**HW6**

311512064鄧書桓

**Part1:**

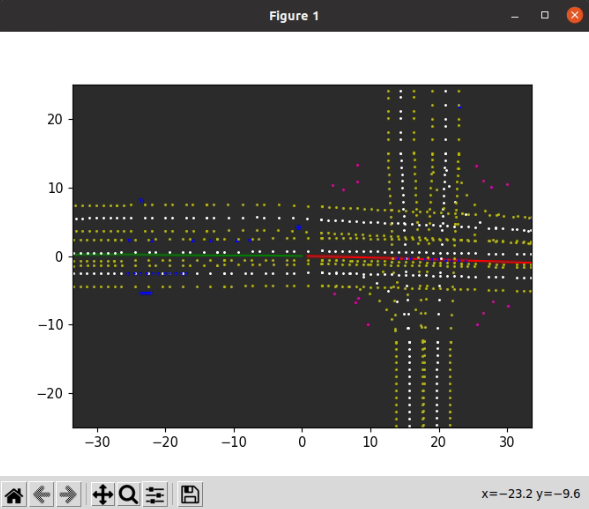
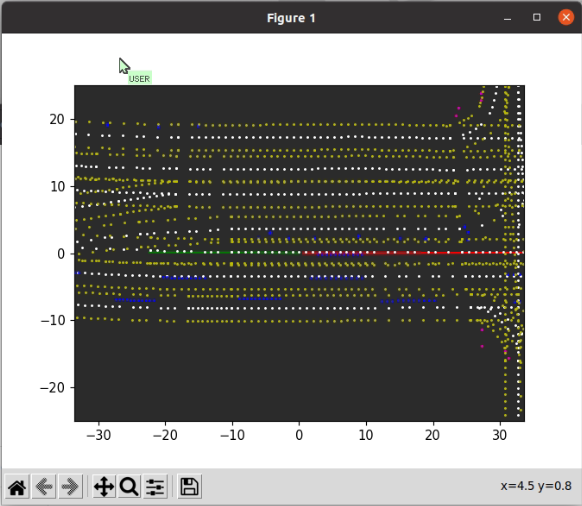
**dataset 視覺化的截圖:**

下面這段code為視覺化道路邊線、斑馬線與周圍物體軌跡(bonus)的部分，其中的那些維度大小是根據該變數.shape()去求得的。

一張含有 文字 的圖片

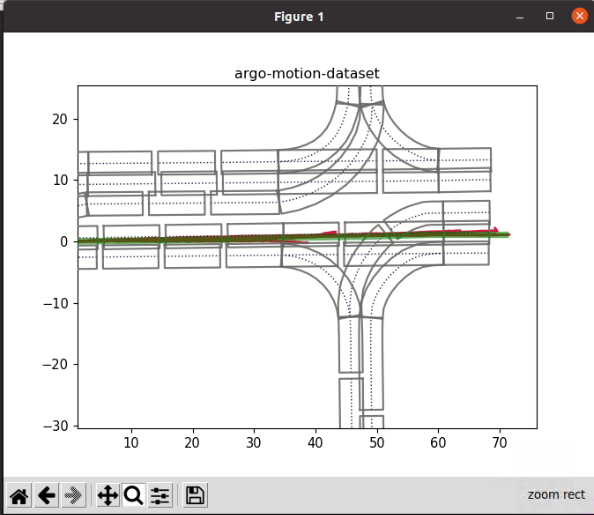
自動產生的描述

下方的figure1為可視化歷史軌跡、未來軌跡、道路中線、道路邊線、斑馬線與周圍物體軌跡的部分。另外，我本來想將檔案中的orig與rot一併顯示，但他們得值好像介於給定的x, y範圍外，所以無法顯示於matplotlib上。

** **

**Part2:**

**Argo軌跡預測截圖:**

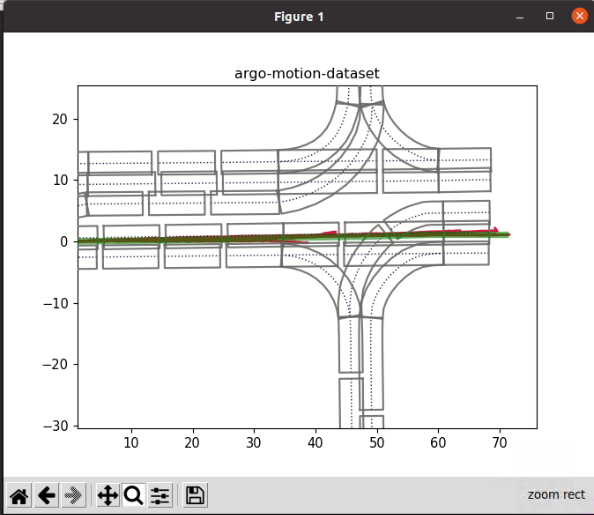


**Model 設計方法:**

由於本來的network train出來的效果就不錯，因此對於整體架構並沒有做過多更改，這邊僅在MLP隱藏層作些許調整，如更改其中幾個神經網路的大小，來使特徵擷取的效果變得更好。另外，我還有在multithread加入nn中的normalization以此來避免over fitting，雖然本來的network中已經有使用droupout來改善了，但是特殊的情境(如轉彎)時的overfitting還是較嚴重。最後，我有將epoch 改成150，以此來提高在轉彎處的預測成功率，經過多次測試可知，當epoch調整到85以上時在過彎處才會有一組預測是成功的。

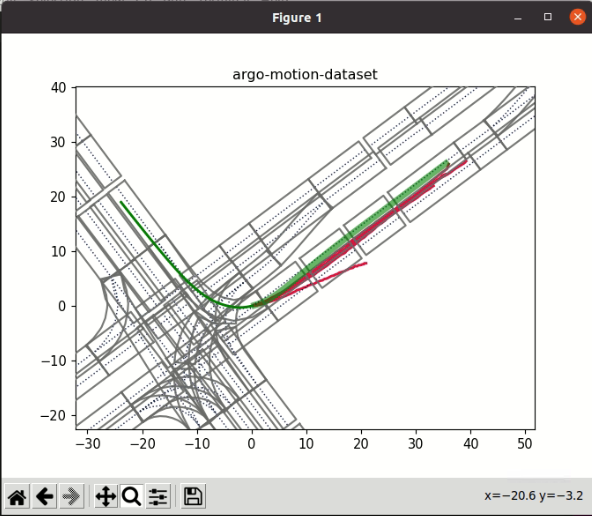
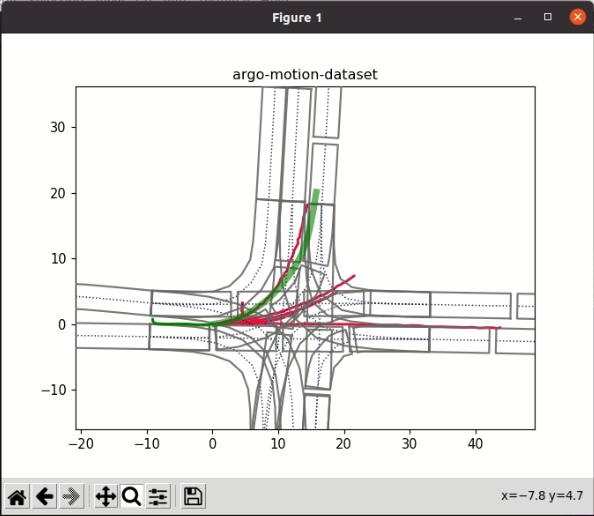
**Case study:**

1. **直線:**



在直線的預測中，原本的model就預測得非常好，6個預測軌跡幾乎吻合原先的ground truth，而其中有一些會插出去的原因我認為是overfitting造成的(這邊的epoch設為150)，會被環境中些許的極值資料影響，但若並非這樣設epoch的話，於後續彎道的預測將會慘不忍睹，因此並沒有再做調整。

1. **彎道:**



由左上截圖可知，在彎道時有5個預測的軌跡是跑偏掉的，僅有一個預測是成功的，我認為是因為彎道本身就比較難估計，而且我拿去train的資料與預測的資料僅有地圖與觀察5秒的資訊，對於彎道的資訊較少，無法藉由周遭的動態障礙物去協助判斷，因此只會有一個成功。另外，需要將epoch調整至85以上才會有一個預測的軌跡能夠符合ground truth，雖然這樣有達成本次的目標，但是因為訓練集與輸入的資料是在相同環境下，因此over fitting的負面影響較小，但這樣的泛化性就極差，在光復路或是其他場景時的預測效果就會很不好。而右上方的截圖顯示預測的效果非常好，我認為是這邊的地圖資訊較充足，因此並不會因為是彎道而預測失敗。

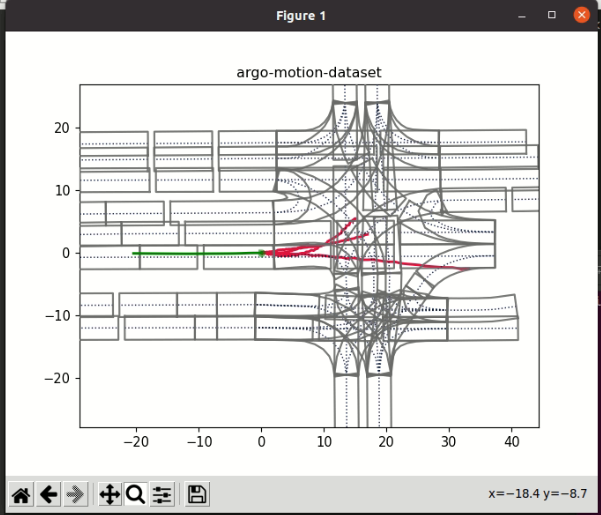
另外，對於上述overfitting的現象，我有在multithread加入nn中的normalization，以此將極值去除，使用的方式為下方第一個截圖。另外，我也有嘗試改變MLP中的通道數量，希望藉此能夠改善於彎道預測結果較差的問題，但效果沒有很好，改變的方式如下方第二章截圖。



一張含有 文字 的圖片

自動產生的描述

1. **急停**

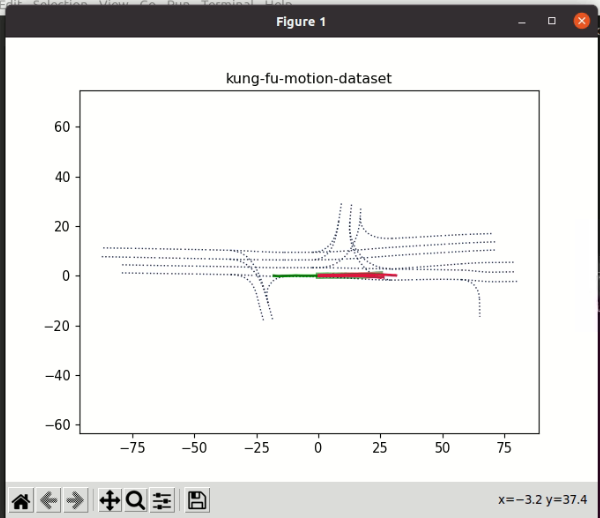


由上方的截圖可知，對於急煞的預測完全失敗，我認為是因為並沒有考量到周遭的動態物體，因此對於急停並沒有辦法預測出來。於此部分，我有設法將周遭動態資訊加進去做train，但是一直失敗。我認為只要將周遭動態物體的資訊也考慮進去，就能夠很好的完成急停的預測。

**Part3:**

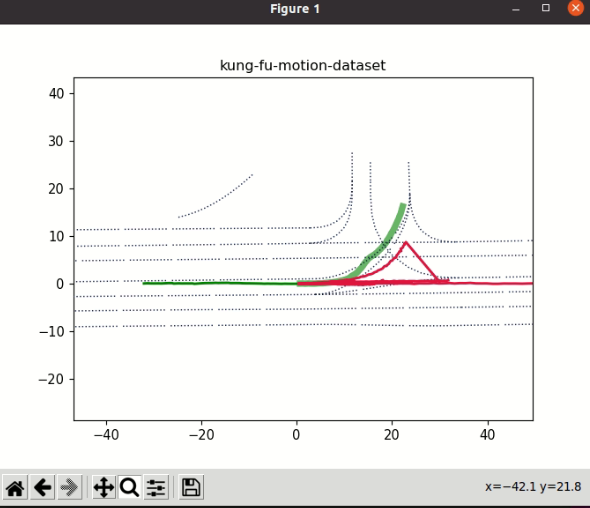
**光復路軌跡預測截圖**

**1.直線:**



由上方的圖可知，直線的預測非常完美，預測出來的軌跡與ground truth 十分吻合，因此對於此部分沒有做過多的調整。

**2.彎道:**

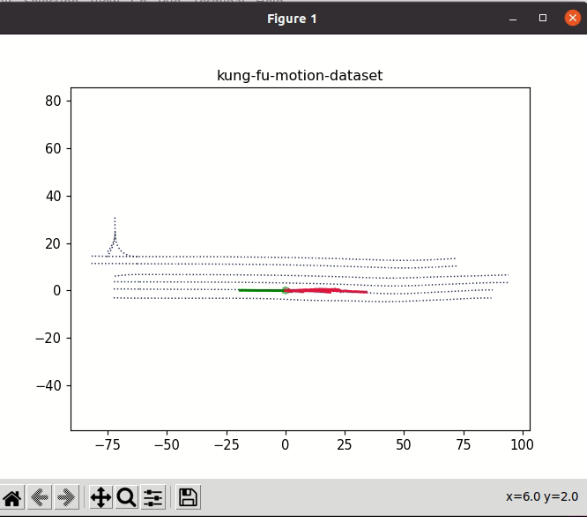


由上方的截圖可知，於彎道時的預測完全失敗，我認為是因為訓練出來的模型已經over fitting的關係，此model僅能作用在Argo環境中。我有設法加入normalization與 新增dropout層來避免overfitting，但是效果微乎其微(所用的Dropout與normalization在下方截圖所示)。我認為若要預測成功，應該要使用更能避免overfitting的方法，這樣可能要讀一下其他paper才有辦法成功。





**3.急煞**



這邊的裝況和在part2的狀況一模一樣，由於沒有將周遭動態資訊給考慮進去，才會造成此問題，我認為只要將周遭動態物體的資訊也考慮進去，就能夠很好的完成急停的預測。