持一致之外，還要繼續使用雷射精確定位這些取樣座標在座標系中的 XYZ 作為 ground truth。最終我們選擇彈性方案：取樣時 XY 軸盡量保持都在平整的方格上，不再重複使用雷射精確紀錄所有點的 XY 軸，視為方格。只使用雷射精確紀錄各點的 Z 座標。
最後取樣共 4*8=32 個點，每點有到五台 AP 的 rtt 距離各 100 多筆，另有座標 ground truth，也可據其計算出到 AP 的距離 ground truth。我們希望教授可以允許之後分割此資料集進行定位系統的訓練及效能驗證，因為每次實驗 AP 位置及角度不可能完全一樣，更有其他諸多變因，並沒有一致性，因此就算系統完成後其實也難以實際模擬定位流程且不切實際。(實際應用當中 AP 的座標及整體抽象座標系統架設完畢後不再變動)

取樣過程中最辛苦的便是"保持手機位置"，由於教授之前提過，將手機放在墊高物上取樣會導致緊鄰墊高物產生更嚴重誤差，希望我們以人手持手機取樣，結果在低位置時，只能化身資工蛙人長期蹲踞且姿勢必須盡可能保持不動。一開始大部分是由浩軒負責持手機，而我以雷射進行各種測量以及從旁輔助浩軒盡量精確選擇取樣位置在 grid 上。我一開始顧及若浩軒辛苦蹲踞我站在旁邊他的心情，所以也陪著他蹲，但直到最後剩下少數幾個取樣點時，浩軒表示"撐不住了"由我接手低位置，我才發現前面我根本也只是作作樣子，總之就是腳非常疼痛。這是我認為今天的工作浩軒最為辛苦的重要原因之一。
雖然以上段落跟研究主題本身並沒有直接相關，但由於教授先前表示是以文件/報告等紀錄做為專題評分依據，而會議內容實際上不會記得太詳細，因此這次實驗特別辛苦的部分，特此紀錄。

最後還有幾點要向教授預先請示：
在我們實驗過程中，大約 21:00 前後忽然湧入大批人潮，工五本就是開放公共空間，人潮喧囂直到 24:00 左右才漸漸消散，甚至也不清楚 AP 是否有被動到。另外雖然人體可能並不會造成太大影響，但人潮在工五一樓的桌椅附近"聚集"，可能已嚴重干擾 AP 訊號，加上人潮持續，迫於時間限制我們無法暫停實驗。此為不可抗力，之後驗證得到的精準度可能會稍微受到衝擊。
第二是先前(12/16)提過佳勳的 LSTM 是以 AP 角度保持一致的情況來訓練及驗證，若要進入實際應用需要以各種不同角度的資料來訓練及驗證。而我先前沒有太關注實驗的部分，今天才發現這部分的資料收集貌似後來沒有進行(12/23)。且佳勳最近十分忙碌，因此最終最差的情形可能會是直接單純使用我撰寫的一小段以 Maximal Likelihood 啟發的 Gradient Descend 程式碼段落，若真是如此最終成果將更加大打折扣。
我們由 18:00 開始，進行完取樣已經凌晨一點多，雖然我隔日(12/30)上午沒有事，但浩軒表示必須早上九點以前起床，由我進行最後一點收尾工作：主要就是稍微拍攝實驗設備配置的幾張照片，以及收納實驗設備待 12/30 回實驗室歸還。但是至此仍未產生任何報告！浩軒必須去睡覺，然而我本已稍嫌低下的組織能力在差勁的精神條件下顯得更加力不從心，這是此次選擇以文字紀錄為主、簡報為輔的原因。教授撥冗閱畢感激不盡。

12/23

浩軒：
實驗於不同樓層收發訊號(變更 z 軸座標)，收集三邊定位用訓練資料

允誠：

已查明，如 12/16 會議中所推測，是小數浮點運算先後次序不同導致(因計算機精度問題故小數運算中「交換率」等數學定律有時會產生誤差)。

12/16

佳勳：
訓練完成 LSTM(記憶量：2 筆資料)，細節在當周報告中
輸入資料處理皆與上周相同

浩軒：
練習訓練 LSTM model

允誠：
已將程式碼改成為將目標與各 AP 距離(r1、r2…)視為後續不斷輸入(定位)的變數而非問題常數。參考「12 月 09 號報告」最後一頁。
但與先前推測不同，出現了不完全相同(本以為應「一模一樣」)的結果。雖不完全相同但精準度仍舊差不多，至於不同的原因有待查明。

12/09

佳勳：
1. 訓練完成 Encoder 以達到 feature extraction
2. 首先刪除頭尾誤差較大的輸入資料
3. 訓練時先將輸入資料每 8 個成為一組 feature
(每次取資料中最大 4 筆跟最小 4 筆，不重複取，不同距離的資料分開取)

浩軒：
1.實作牆面是否影響 RTT 距離的實驗
2.在排除牆面因素後再蒐集一次上禮拜圓半徑資料
3.圖像化資料

允誠：
教授要求我統整之前的產出做報告
(參考當日投影片) https://docs.google.com/presentation/d/17dGvc576FLEJsL4dueMCyFIrC7H-kP-1/edit?usp=sharing&ouid=101754165731219897087&rtpof=true&sd=true

12/02

佳勳：
訓練完成 Encoder 以達到 feature extraction

浩軒：

蒐集半徑為 10M 圓每 22.5 度 WiFi-RTT 誤差資料
篩選資料

允誠：
convergence
(參考當日投影片) https://docs.google.com/presentation/d/1rD3M4ifikmnTWy-knDZqag5CPKG5dsMY/edit?usp=sharing&ouid=101754165731219897087&rtpof=true&sd=true

11/17 :

佳勳：
1. 利用 WiFiRTT 以及 Nest 在工五館二樓實驗取得測試模型用資料
2. 篩選資料
3. 以 Google Colab 訓練 Autoencoder 模型

浩軒：與佳勳一起蒐集資料

允誠：最小平方誤差估計 Gradient Descent
1. 此方法原理與 maximum likelihood 相同，差別在於結果是離散的 grid 還是估計的實數座標。
2. 此類方法不適合及時動態定位應用，因為本質是 try and error 持續降低誤差，若要求速度則精準度下降甚至完全錯誤。

11/10：

佳勳：在工五館跟彥廷還有 Hanas 練習 WiFiRTT 實驗