Introduction to IoT Data Analytics

B10601002 廖品捷 – 實作 + 書面

B10730004 吳承翰 – 實作 + 書面

A10815004 李筱慧 – ppt製作

Github: <https://github.com/JamesLiao714>

數據預處理與標準化：

1. 將timestamp再細分為星期、小時、分鐘。
2. 將空白的資料以該資料前30min的數據先填補，以作統計用。
3. 用公式 ”(資料值 - 資料平均) / 標準差” 做標準化。

第一階段：針對三種需要填補的資料分別使用全連接層計算填補值

剛開始的使用模型為三層中間層，神經元數量分別為32、64、32，皆使用relu，輸出層則為linear；使用參數為缺少資料前30sec、60sec的數值與當前時間的另外兩筆資料，共4個參數；epoch數為15。這樣的模型與參數validation後的結果Loss大約在30上下，非常不理想。

接下來嘗試增加模型複雜度：模型不變，但將30sec前的另外兩筆資料也列入考慮，於是有了6個參數，並增加epoch數至20。如此一來上傳Kaggle後計算的Score為6.9左右。

再次增加模型複雜度：參數增加當天的星期與當前時間，共8個參數，其餘與先前無異。上傳Kaggle的Score約為6.3左右。接著嘗試正規化後再進行計算，結果卻非常不理想，增加epoch數至100後上傳仍有15以上的Score。

先放棄正規化，並嘗試降低模型複雜度：模型為三層中間層，神經元數量分別為32、32、32，皆使用relu，輸出層則為linear；使用參數為缺少資料前30sec的數值、當前時間的另外兩筆資料與星期、時間，共5個參數；epoch數為20。這樣的模型與參數結果Score大約為5.4，是目前最理想的結果。

第二階段：開始進行全連接層外的嘗試

1. energy minimization：

調整的方向主要是透過α值的改變來看結果的變化：一開始設的是0.01，丟到Kaggle上後的成績是7.14；接著逐步降低α，當值來到0.00001時，分數變成6.4，接下來的進步幅度降低，於是停止嘗試。

1. LSTM與NN：

將全連接層的模型換成LSTM，參數則維持，如此一來validation後的結果Loss約為7，並沒有特別理想。而後回頭嘗試全連接層，模型中增加一層128個神經元的relu並使用內建的regularizer，結果Loss進入4出頭，效果極佳，但上傳資料格式出了一點問題。

1. XGBoost：

參數的部分與先前無異，結果卻一鳴驚人，出現1.529的數字；但繼續測試後的結果並不穩定，偶而Score還是會上升到6左右

**預測資料**：針對各資料分別使用全連接層進行預測

模型為三層中間層，神經元數量分別為32、32、32，皆使用relu，輸出層則為linear；使用參數為欲預測資料前30sec、60sec的三筆資料與當前的星期、時間，共8個參數；epoch數為20。由於對Score的影響不大，於是將重點放在增進填補值的品質。

**比較：**

* Xgboost：

1.不穩

2.預測方法違和

3.還要轉換y\_value成整數

* ＮＮ（custom layer with regularizer）

1.穩定且效果好

2.參數及神經元layer難懂較難理解

* LSTM

1.概念上感覺要是最好，但loss function結果卻最大

2.Overfitting需進行參數調整

結語：

雖然上傳的最好結果是用xgboost跑的，但未來再做相關預測時，我選擇的model應該還是用上述的NN model，以比較穩固的方式來預測。關於參數的設置我們都了解不深，只能透過經驗法則來摸索，尚未接觸一套完整的選擇方式，這是這次作業中最大的阻礙；此外，參考網路上前人所寫的原始碼與模型是非常重要的，可以事半功倍。希望往後得到更多經驗、深入了解更多方法，在資料科學這條路上持續努力，以在未來更有效率地做出更準確的成果。