The background features a stylized illustration of an industrial landscape. On the left, two smokestacks emit dark, billowing smoke. In the foreground, there are several buildings with grid-like windows. On the right, another set of smokestacks is shown, with one emitting a large plume of smoke. The sky is a light beige color, and a large, semi-circular shape with diagonal hatching is positioned at the top center, resembling a sun or moon. The overall aesthetic is clean and modern, using a limited color palette of beige, brown, and dark grey.

# 十面霾伏

## 以LSTM預測PM2.5濃度

Group5  
110034544  
賴筱庭

# TABLE OF CONTENTS

01

## BACKGROUND

1. 研究動機與目的
2. 5W1H

02

## DATA PRE-PROCESS

1. 資料集描述
2. 資料預處理

03

## LSTM MODEL

1. 模型介紹
2. 模型建構

04

## TRAINING RESULT

1. 參數調整結果
2. 未來濃度預測

05

## CONCLUSION

結論與未來展望



# BACKGROUND

## | 01 |

1. 研究動機與目的
2. 5W1H



# 研究動機與目的

- PM2.5無法經由鼻腔去除，因此會進入人體
- PM2.5容易附著戴奧辛、重金屬等有害物質
- 長期吸入易引起過敏、氣喘、肺癌、心血管疾病
- 短期暴露在高濃度PM2.5的環境中，也會提高呼吸道疾病及死亡的風險



希望能利用深度學習模型來預測PM2.5的濃度，以提供PM2.5的預警功能



# 5W1H

## Why

暴露在高濃度的PM2.5下，會提高呼吸道疾病以及死亡風險

## What

預測PM2.5濃度，以提供PM2.5預警功能

## Where

新竹市

## Who

民眾、環保署相關科研人員

## When

欲對PM2.5濃度進行預測時

## How

Long Short-Term Memory

An illustration of an industrial landscape with various factory buildings and smokestacks. Two large, dark, irregular shapes representing smoke or steam rise from the buildings. The background features rolling hills and a small sun or moon. The overall color palette is warm, consisting of shades of brown, tan, and beige.

# DATA PRE-PROCESS

| 02 |

1. 資料集描述
2. 資料預處理

# 資料來源



# 資料集描述

- 起始於2019/3
- 每小時觀測、更新一次
- 本次選擇新竹市\_新竹站
- 2019/3~2021/11，共23516筆

協力單位



環境監測及資訊處  
業務職掌 環境監測規劃管理與品質保證 空氣品質監測規劃與網站管理 環境資訊規劃設計 環境資訊操作維護 服務項目摘要 水質監測、空氣監測、網路行政、環境資訊 [讀取更多](#)

追蹤者 72 資料集 216

空氣品質小時值\_新竹市\_新竹站

管 追蹤人數 0人

資料集

資料集名稱	下載
空氣品質小時值_新竹市_新竹站	
空氣品質小時值_新竹市_新竹站 (2021/12)	
空氣品質小時值_新竹市_新竹站 (2021/11)	
空氣品質小時值_新竹市_新竹站 (2021/10)	
空氣品質小時值_新竹市_新竹站 (2021/09)	
空氣品質小時值_新竹市_新竹站 (2021/08)	



# 資料集描述

欄位名稱	欄位意義
SiteId	觀測站代碼
SiteName	觀測站站名
County	觀測站所在縣市
ItemId	觀測物代碼
ItemName	觀測物中文名稱
<b>ItemEngName</b>	<b>觀測物英文名稱</b>
ItemUnit	觀測物單位
MonitorDate	觀測日期、時間
<b>Concentration</b>	<b>濃度 ( 數值 )</b>

SiteId	SiteName	County	ItemId	ItemName	ItemEngName	ItemUnit	MonitorDate	Concentration
24	新竹	新竹市	2	一氧化碳	CO	ppm	2019/3/31 23:	0.25
24	新竹	新竹市	1	二氧化硫	SO2	ppb	2019/3/31 23:	1.9
24	新竹	新竹市	4	懸浮微粒	PM10	$\mu$ g/m3	2019/3/31 23:	35
24	新竹	新竹市	7	二氧化氮	NO2	ppb	2019/3/31 23:	6.17
24	新竹	新竹市	33	細懸浮微	PM2.5	$\mu$ g/m3	2019/3/31 23:	12
24	新竹	新竹市	3	臭氧	O3	ppb	2019/3/31 23:	51.4
24	新竹	新竹市	3	臭氧	O3	ppb	2019/3/31 22:	51.1
24	新竹	新竹市	1	二氧化硫	SO2	ppb	2019/3/31 22:	1.6
24	新竹	新竹市	33	細懸浮微	PM2.5	$\mu$ g/m3	2019/3/31 22:	9
24	新竹	新竹市	7	二氧化氮	NO2	ppb	2019/3/31 22:	6.84
24	新竹	新竹市	4	懸浮微粒	PM10	$\mu$ g/m3	2019/3/31 22:	34
24	新竹	新竹市	2	一氧化碳	CO	ppm	2019/3/31 22:	0.26

# 資料預處理

1. 資料整合
2. 資料清理
3. 資料排序
4. 缺失值處理
5. 平穩性檢測
6. 切分資料集
7. 資料正規化
8. 整理訓練資料

33	細懸浮微粒PM2.5	$\mu\text{g}/\text{m}^3$	2019/3/5 11:00	24
7	二氧化氮 NO2	ppb	2019/3/5 11:00	x
4	懸浮微粒 PM10	$\mu\text{g}/\text{m}^3$	2019/3/5 11:00	37
3	臭氧 O3	ppb	2019/3/5 11:00	44.3
2	一氧化碳 CO	ppm	2019/3/5 11:00	0.64
1	二氧化硫 SO2	ppb	2019/3/5 11:00	2.4
33	細懸浮微粒PM2.5	$\mu\text{g}/\text{m}^3$	2019/3/5 10:00	x
7	二氧化氮 NO2	ppb	2019/3/5 10:00	x
4	懸浮微粒 PM10	$\mu\text{g}/\text{m}^3$	2019/3/5 10:00	x
3	臭氧 O3	ppb	2019/3/5 10:00	x
2	一氧化碳 CO	ppm	2019/3/5 10:00	x
1	二氧化硫 SO2	ppb	2019/3/5 10:00	x

資料整合與清理

資料排序

資料正規化

```
[ ] def normalize(train_set):  
    from sklearn.preprocessing import MinMaxScaler  
    #把數據縮放到到[0,1]之間  
    sc = MinMaxScaler(feature_range = (0, 1))  
    #需將資料做reshape的動作,使其shape為(資料長度,1)  
    train_set = train_set.values.reshape(-1,1)  
    training_set_scaled = sc.fit_transform(train_set)  
    return training_set_scaled, sc  
    return dt
```

整理訓練資料

```
[ ] def training_data_sorting(train_set, training_set_scaled, period_num):  
    X_train = []  
    y_train = []  
  
    for i in range(period_num, len(train_set)):  
        X_train.append(training_set_scaled[i-period_num : i-1, 0])  
        y_train.append(training_set_scaled[i, 0])  
    X_train, y_train = np.array(X_train), np.array(y_train)  
    X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))  
    return X_train, y_train  
    return train, test, train_set, test_set
```

# LSTM MODEL

| 03 |

1. 模型介紹
2. 模型建構



# 循環神經網路—RNN

- 以Slot Filling應用於智慧訂票系統為例，系統會自動將使用者的輸入分類到「出發地」「目的地」跟「時間」
- 若使用者對系統說：

1. 我要在11/2到達台北。
  2. 我要在11/2離開台北。
- 對前饋神經網路來說是沒有差異的

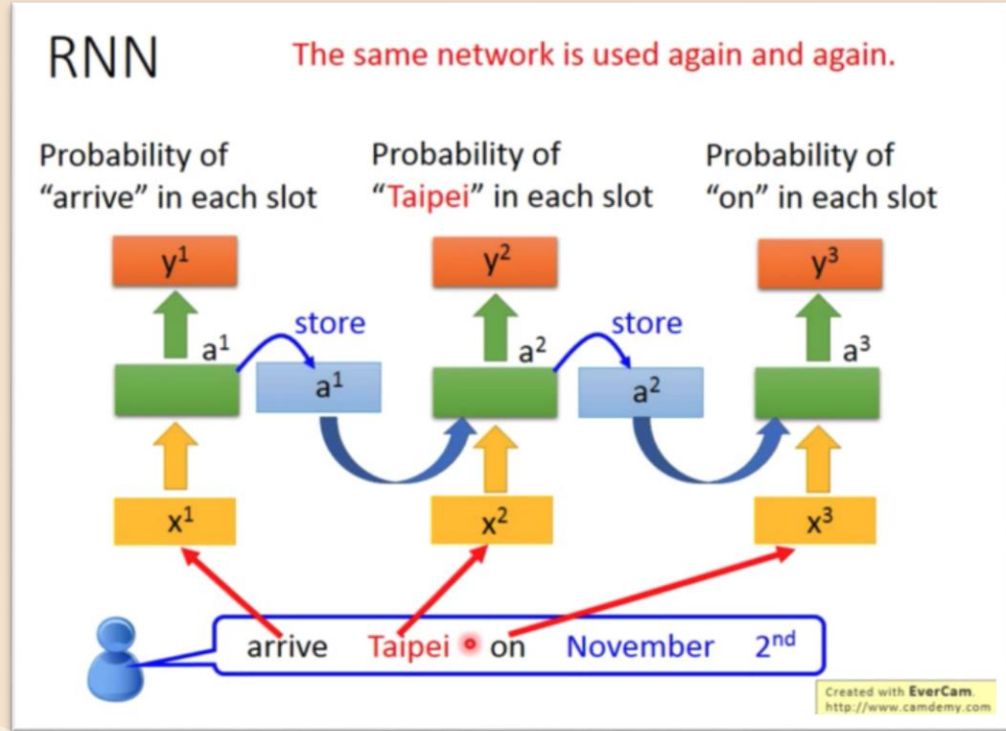
如果模型具有記憶力？



# 循環神經網路—RNN

- 具有記憶力的模型
- 強調資料間前後的關聯性
- 在處理時間序列資料時會有較好的成果

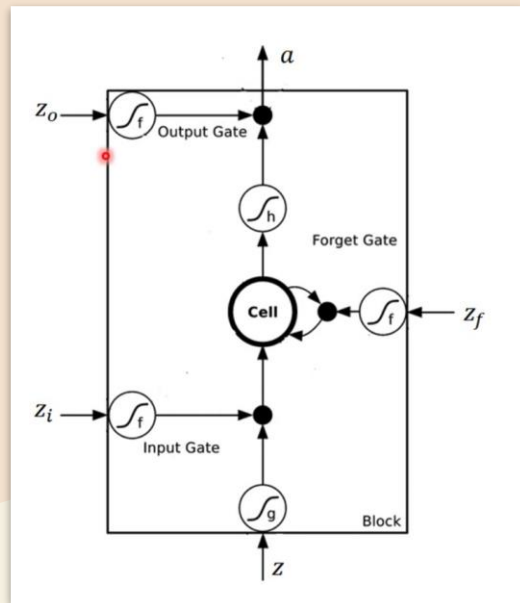
x Memory Cell的值會在下一個時間點就被洗掉



# 長短期記憶—LSTM

- 在Memory Cell取用上比較複雜，記憶時間也較長
- 加入了輸入閥 ( Input Gate )、遺忘閥 ( Forget Gate )、輸出閥 ( Output Gate )

名稱	功能	閥門開啟
輸入閥	控制是否將值輸入	輸入值將進入
遺忘閥	是否將Memory Cell裡的值清洗	會記得之前保存的值
輸出閥	控制是否將本次的計算結果輸出，若未輸出結果即為0	可輸出值



# 模型建構

```
[ ] def LSTM_model(train, test, X_train, y_train, sc, period_num, neurons_num, epochs_num, batch_num)
    import keras
    from keras.models import Sequential
    from keras.layers import Dense
    from keras.layers import LSTM
    from keras.layers import Dropout, BatchNormalization
    keras.backend.clear_session()
    regressor = Sequential()
    regressor.add(LSTM(units = neurons_num, activation='relu', input_shape = (X_train.shape[1], 1)))
    regressor.add(Dense(units = 1))
    regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')
    history = regressor.fit(X_train, y_train, epochs = epochs_num, batch_size = batch_num)
    plt.title('train_loss')
    plt.ylabel('loss')
    plt.xlabel('Epoch')
    plt.plot(history.history["loss"])
    plt.show()
```

# 參數調整

1. period\_num：用過去幾小時的資料進行預測  
4hr、24hr(1天)、48hr(2天)
2. neurons\_num：神經元數量  
1、32、128
3. epochs\_num：迭代次數  
50、100
4. batch\_num：批量大小  
40、160、320




進行全因子實驗

period	neurons	epochs	batch
4	1	50	40
24	32	100	160
48	128		320

共54種





# TRAINING RESULT

| 04 |

1. 參數調整結果
2. 未來濃度預測

# 參數調整結果\_各因子分析

以下將利用T檢定分析4個因子的各個水準是否有顯著差異(信心水準95%)：

- period：3個水準間皆無顯著差異
- neuron：1顆與32、128顆有顯著差異  
32顆與128顆無顯著差異
- epoch：2個水準間無顯著差異
- batch：3個水準間皆無顯著差異

```
> #batch40 vs batch160
> #var.test(group1, group2, conf.level = 0.95)
> t.test(group1, group2, conf.level=0.95, var.eual=F)

welch Two sample t-test

data: group1 and group2
t = -1.7876, df = 25.227, p-value = 0.08587
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.0040044036 0.0002821814
sample estimates:
mean of x mean of y
0.004038222 0.005899333

> #batch40 vs batch320
da> #var.test(group1, group3, conf.level = 0.95)
t -> t.test(group1, group3, conf.level=0.95, var.eual=F)
al'
95      welch Two sample t-test
-1
sa|data: group1 and group3
|t = -1.0484, df = 28.151, p-value = 0.3034
0,|alternative hypothesis: true difference in means is not equal to 0
95|percent confidence interval:
-0.0027992388 0.0009035721
sample estimates:
mean of x mean of y
0.004038222 0.004986056


> #batch160 vs batch320
> #var.test(group2, group3, conf.level = 0.95)
> t.test(group2, group3, conf.level=0.95, var.eual=T)

welch Two sample t-test

data: group2 and group3
t = 0.75674, df = 32.897, p-value = 0.4546
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.001542399 0.003368955
```

# 參數調整結果

以下將截取測試集誤差 ( Test\_MSE ) 最小的前10個參數組合：

No.	預測期數	神經元	Epoch	Batch	Test
 1	24	<b>128</b>	50	160	0.003466
2	48	<b>32</b>	50	320	0.003484
3	48	<b>32</b>	100	160	0.003484
4	48	<b>128</b>	100	160	0.003487
5	24	<b>128</b>	50	320	0.003489
6	24	<b>32</b>	50	40	0.00349
7	24	<b>32</b>	50	160	0.00349
8	48	<b>32</b>	50	160	0.003496
9	48	<b>128</b>	100	40	0.003501
10	48	<b>128</b>	50	40	0.003507

MSE<0.0035

# 未來濃度預測

使用者也可利用模型自行預測未來一小時的PM2.5濃度：

```
The concentration of PM2.5 in Hsinchu will be : 24.132246
```

```
#預測新的一筆
X_Future = []

j = (len(inputs)+1)
X_Future.append(inputs[j-period_num:j-1, 0])

X_Future = np.array(X_Future)
X_Future = np.reshape(X_Future, (X_Future.shape[0], X_Future.shape[1], 1))
Y_Future_predict = regressor.predict(X_Future)
Y_Future_predict = sc.inverse_transform(Y_Future_predict)
```

# CONCLUSION

| 05 |

結論與未來展望



# 結論與未來展望



## ALREADY DONE

1. 分析各地的空氣品質資料
2. 使用者可自行預測未來空氣品質



## FUTURE VISION

1. 建立視覺化平台
2. 自動抓取最新資料，結合平台提供預警功能
3. 考慮溫度、風向等多變量對空氣品質的影響



THANK YOU

---

FOR YOUR ATTENTION.

