深度學習及其在生物產業的應用

題目:萵苣生長預測

姓名:陳一禕

學號:B1031024

2024/6/8

### 題目

萵苣生長預測

### 背景

植物工廠可以將室內環境控制在我們需要的範圍內，但是成本相對較高，適合栽培周期短的作物例如萵苣或是用來育苗，在種植期間更加注重植物生長周期，前一天或後一天出貨在業界就會受到很大的影響，植物的重量等品質也要兼顧，因此想利用這次課程做專題的機會訓練出能夠預測萵苣生長趨勢的結果，通過計算葉片大小及重量，查看前面的數值推估出未來可能會生長多寡。

### 研究目的(以條列式列出)

1.拍攝照片，進行影像處理計算出葉面積

2.將資料蒐集整理

3.撰寫「預測萵苣未來生長情況」之程式

4.嘗試回看不同筆資料 比較結果

### 四、使用電腦設備規格系統與其他設備

電腦型號:ASUS TUF Gaming F15

處理器:i7

GPU:3050

作業系統:Window 10

拍攝資料原圖設備:iPhone 12 mini

### 五、研究步驟(以流程圖表示)

顯示預測結果討論分析

撰寫預測程式

將影像處理的結果存成excel檔傳至雲端

### 六、程式碼區段說明

|  |  |  |
| --- | --- | --- |
| 行 | 程式碼 | 說明 |
| 1  2  3  4  5  6  7  8 | import numpy as np  import pandas as pd  from sklearn.model\_selection import train\_test\_split  from sklearn.preprocessing import StandardScaler  from tensorflow.keras.models import Sequential  from tensorflow.keras.layers import GRU, Dense  from tensorflow.keras.optimizers import Adam  import matplotlib.pyplot as plt | 導入需要的函式庫 |

|  |  |  |
| --- | --- | --- |
| 行 | 程式碼 | 說明 |
| 1  2 | from google.colab import drive  drive.mount('/content/drive') | 連接到雲端 |

|  |  |  |
| --- | --- | --- |
| 行 | 程式碼 | 說明 |
| 1  2  3  4 | file\_path = 'drive/MyDrive/letturce\_growing.xlsx'  df = pd.read\_excel(file\_path)  df.head() | 檔案位址  讀取文件  顯示前面數據 |

|  |  |  |
| --- | --- | --- |
| 行 | 程式碼 | 說明 |
| 1  2  3  4  5 | df['Date'] = pd.to\_datetime(df['Date'])  features = ['Temperature', 'Humidity', 'CO2', 'area', 'height']  target = 'weight'  scaler = StandardScaler()  df[features] = scaler.fit\_transform(df[features]) | datetime格式  定義特徵  定義目標  將數據標準化 |

|  |  |  |
| --- | --- | --- |
| 行 | 程式碼 | 說明 |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | def create\_sequences(data, sequence\_length):  xs, ys = [], []  for i in range(len(data) - sequence\_length):  x = data[i:(i + sequence\_length), :-1]  y = data[i + sequence\_length, -1]  xs.append(x)  ys.append(y)  return np.array(xs), np.array(ys)  sequence\_length = 2  plant\_data = []  for plant\_id in df['Plant ID'].unique():  plant\_df = df[df['Plant ID'] == plant\_id].drop(columns=['Plant ID', 'Date']).values  X, y = create\_sequences(plant\_df, sequence\_length)  plant\_data.append((X, y))  X\_all = np.concatenate([data[0] for data in plant\_data])  y\_all = np.concatenate([data[1] for data in plant\_data])  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_all, y\_all, test\_size=0.7, random\_state=42) | 定義一個函數  輸入序列和目標值  數據範圍  第一個至倒數二個  第一個到最後一個  把x加到xs  把y加到ys  返回  設置回看天數  儲存時間序列  創建序列數據  獲取id  其他數據轉numpy  X,y加到列表中  X合併成大數組  y合併成大數組  把數據拆分成訓練集跟測式集 |

|  |  |  |
| --- | --- | --- |
| 行 | 程式碼 | 說明 |
| 1  2  3  4  5 | model = Sequential()  model.add(GRU(50, activation='relu', input\_shape=(sequence\_length, X\_train.shape[2])))  model.add(Dense(2))  model.compile(optimizer=Adam(learning\_rate=0.004), loss='mse')  history = model.fit(X\_train, y\_train, epochs=430, batch\_size=62, validation\_split=0.2) | 建模型  加GRU層及參數  加Dense  編譯模型  訓練模型 |

|  |  |  |
| --- | --- | --- |
| 行 | 程式碼 | 說明 |
| 1  2  3  4  5  6 | # 繪製損失函數  plt.figure(figsize = (12, 6))  plt.plot(history.history['loss'], label='loss')  plt.plot(history.history['val\_loss'], label='val\_loss')  plt.legend(loc='best')  plt.show() | 圖形尺寸  畫出Loss的曲線  Val\_loss的曲線  自動選擇最佳位置  顯示結果 |

|  |  |  |
| --- | --- | --- |
| 行 | 程式碼 | 說明 |
| 1  2  3  4  5 | gru\_weights = model.layers[0].get\_weights()  dense\_weights = model.layers[1].get\_weights()  feature\_means = np.mean(np.abs(gru\_weights[0]), axis=1)  for feature, mean\_weight in zip(features, feature\_means):  print(f"{feature}: {mean\_weight}") | 提取GRU層的權重  提取Dense層的權重  計算特徵權重平均值  顯示平均權重 |

|  |  |  |
| --- | --- | --- |
| 行 | 程式碼 | 說明 |
| 1  2  3  4  5  6  7  8 | mse = model.evaluate(X\_test, y\_test)  print(f'MSE: {mse}')  # 為每顆萵苣進行預測並繪圖  for plant\_id in df['Plant ID'].unique():  plant\_df = df[df['Plant ID'] == plant\_id].drop(columns=['Plant ID']).sort\_values(by='Date')  plant\_values = plant\_df.drop(columns=['Date']).values  dates = plant\_df['Date'].values[sequence\_length:]  X, y = create\_sequences(plant\_values, sequence\_length)  y\_pred = model.predict(X)  # 繪圖  plt.figure(figsize=(10, 6))  plt.plot(dates, y, label='Actual Plant Weight')  plt.plot(dates, y\_pred, label='Predicted Plant Weight')  plt.title(f'Plant ID {plant\_id} - Actual vs Predicted Plant Weight')  plt.xlabel('Date')  plt.ylabel('Plant Weight')  plt.legend()  plt.show() | 評估模型  顯示均方誤差  對每株萵苣預測  取id對應的所有數據並按日期排列  保留特徵和目標值  獲取日期  使用之前定義的函數得到預測值  圖形大小  實際值  預測值  標題  X軸  Y軸  顯示圖例  繪製結果 |

### 七、結果與討論(依據研究目的逐項以圖表等列出結果，並進行說明與討論)

1.模型訓練結果:

loss: 0.8870 - val\_loss: 11.2965

參數值設定:

test\_size=0.7

learning\_rate=0.004

epochs=430

batch\_size=62

validation\_split=0.2

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述

損失函數圖:

一張含有 文字, 螢幕擷取畫面, 行, 陳列 的圖片

自動產生的描述

3.權重

一張含有 文字, 字型, 螢幕擷取畫面, 白色 的圖片

自動產生的描述

2.預測結果

均方誤差值:



各株實際值與預測值比較:

一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 行, 圖表 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

自動產生的描述 一張含有 文字, 圖表, 繪圖, 行 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述

一張含有 文字, 圖表, 行, 螢幕擷取畫面 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 行, 圖表 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 行, 圖表 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 繪圖, 行 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片

自動產生的描述

此次測試結果我認為是最好的一次，因為loss值有降至0.8，val\_loss值沒有上升，表示我沒有過度訓練。其中被紅色框起來的預測圖id為8、11、15號，這些是與預測值相差較大的植物，當我回去看這三棵植物的紀錄時可以看到，在area跟hieght的值是比較有落差的，但是預測的權重值顯示這兩項對之後預測是最重要的，我認為也許是影像處理的部分還有待修正，造成一開始資料不是這麼精準。id 11、15號的植物我發現這兩株是長不好的，可能需要先去除這兩筆資料再訓練。

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述

### 八、參考文獻

1.ChatGPT

### 九、附錄(本專題的程式檔ipynb檔案或py檔，相關的影像、聲音檔等)

1.本專題的程式檔(ipynb檔)

2.預測及訓練的資料檔(excel檔)

3.事前將影像轉換為數據的影像處裡程式檔(py檔)

4.照片檔案夾(JPG檔)