

【智慧加工系統】課程

《跟 NVIDIA 學深度學習》作業三

第 5、6 章回家作業

授課教師：陳響亮 教授

作業繳交時限：12/09 (二) 24:00

助教：張朝雄

經過老師深入淺出的講解課本《跟 NVIDIA 學深度學習》第 5、6 章，介紹了循環神經網路的架構。同學們一定迫不及待想要開始實作，作業為 1 個，滿分 100 分：

(3) 作業三：多種實驗條件下，預測 CNC 加工數據(100 分)

本次作業三，請壓縮成 zip，需要分別包含檔案如下：

1. 環境建置文件 (requirement.txt)
2. 完整模型建立與訓練程式碼
3. 測試模型程式碼
4. 模型權重檔 (包含：自己堆疊與訓練的模型權重檔、現有架構訓練出來的模型權重檔)
5. 模型預測的 45 張變量與真實值比較折線圖
6. 模型預測測試集答案(csv 格式繳交，"test_學號.csv")
7. 實作報告 (Word 文稿格式，請參考提供的模板)

※ 壓縮檔檔名："學號_名字_深度學習_作業三.zip"

※ 檔案過大可於課堂作業繳交時間，攜帶隨身碟直接繳交檔案

※作業相關檔案連結：包含資料集、繳交模板與繳交範例

<https://drive.google.com/drive/folders/1rXRkW8PqUflzSz69979HcxFfidGJ5MBT?usp=sharing>

作業三：多種實驗條件下，預測 CNC 加工數據

作業三要請同學練習以 Sequential 序列模型進行 RNN 模型或是 LSTM 模型建構並訓練，完成多種實驗條件下，預測 CNC 加工數據的任務。並撰寫完整實作作業報告。本作業使用 CNC Mill Tool Wear 公開資料集重新處理，資料來自 **CNC 銑床在實際加工過程中的感測與控制信號（位置、速度、電流、電壓、進給率...）**，有 48 個變量，但是要最終預測目標為 45 個變量，每 0.1 秒(100 ms)取樣一次。

(原始連結：<https://www.kaggle.com/datasets/shasun/tool-wear-detection-in-cnc-mill/data>)

STEP 1：匯入檔案並切分資料集(5分)

- 本作業使用之資料集經過助教處理，務必至第一頁 google drive 連結下載檔案。(不要從原始 github 或是 kaggle 連結下載。)
<https://drive.google.com/drive/folders/1rXRkW8PqUflzSz69979HcxFfidGJ5MBT?usp=sharing>
- 解壓縮後的資料集結構如下圖 3.1，其中，Hw3_train 裡面共有 8 組 CNC 加工實驗的數據，以及一個 train_summary.csv 的檔案，用來描述這幾個實驗的條件。作為同學分割訓練集(train)與驗證集(validation)使用。
- Hw3_test_inputs 內的 csv，作為測試集 (test)。因此，**提供 100 組樣本，每組樣本都有 10 筆數據，請同學預測每組樣本未來 50 筆數據的 45 個欄位數值。欄位如下**

# X 軸 "X1_ActualPosition", "X1_ActualVelocity", "X1_ActualAcceleration", "X1_CommandPosition", "X1_CommandVelocity", "X1_CommandAcceleration", "X1_CurrentFeedback", "X1_DCBusVoltage", "X1_OutputCurrent", "X1_OutputVoltage", "X1_OutputPower",	# Y 軸 "Y1_ActualPosition", "Y1_ActualVelocity", "Y1_ActualAcceleration", "Y1_CommandPosition", "Y1_CommandVelocity", "Y1_CommandAcceleration", "Y1_CurrentFeedback", "Y1_DCBusVoltage", "Y1_OutputCurrent", "Y1_OutputVoltage", "Y1_OutputPower",	# Z 軸 "Z1_ActualPosition", "Z1_ActualVelocity", "Z1_ActualAcceleration", "Z1_CommandPosition", "Z1_CommandVelocity", "Z1_CommandAcceleration", "Z1_CurrentFeedback", "Z1_DCBusVoltage", "Z1_OutputCurrent", "Z1_OutputVoltage",
# S 軸 (主軸) "S1_ActualPosition", "S1_ActualVelocity", "S1_ActualAcceleration", "S1_CommandPosition", "S1_CommandVelocity", "S1_CommandAcceleration", "S1_CurrentFeedback", "S1_DCBusVoltage", "S1_OutputCurrent", "S1_OutputVoltage", "S1_OutputPower", "S1_SystemInertia",		# 進給率 "M1_CURRENT_FEEDRATE"

- 針對測試集(validation)使用模型預測的結果以 csv 格式留存紀錄，格式如 **submission_template.csv** 所示，一共有 45 個變量和 50 個時間，總共 $45 \times 50 = 2250$ 欄，

請同學務必再次確認自己的繳交格式正確。

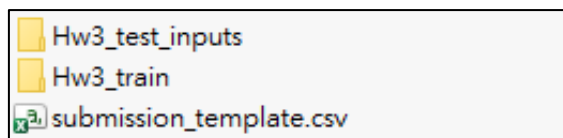


圖 3.1 資料集的資料夾結構

STEP 2 : 資料前處理(10 分)

依課本方法或是查找額外方法進行資料前處理。

STEP 3 : 以 Keras 或是 PyTorch Sequential 序列(或是之後會教的 Function API)，設計堆疊模型，並且輸出模型架構摘要(10 分)

STEP 4 : 完成模型訓練 (10 分)

- 不同超參數配置的訓練過程 loss、MSE、MAE 變化，以 csv 格式留存紀錄。
- 針對 experiment_14.csv 使用模型預測，結果以 csv 格式留存紀錄，並使用 matplotlib 繪製真實數值與預測數值沿時序的差異折線圖，如圖 3.2。**記得要每個變量分別繪製，一共要 45 張圖。**

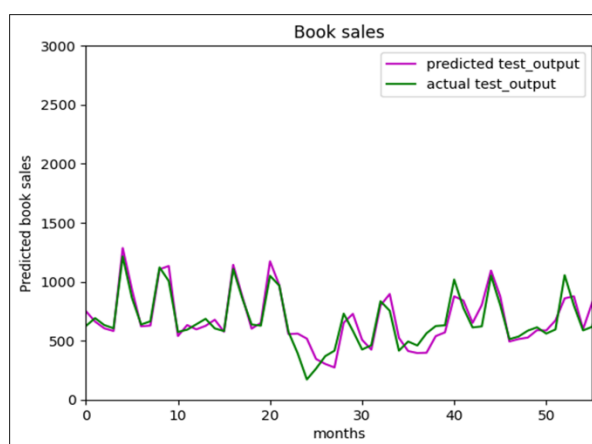


圖 3.2 教材第 5 章用 RNN 預測書店銷售額與真實銷售額折線圖

STEP 5 : 使用模型預測測試集，並儲存欲提交之模型預測結果(20 分)

- 針對測試集(test)使用模型預測的結果以 csv 格式留存紀錄，格式如 submission_template.csv 儲存 sample_id 與預測結果，檔名請命名為” test_學號.csv”
- 由助教比對” test_學號.csv”與正解計分，以 WRMSE(加權均方根誤差)自動計算分數比例，詳細計分方式見補充說明。

STEP 6：嘗試現有的時間序列模型並討論比較與自己製作的架構差異(20 分)

可嘗試 1 個以上的不同模型，如：GRU、2DLSTM.....等模型，並且，建議以 MAE、MSE、RMSE.....等指標，對比與討論不同模型間的訓練結果。

STEP 7：報告撰寫(25 分)

建議至少 7 頁，並搭配圖表說明，內容格式如 google drive 附件之作業模板.docx

補充說明：WRMSE

這個作業的自動評分部分滿分 20 分，評分流程如下：

1. 變數標準化（以訓練資料為基準）

對每個變數 v ，助教會用 **訓練實驗資料** 先計算：

- 平均值： μ_v
- 標準差： σ_v

接著對你的預測誤差做「以標準差為單位」的標準化：

$$e_{s,h,v} = \frac{\hat{y}_{s,h,v} - y_{s,h,v}}{\sigma_v}$$

- $y_{s,h,v}$ ：真實值（助教用隱藏資料計算，不公開）
- $\hat{y}_{s,h,v}$ ：你提交的預測
- $e_{s,h,v}$ ：單位是 誤差相對 σ_v 為「幾個標準差」

2. 時間衰減權重 w_h

對不同時間點 h 設定時間權重 w_h ：越接近現在（第 1 秒）的預測比較重要，越遠（第 4~5 秒）的預測容許誤差可以大一點。也就是說，前面幾秒的誤差會被算得比較重。將 50 個時間步分成 5 個區間（每秒 10 筆）：

- 第 1 秒內：1.0
- 第 2 秒內：0.8
- 第 3 秒內：0.6
- 第 4 秒內：0.4
- 第 5 秒內：0.2

然後做一次縮放，讓它們的總和剛好為 50（也就是 H ）：

$$\sum_{h=1}^{50} w_h = 50$$

3. 加權 MSE 與 WRMSE

把所有 sample(S)、所有變數(V)、所有時間(h)的誤差，用時間權重加總後取平均：

$$\text{WMSE} = \frac{1}{|S| \cdot |V| \cdot H} \sum_{s \in S} \sum_{v \in V} \sum_{h=1}^H w_h e_{s,h,v}^2$$

然後再開根號，得到 **加權 RMSE**：

$$\text{WRMSE} = \sqrt{\text{WMSE}}$$

WRMSE 沒有「物理單位」，可以解讀為：「平均來說，預測誤差有幾個標準差大」。最後會把 WRMSE 線性轉換成 0~20 分。

- 若 **WRMSE** ≈ 0 （幾乎完美預測） \rightarrow 得分 ≈ 20 分
- 若 **WRMSE** ≈ 1 （平均誤差約 1 個標準差） \rightarrow 約 10 分
- 若 **WRMSE** ≥ 2 （平均誤差超過 2 個標準差） \rightarrow 此部分分數趨近 0 分

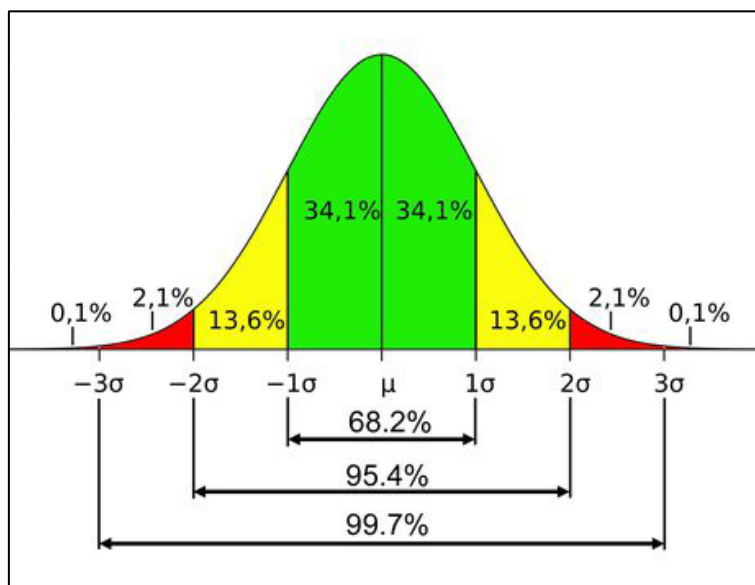


圖 3.3 標準差與數值分布關係