HW1 房屋估價

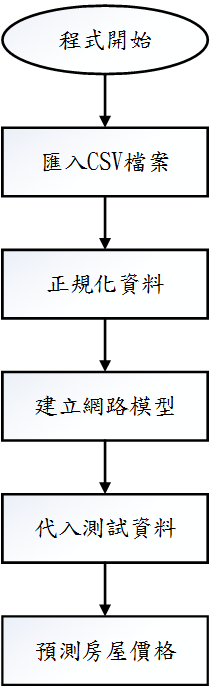
107368020 黃立

1. **作法說明：**

在Windows下建立TensorFlow與Keras模組，使用jupyter notebook來撰寫程式，程式是利用多層神經網路實現迴歸算法，計算房屋之差價。

1. **程式方塊圖與寫法：**

2.1 方塊圖：

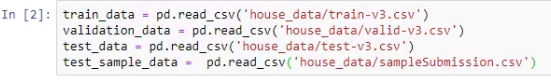


* 1. 程式寫法：

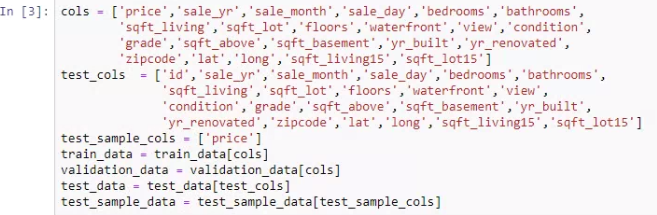
Step1：import所需要的函式庫



Step2：匯入csv檔(train-v3，validation-v3，test-v3)



Step3：選取訓練網路時所需要的欄位

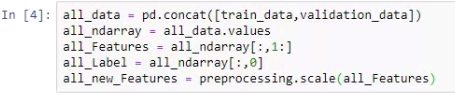


Step4：將train-v3和 validation-v3合併

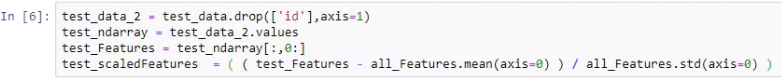
Step5：將表格型式轉成Array

Step6：分類特徵值和目標值

Step7：正規化特徵值的數值

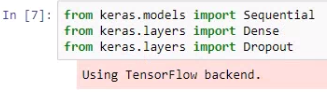






Step8：建立類神經網路模型

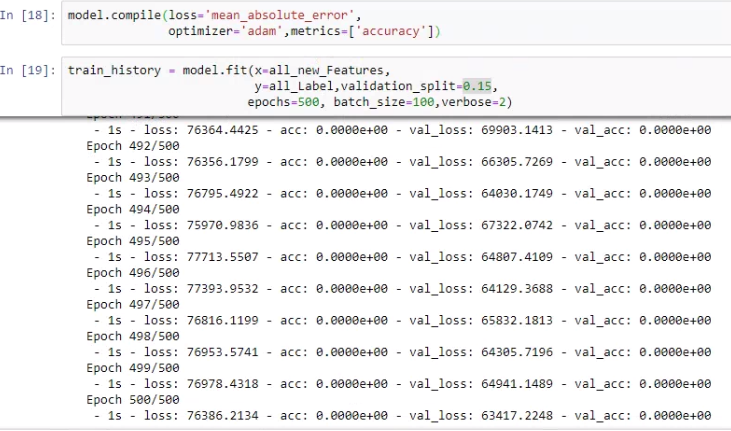
其網路設計是輸入層1層，隱藏層4層，輸出層1層，搭配drop\_out層以避免overfitting，權重初始化使用he\_normal分布，活化函數使用relu，輸出層使用linear迴歸分析房屋的價格



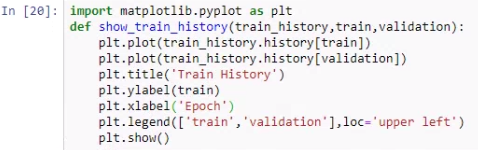


Step9：將訓練資料參數代入網路中

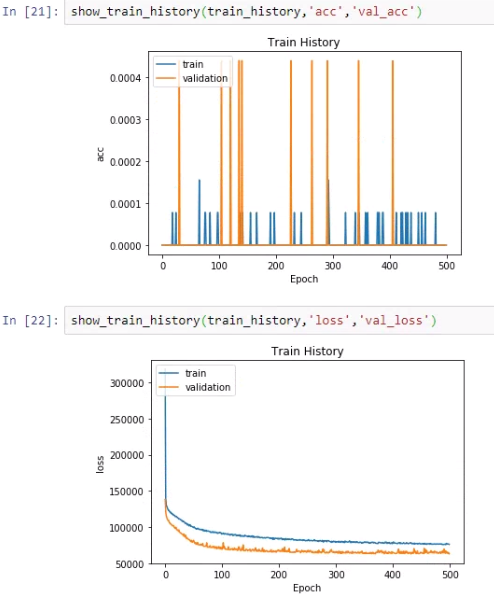
建立損失函數mean\_absolute\_error，優化器使用adam來修正權重與偏權值的誤差，並且設定網路參數，將訓練資料分15%為網路驗證所用，訓練週期為500，小批次學習設定100筆



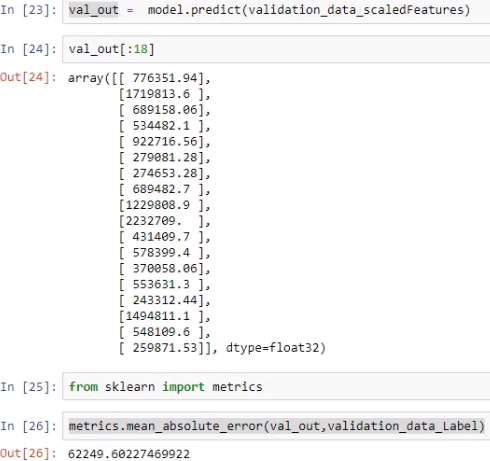
Step10：畫出損失函數與準確度的圖形評估網路訓 練之成效



Step11：將驗證資料代入網路層中，查看訓練效果



Step12：利用訓練好的網路模型將測試資料代入，預測其價格

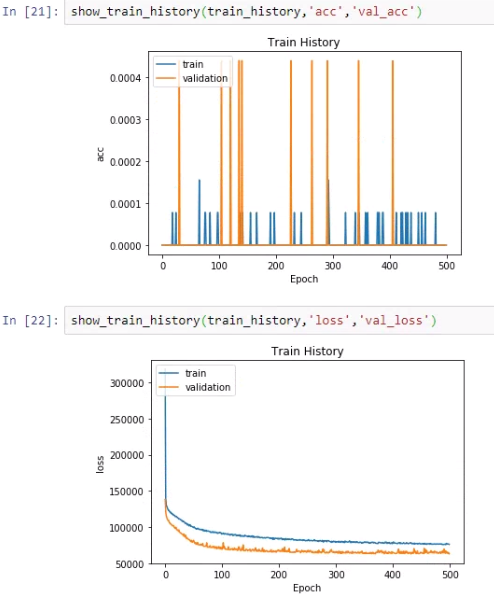


Step13：將輸出資料匯出成csv檔案



1. **畫圖做結果分析:**

此實驗所預測之準確值與損失函數之圖形



1. **討論預測值誤差很大，是怎麼回事？**
2. 代入網路參數過少，導致訓練成效不佳
3. 正規化方式不對，使用最大最小值的正規化方式不如使 用標準化的正規化方式來的好
4. 訓練網路次數設定不高
5. 網路模型層數設定不當
6. **如何改進？**

反覆試驗，代入較佳的參數

最大最小值的正規化易受離群值影響，所以改用標準化的方式來正規化參數

反覆試驗，調整較好的訓練周期，之前模擬時使用300的訓練週期，但比不上使用500的訓練週期

反覆試驗神經網路模型的層數與神經元設計，之前所使用2,3,5層的神經網路都比不上設計為4層時的神經網路;每層的神經元個數也是依靠反覆試驗測得，太多太少都會造成訓練效果不佳

1. **Kaggle排名**

