



研究所課程專題及作品

-----• 報告者：國立中興大學 陳怡秀 •-----



目錄

CONTENTS

01

機器學習-稻熱病預測

02

高等影像處理

03

時空資料分析

03-1

Moving MNIST

03-2

股價預測

03-3

太陽能源預測

03-4

UCF101動作識別



01



機器學習-稻熱病預測



稻熱病菌是一種植物病原菌，稻熱病每年摧毀了可供超過6000萬人食用的水稻，目前已知85個國家有稻熱病發生。

水稻若是在葉片發病，稻葉會先出現褐色或暗綠色的小斑點，五至七天就會擴大成黃褐色紡錘狀的病斑，待葉片組織被破壞殆盡，全株便會枯乾。

訓練資料集有2368筆資料，測試資料集有1183筆資料。

StnPres	StnPresMax	StnPresMin	Temperature	T Max	T Min	RH	RHMin	Precp	Tc	T cd	StnPres.1
	StnPresMax.1	StnPresMin.1	Temperature.1	T Max.1	T Min.1	RH.1	RHMin.1	Precp.1	Tc.1	T cd.1	StnPres.2
	StnPresMax.2	StnPresMin.2	Temperature.2	T Max.2	T Min.2	RH.2	RHMin.2	Precp.2	Tc.2	T cd.2	StnPres.3
	StnPresMax.3	StnPresMin.3	Temperature.3	T Max.3	T Min.3	RH.3	RHMin.3	Precp.3	Tc.3	T cd.3	StnPres.4
	StnPresMax.4	StnPresMin.4	Temperature.4	T Max.4	T Min.4	RH.4	RHMin.4	Precp.4	Tc.4	T cd.4	StnPres.5
	StnPresMax.5	StnPresMin.5	Temperature.5	T Max.5	T Min.5	RH.5	RHMin.5	Precp.5	Tc.5	T cd.5	StnPres.6
	StnPresMax.6	StnPresMin.6	Temperature.6	T Max.6	T Min.6	RH.6	RHMin.6	Precp.6	Tc.6	T cd.6	StnPres.7
	StnPresMax.7	StnPresMin.7	Temperature.7	T Max.7	T Min.7	RH.7	RHMin.7	Precp.7	Tc.7	T cd.7	StnPres.8
	StnPresMax.8	StnPresMin.8	Temperature.8	T Max.8	T Min.8	RH.8	RHMin.8	Precp.8	Tc.8	T cd.8	StnPres.9
	StnPresMax.9	StnPresMin.9	Temperature.9	T Max.9	T Min.9	RH.9	RHMin.9	Precp.9	Tc.9	T cd.9	StnPres.10
	StnPresMax.10	StnPresMin.10	Temperature.10	T Max.10	T Min.10	RH.10	RHMin.10	Precp.10	Tc.10	T cd.10	StnPres.11
	StnPresMax.11	StnPresMin.11	Temperature.11	T Max.11	T Min.11	RH.11	RHMin.11	Precp.11	Tc.11	T cd.11	StnPres.12
	StnPresMax.12	StnPresMin.12	Temperature.12	T Max.12	T Min.12	RH.12	RHMin.12	Precp.12	Tc.12	T cd.12	StnPres.13
	StnPresMax.13	StnPresMin.13	Temperature.13	T Max.13	T Min.13	RH.13	RHMin.13	Precp.13	Tc.13	T cd.13	StnPres_mean
	StnPresMax_mean	StnPresMin_mean	Temperature_mean	T Max_mean	T Min_mean	RH_mean	RHMin_mean	Precp_mean	Tc_mean	T cd_mean	
	RH_count	Precp_count	Precp_f_con_RH	con_Precp		con_good_tmp	Precp_T_month				

label
1
0
0
0
1

X有172項特徵。
包含14天的各項天氣資料數值，如：大氣壓力、溫度、雨量、濕度、孢子萌發率等等。

label,
預測水稻是否有稻熱病(RiceBlast)，結果為0或1。

結果比較

分類器	accuracy	precision	recall	f1_score
Decision Tree	0.6904	0.6980	0.6910	0.6945
LogisticRegression	0.7081	0.6992	0.7492	0.7233
SGDClassifier	0.7073	0.7177	0.7010	0.7092
RandomForest	0.7733	0.7470	0.8389	0.7903
GaussianNB	0.7014	0.7005	0.7226	0.7114
SVC	0.7200	0.6978	0.7940	0.7428
GridSearchCV (kernel='sigmoid', C=0.1,degree= 2, gamma=0.1)	0.7200	0.6978	0.7940	0.7428
VotingClassifier	0.7623	0.7385	0.8256	0.7796
BaggingClassifier	0.7360	0.7210	0.7857	0.7520
StackingClassifier	0.7868	0.7589	0.8522	0.8028
neural network - MLPClassifier(sklearn)	0.7360	0.7369	0.7492	0.7430
neural network(tensorflow)	0.7453	0.7319	0.7890	0.7594
neural network(tensorflow)- 三層 hidden layers	0.7750	0.7684	0.7990	0.7834
XGBoost	0.8037	0.7846	0.8472	0.8147



02

高等影像處理





影像處理



原圖



灰階



負片



Gamma < 1



對比拉開



Gamma > 1



胡椒鹽雜訊



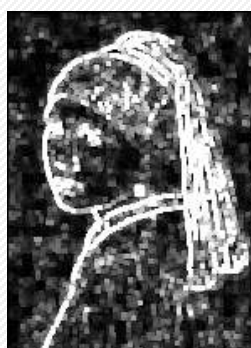
Laplacian 邊緣偵測



OTSU 二值化 (*Bonus)



3*3 中值濾波器



3*3 最大值濾波器

1. 使用JAVA對圖片進行轉換。



03



時空資料分析



03-1

時空資料分析-Moving MNIST



Moving MNIST資料集包含 10,000 個影片序列，每個影片序列由 20 幀組成。

在每個視訊序列中，兩個數字在幀周圍獨立移動，該幀的空間解析度為 64×64 像素。這些數字經常相互交叉並從框架邊緣反彈。



結果比較

LSTM 模型的准确度: 0.9293838143348694
MAE: 0.08372804522514343
MSE: 0.03961997851729393

ConvLSTM 模型的准确度: 0.9306845664978027
MAE: 0.07992171496152878
MSE: 0.03644556552171707

準確度: ConvLSTM 模型的準確度稍高於 LSTM 模型, 顯示 ConvLSTM 模型在對資料進行分類時略微更準確一些。

MAE: 較低的 MAE 值表示 ConvLSTM 模型的預測結果與實際標籤的平均差異較小, 因此 ConvLSTM 模型在預測上較為優秀。

MSE: 較低的 MSE 值顯示 ConvLSTM 模型的預測結果與實際標籤的變異數較小, 因此 ConvLSTM 模型在損失函數值上也較優。



03-2

時空資料分析-股價預測



**yfinance 是一個 Python 套件，
提供了一種簡單的方法來從
Yahoo 財經獲取金融資料。**

1. Date : 日期，表明資料點的具體日期。
2. Open : 開盤價，指股票在該交易日開市時的價格。
3. High : 最高價，指股票在該交易日的最高交易價格。
4. Low : 最低價，指股票在該交易日的最低交易價格。
5. Close : 收盤價，指股票在該交易日結束時的價格。
6. Adj Close : 調整後收盤價，將股票分割和股息等因素考慮進去後的收盤價。
7. Volume : 交易量，表示在該交易日內買賣該股票的總股數。



使用LSTM預測APPLE inc的收盤價

輸入：

從歷史收盤價資料中構建訓練集，其中每個樣本包含了過去 60 天的收盤價歷史數據，用於訓練 LSTM 模型。因此，特徵維度為 (60, 1)，其中 60 是時間步長，1 是每個時間步的特徵數量。

輸出：

每個樣本的輸出特徵是相應未來一個時間點的股價，即未來的收盤價。

```
# Build the LSTM model
model = Sequential()
model.add(LSTM(128, return_sequences=True, input_shape= (x_train.shape[1], 1)))
model.add(LSTM(64, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))

optimizer = Adam(learning_rate=0.001) # 設置學習率為0.001
model.compile(optimizer=optimizer, loss='mean_squared_error')
```

Average Accuracy: [98.59078082]



03-3

時空資料分析-太陽能源預測



全球天氣資料集

(GlobalWeatherRepository.csv)

該資料集提供了世界各地首都的每日天氣資訊。

與預報資料不同，此資料集提供了一組全面的特徵，反映了世界各地當前的天氣狀況。

提供超過40多個功能，包括溫度、風力、壓力、降水、濕度、能見度、空氣品質測量等。該數據集對於分析全球天氣模式、探索氣候趨勢以及了解不同天氣參數之間的關係非常有價值。

+

IEA 月度電力統計資料集

(data.csv)

該資料集是從國際能源總署 (IEA) 網站收集的，特別是從其每月電力統計工具中收集的。該數據包含 2010 年至 2022 年各國每月的能源生產資訊。能源生產以吉瓦時 (GWh) 為單位，涵蓋水力、風能、太陽能、地熱能、核能、化石能源等一系列能源產品/燃料等。



使用LSTM預測挪威的太陽能發電量

輸入：

全球天氣資料集-天氣資料：包括溫度、濕度、日出時間、日落時間等。

IEA 月度電力統計資料集-能源的歷史產量資訊：國家、該產品在該國總發電量中所佔的份額、發電量等。

輸出：

太陽能發電量預測值
以每天為時間間隔來預測未來相同一段時間內的太陽能發電量。

```
class LSTMModel(nn.Module):
    def __init__(self, input_size, hidden_size, output_size, num_layers, dropout_prob=0.5):
        super(LSTMModel, self).__init__()
        self.num_layers = int(num_layers)
        self.hidden_size = hidden_size
        self.lstms = nn.ModuleList([nn.LSTM(input_size if i == 0 else hidden_size, hidden_size, batch_first=True) for i in range(self.num_layers)])
        self.dropout = nn.Dropout(dropout_prob)
        self.fc = nn.Linear(hidden_size, output_size)

    def forward(self, x):
        batch_size = x.size(0)

        out = x
        for lstm in self.lstms:
            # 初始化隱藏狀態的參數
            h0 = torch.zeros(1, batch_size, self.hidden_size).to(x.device)
            c0 = torch.zeros(1, batch_size, self.hidden_size).to(x.device)

            out, _ = lstm(out, (h0, c0))
            out = self.dropout(out)

        out = self.fc(out[:, -1, :])
        return out
```

R2 Score: 0.9940141725193333,

Mean Squared Error: 0.00611870875582099,

Mean Absolute Error: 0.044078100472688675



03-4

時空資料分析-UCF101動作識別



資料集



1. 總影片數： 13,320個
2. 影片總長度： 27小時
3. 影片來源： YouTube擷取
4. 影片類別： 101 種
5. 每個類別分為25組。每組可以包含4-7個動作影片。
來自同一組的影片可能具有一些共同特徵，例如相似的背景、相似的視點等。
6. 包括5類動作： 人與物體交互，單純的肢體動作，人與人交互，演奏樂器，運動。

使用ConvLSTM進行UCF101 Action Recognition

輸入：

只取最後60個類別中的影片，連續的
幀圖像作為模型的輸入。

將幀圖像調整為相同大小（60x60像素）
和歸一化（將像素值縮放到0到1間）。
圖像資料的形狀：（7832, 5, 60, 60, 3）

輸出：

對每個影片預測的動作類別標籤。
標籤資料的形狀：（7832,）
類別數量：60種

```
model = Sequential()

model.add(ConvLSTM2D(filters = 16, kernel_size=(3,3), activation='LeakyReLU', data_format='channel_s_last', return_sequences=True, recurrent_dropout=0.2, input_shape=[x_train.shape[1], x_train.shape[2], x_train.shape[3], 3]))
model.add(MaxPooling3D(pool_size=(1,2,2), padding='same', data_format='channels_last'))
model.add(TimeDistributed(Dropout(0.2)))

model.add(ConvLSTM2D(filters = 16, kernel_size=(3,3), activation='LeakyReLU', data_format='channel_s_last', return_sequences=True, recurrent_dropout=0.2))
model.add(MaxPooling3D(pool_size=(1,2,2), padding='same', data_format='channels_last'))
model.add(TimeDistributed(Dropout(0.2)))

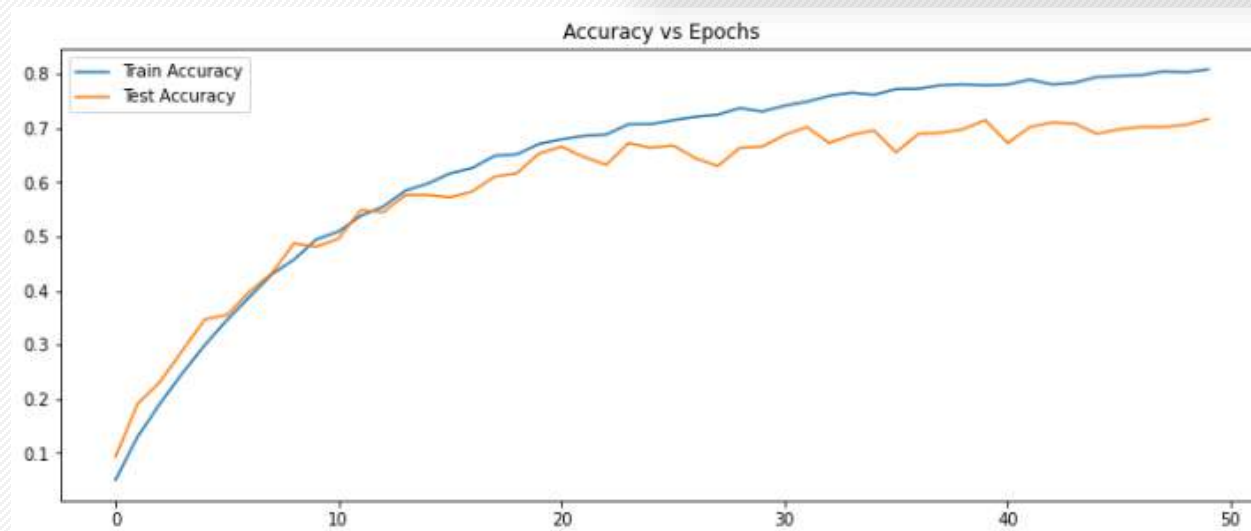
model.add(ConvLSTM2D(filters = 16, kernel_size=(3,3), activation='LeakyReLU', data_format='channel_s_last', return_sequences=True, recurrent_dropout=0.2))
model.add(MaxPooling3D(pool_size=(1,2,2), padding='same', data_format='channels_last'))
model.add(TimeDistributed(Dropout(0.2)))

model.add(Flatten())

model.add(Dense(60, activation='softmax'))
model.summary()
```

```
#Test Accuracy
accuracy_score(y_test, predicted_classes)
```

0.7170212765957447



THE END

Thanks For Listening!