



「當AI遇上股票」 課程經驗分享

顏永進 / 20231020

from zero to error hero

大綱

1. FAQ
2. 課程背景
3. 課程設計
4. 課程簡介
5. 課程體驗
6. Q&A

1. FAQ

- Q1: 模型準確嗎(能賺錢嗎)?
- A1: 我人不是還在這裡上課嗎...QQ



1. FAQ

- Q2: 不準還弄這個主題幹嘛?

- A2:

1. 純粹希望學生能對AI與理財提高一點興趣，能找到AI在生活中可以帶來收益的實際應用(書中自有黃金屋)
2. 學校教育太過缺乏理財知識
3. 這是我學AI接觸到時間序列後第一個有強烈學習與實作動機的主題
4. 不能只有我一個被騙...-_-

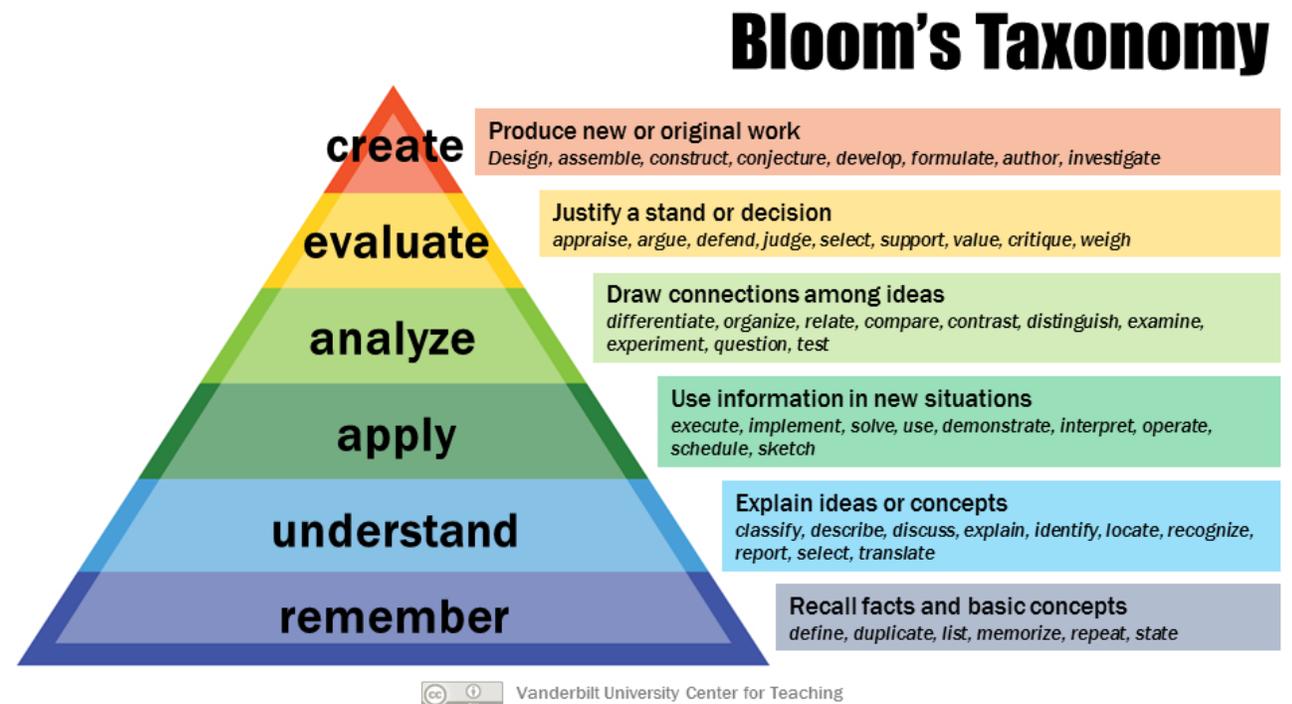
2. 課程背景

- 1) 開設課程的幾個考慮因素
- 2) 學校資訊相關課程
- 3) 授課對象與課程目標

1) 開設課程的幾個考慮因素

2. 課程背景

- 學生具備的先備知識:
 - 有沒有學過程式設計?
 - 有沒有學過python?
- 學生的修課動機:
 - 是被迫參加的必修?
 - 還是自主參加的選修?
 - 或是半自主的學群必修?



2) 學校的資訊相關課程

2. 課程背景

年級	類別	學分	課程
一年級	必修	2	C++程式設計
一年級	選修	2	基礎程式設計
二年級	選修	2	APCS 與演算法
二年級	選修	2	資安實務
二年級	(上)選修	2	Python
二年級	(下)選修	2	AI選修
三年級	必修(電資學群)	4	進階程式設計(Python) ₇

先備知識(上學期PYTHON)

- 希望能具備的能力
 - 熟悉Python基本語法
 - 變數、輸入、輸出
 - 條件判斷、迴圈
 - 熟悉Python特殊的資料型別
 - Function
 - 第三方模組的安裝/管理
 - Numpy, Pandas, Matplotlib
 - 簡單的專題實作

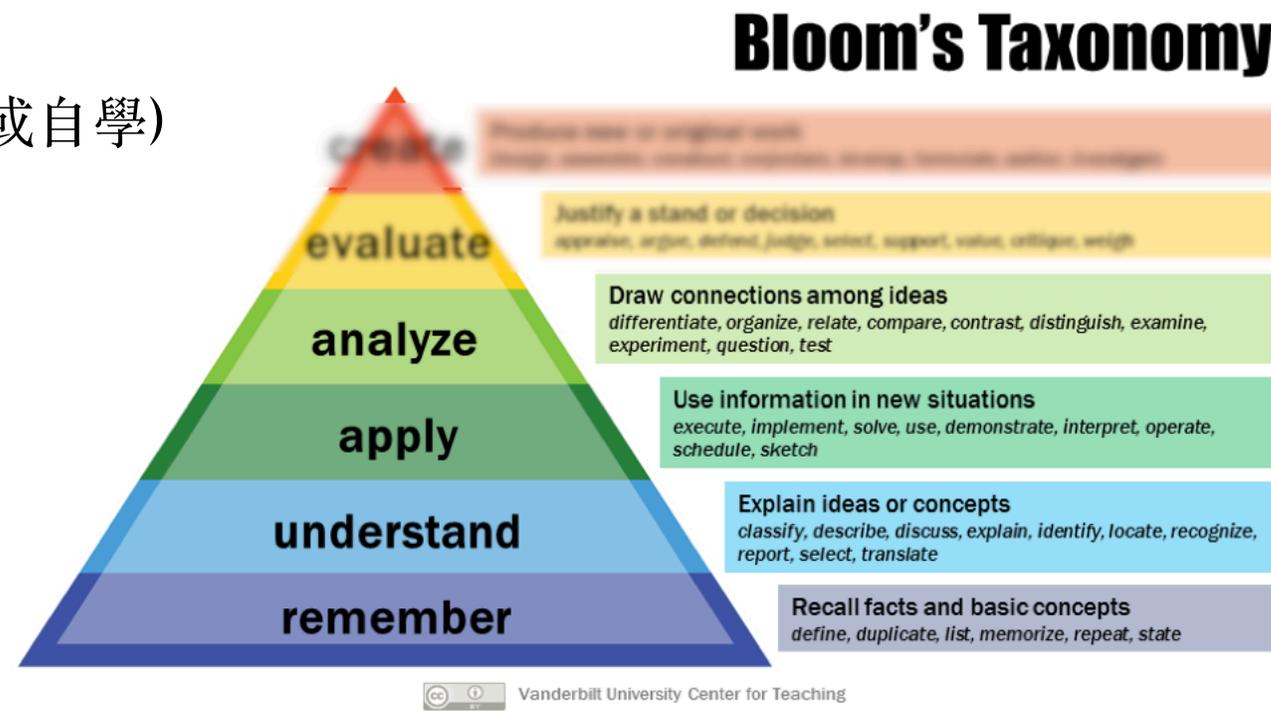
選修：PYTHON

週五	
307	多元選修
1. 課程簡介 2. 變數、輸入與輸出	1. 課程簡介 2. IDE / 輸入輸出
中秋節	
3. 運算式與運算子	3. 運算式與運算子
4. Python內建函數	4. Python內建函數
5. 判斷結構	5. 判斷結構
6. 迴圈	6. 迴圈
7. List / While	7. List / While
8. 巢狀迴圈/List	9. 資料型別
高三資訊科期中考	高二選修期中考
8. 巢狀迴圈/List	10. Function
9. 資料型別	12. Numpy / Matplotlib
9. 資料型別	13. Pandas / Matplotlib
全校第2次定期考查	
10. Function	期末Project
校慶運動會	
高三資訊科期中考	期末Project
第3次定期考查	
	期末Project
	期末Project DEMO/onLine
休業式	

3)授課對象與課程目標

2. 課程背景

- 課程名稱: AI入門
- 對象: 高二下 (已修高二上Python選修或自學)
- 目的:
 1. 理解基本AI運作原理
 2. 進行資料預處理
 3. 挑選合適模型解決問題
 4. 評估效能
 5. 略能解釋模型



3. 課程設計

- 1) 課程設計的三種方法
- 2) 本課程的設計流程
- 3) 教授本課程需要什麼能力



1)課程設計的三種方法

3. 課程設計

- GCPM
- 素養導向課程設計方法
 - Top-Down
 - Bottom-Up



1)課程設計的三種方法: GCPCM

3. 課程設計

- 開設一門選修(AI入門)課的SOP:

1. **GCPCM: Google** 人工智慧導論 課程計畫

2. 1091 學年度 延伸選修通識人工智慧導論 課程計畫

3. **GCPCM: Copy / Paste / Modify**



國立臺南一中__學年度第__學期多元選修課程授課計畫

科目名稱	AI入門	授課教師	顏永進
課程設計 期程	<input type="checkbox"/> 上學期 <input checked="" type="checkbox"/> 下學期 <input type="checkbox"/> 1學年		
開課年段	<input type="checkbox"/> 高一 <input checked="" type="checkbox"/> 高二 <input type="checkbox"/> 高三	預定修課 人數上限	<input checked="" type="checkbox"/> 25人 <input type="checkbox"/> 38人 <input type="checkbox"/> 其他(15 人)
十二年國教 核心素養 (可複選) <input type="checkbox"/> 自主行動 <input type="checkbox"/> 溝通互動 <input type="checkbox"/> 社會參與	<input type="checkbox"/> A1身心素質與自我精進 <input checked="" type="checkbox"/> A2系統思考與解決問題 <input checked="" type="checkbox"/> A3規劃執行與創新應變 <input checked="" type="checkbox"/> B1符號運用與溝通表達 <input checked="" type="checkbox"/> B2科技資訊與媒體素養 <input type="checkbox"/> B3藝術涵養與美感素養 <input type="checkbox"/> C1道德實踐與公民意識 <input type="checkbox"/> C2人際關係與團隊合作 <input type="checkbox"/> C3多元文化與國際理解		
議題融入 課程 (至多3項)	<input type="checkbox"/> 性別平等 <input type="checkbox"/> 人權教育 <input type="checkbox"/> 環境教育 <input type="checkbox"/> 海洋教育 <input type="checkbox"/> 品德教育 <input type="checkbox"/> 生命教育 <input type="checkbox"/> 法治教育 <input checked="" type="checkbox"/> 科技 <input checked="" type="checkbox"/> 資訊 <input type="checkbox"/> 能源 <input type="checkbox"/> 安全 <input type="checkbox"/> 防災 <input type="checkbox"/> 家庭教育 <input type="checkbox"/> 生涯規劃 <input type="checkbox"/> 多元文化 <input type="checkbox"/> 閱讀素養 <input type="checkbox"/> 戶外教育 <input type="checkbox"/> 國際教育 <input type="checkbox"/> 原住民族教育		
大學學群 對應 (至多3項)	<input checked="" type="checkbox"/> 資訊 <input checked="" type="checkbox"/> 工程 <input checked="" type="checkbox"/> 數理化 <input type="checkbox"/> 醫藥衛生 <input type="checkbox"/> 生命科學 <input type="checkbox"/> 生物資源 <input type="checkbox"/> 地球環境 <input type="checkbox"/> 藝術 <input type="checkbox"/> 社會心理 <input type="checkbox"/> 大眾傳播 <input type="checkbox"/> 外語 <input type="checkbox"/> 文史哲 <input type="checkbox"/> 教育 <input type="checkbox"/> 法政 <input type="checkbox"/> 管理 <input type="checkbox"/> 財經 <input type="checkbox"/> 遊憩與運動		
課程目標 (請條列)	1. 瞭解AI基本概念 2. 學習各種常見的AI學習類型		
課程內容 綱要 (請條列授課 主題)	1. 人工智慧簡介 2. 人工智慧相關背景知識 3. 監督式學習 4. 非監督式學習 5. 增強式學習 6. 深度學習		
教材來源	https://www.facebook.com/aik12.edu/		
預期效益	學習人工智慧的基本概念		
評量方式	出席: 10% 課堂表現: 10% 課堂練習作業: 20% 期末project: 60%		

備註

本課程主要以理論探究為主，以python語言實作，只適合初學者，不要幻想在修完本課程後即能變身為人工智慧領域專家，對於人工智慧已有基礎的同學可以不用浪費時間來選課。

教學預定進度表

週次	課程單元	課程內容	備註
1	人工智慧簡介	人工智慧簡介	
2	人工智慧簡介	人工智慧簡介	
3	人工智慧相關背景知識	人工智慧相關背景知識	
4	人工智慧相關背景知識	人工智慧相關背景知識	
5	監督式學習	監督式學習	
6	監督式學習	監督式學習	
7	非監督式學習	非監督式學習	
8	非監督式學習	非監督式學習	
9	增強式學習	增強式學習	
10	增強式學習	增強式學習	
11	深度學習	深度學習	
12	深度學習	深度學習	
13	人工智慧實作練習： Hello world	人工智慧實作練習：Hello world	
14	Project實作#1	Project實作#1	
15	Project實作#2	Project實作#2	
16	Project實作#3	Project實作#3	

進度規劃說明

- 高一、二上、下學期課程：請規劃16週。
- 高三上、下課程：請規劃16週（高三下實際上課週數為10-12週）。

成果

1) 課程設計的三種方法: Top-Down

1. 選擇課程主題 (單元)
2. 挑選核心素養
3. 單元目標 (核心素養+學科本質)
4. 分析起點行為
5. 設想連結的情境脈絡
6. 選定學習重點
7. 設計學習評量

1)課程設計的三種方法: Buttom-Up

- UbD: Understanding by Design (重視理解的課程設計)
- 逆向設計(Backward Design)：反向思考
 1. 先確定**預期的學習成果(學習目標)**
 2. 針對學生應該如何表現出對這門課的理解**(學生該具備何種能力)**，去**設計測驗、評量方式**，
 3. 最後才思考**教學活動內容**。



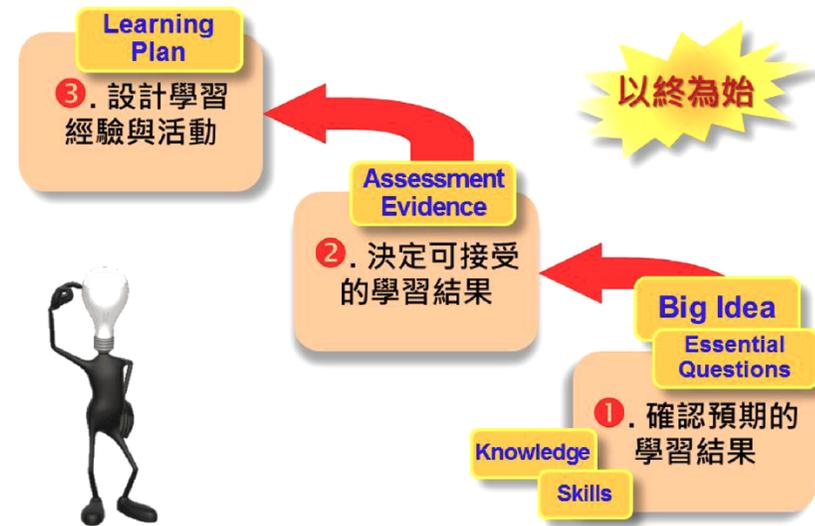
UbD是什麼？

1)課程設計的三種方法: Bottom-Up



2)本課程的設計流程

- UbD課程設計三階段
 - 階段1: 預期的學習結果
 - 階段2: 評量的證據
 - 階段3: 學習計畫



2)本課程的設計流程

- 階段1: 預期結果

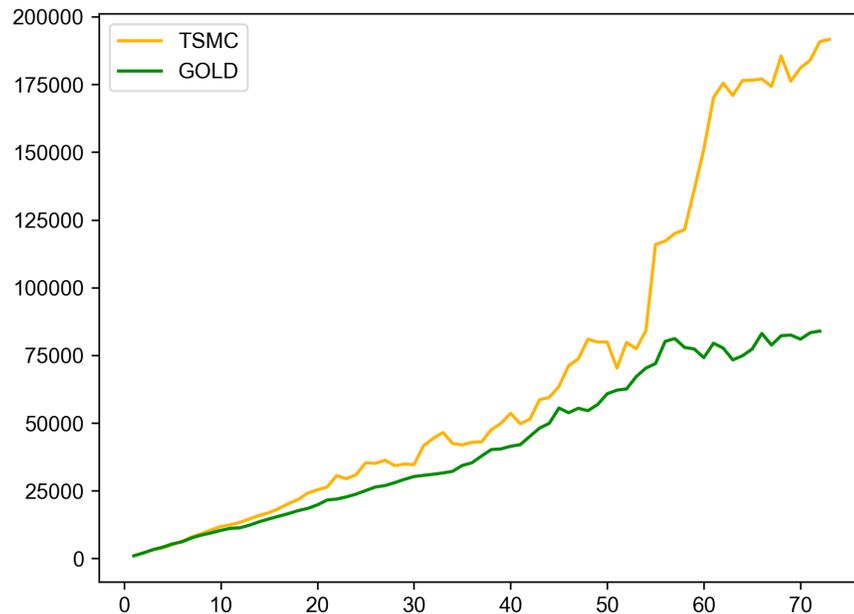
1. 學生能利用傳統工具規畫投資
2. 學生能將AI模型應用股票漲跌預測
3. 學生能修改模型、評估模型效能
4. 學生能將AI模型應用於其他與生活相關的實際事物中



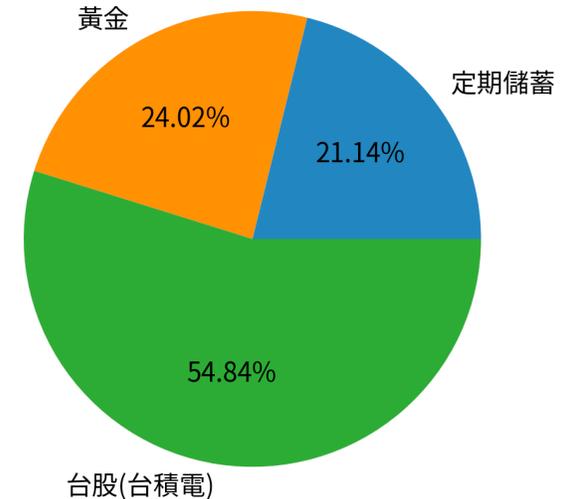
2)本課程的設計流程

階段1: 預期結果

	A	B	C	D	E	F	G	H
1	月份	日期	時間	買入	賣出	資金	購入單位	累計單位
2	1	0105-01-04	09:03:23	1,121	1,133	1000	0.8921	0.892
3	2	0105-02-01	09:02:23	1,198	1,213	1000	0.8347	1.726
4	3	0105-03-01	09:01:45	1,316	1,330	1000	0.7599	2.486
5	4	0105-04-01	09:02:03	1,272	1,286	1000	0.7862	3.272
6	5	0105-05-03	09:01:44	1,330	1,344	1000	0.7519	4.024
7	6	0105-06-01	09:01:43	1,270	1,283	1000	0.7874	4.812
8	7	0105-07-01	09:03:45	1,362	1,377	1000	0.7342	5.546
9	8	0105-08-01	09:03:24	1,368	1,382	1000	0.7310	6.277
10	9	0105-09-02	09:03:45	1,331	1,345	1000	0.7513	7.028
11	10	0105-10-03	09:04:24	1,319	1,333	1000	0.7582	7.786
12	11	0105-11-01	09:03:25	1,287	1,301	1000	0.7770	8.563
13	12	0105-12-01	09:03:44	1,195	1,208	1000	0.8368	9.400
14	13	0106-01-03	09:03:25	1,192	1,206	1000	0.8389	10.239
15	14	0106-02-02	09:03:24	1,216	1,229	1000	0.8224	11.061
16	15	0106-03-01	09:03:24	1,220	1,232	1000	0.8197	11.881
17	16	0106-04-05	09:03:42	1,220	1,229	1000	0.8197	12.701
18	17	0106-05-02	09:03:25	1,212	1,225	1000	0.8251	13.526
19	18	0106-06-01	09:03:21	1,219	1,232	1000	0.8203	14.346
20	19	0106-07-03	09:03:25	1,204	1,218	1000	0.8306	15.177



總金額: 349482, 投報率: 61.80%



2)本課程的設計

• 階段1: 預期結果

```
[ ] 1 df = yf.Ticker('2330.TW').history(period='10y')
2 data = df.filter(['Open', 'High', 'Low', 'Volume', 'Close'])
3
4 scalerX = MinMaxScaler(feature_range=(0, 1))
5 scalerY = MinMaxScaler(feature_range=(0, 1))
6 all_x = data[['Open', 'High', 'Low', 'Volume', 'Close']]
7 all_y = data['Close']
8 scal_all_x = scalerX.fit_transform(all_x.values)
9 scal_all_y = scalerY.fit_transform(all_y.values.reshape(-1, 1))
10
11 previousNDays = 10
12 x_data, y_data = [], []
13 for i in range(len(scal_all_x) - previousNDays):
14     x = scal_all_x[i:i+previousNDays]
15     y = scal_all_y[i+previousNDays]
16     x_data.append(x)
17     y_data.append(y)
18 x_data, y_data = np.array(x_data), np.array(y_data)
19
20 x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_s
```

效能奇差，沒辦法用以預測股價。可能的理由：

1. 卷積核大小1
2. 模型層數太淺
3. 數據集不是隨機分割
4. 採用特徵不夠全面

修正以上的問題之後應該能夠改進。

我的版本：

將考量的特徵增加到5個，兩層卷積六層全連接層。

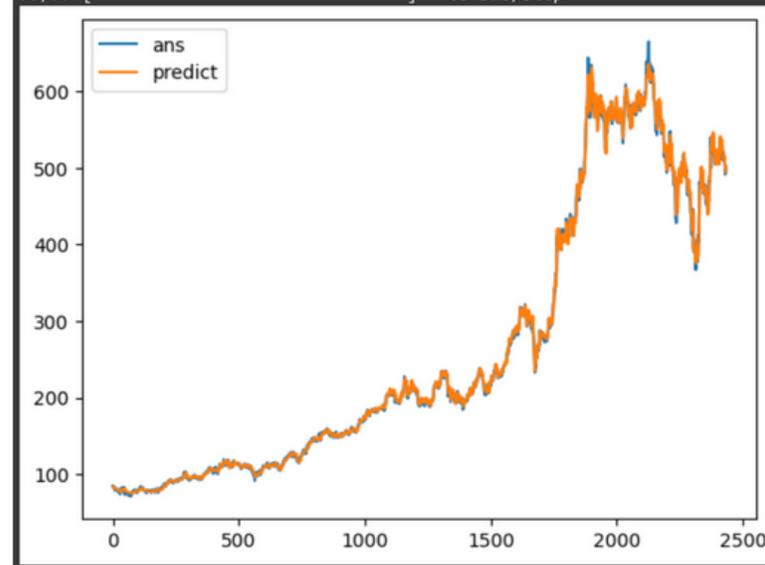
資料集改用sklearn提供的方法分割，取30%為測試集。

共訓練400個epoch，取其中最好的一次迭代模型，達成測試集上均方差 $9.7103e-5$ 。

下圖為針對全部數據的預測(非僅限於測試集)，折線圖才能正常顯示，並能看出整體的預測與實際值的走勢。

相比實際數據，在股價的局部極值，AI預測較為保守，所以會略為失準

```
23/23 [=====] - 0s 4ms/step - loss: 9.7103e-05 - mean_squared_error: 9.7103e-05
MSE: 9.710263111628592e-05
77/77 [=====] - 0s 3ms/step
```



2)本課程的設計流程

- 階段1:
預期
結果

The screenshot displays a video analysis application interface. On the left, a video frame shows a person in a white t-shirt and blue shorts standing in a hallway, enclosed in a green bounding box. Text above the bounding box reads "height_ratios: 90.83 %". In the top left corner, it says "Camera 專題展示影片" and "detected num: 2". On the right, a code editor shows Python code for object detection and height ratio calculation. Below the code, a terminal window displays the execution of the Python script. A notification box in the bottom right corner indicates a change in the number of detected people: "人數變動 偵測到的人數增加1人,且有人出現在相當接近處,偵測人數2人。"

2)本課程的設計流程

- 階段2: 評量方式

1. 理財達人賽獲利(依排名評分)
2. 股票走勢預測準備性(依排名評分)
3. 模型的進化歷程(各組修正不同版本的記錄)
4. 專題實作:
 - 專題創意、及實用價值
 - 模型應用、修訂以及效能評估
 - 對專題架構與內容的熟悉程度

2)本課程的設計流程

• 階段3: 學習活動

1. AI基本概念: 各種常見的AI學習類型及基本原理。
2. AI模型的實作練習
3. 投資與理財
4. 股票漲跌影響因素分析
5. CNN, RNN, LSTM簡介及股價預測實作
6. 資料處理: 整理、擴增



3)教授本課程需要什麼能力

3. 課程設計

- 基本的Python語法
- Python套件使用經驗: Numpy, Matplotlib, PANDAS...
- 略懂AI運作原理
- AI領域的Hello world(MNIST)使用經驗
- 能使用Google / YouTube

4. 課程簡介

- 1) 版本1: 起源
- 2) 版本2: 加入現有選修課程
- 3) 版本3: 新興科技計畫
- 4) 課程內容
- 5) 專題成果發表

1) 版本1: 起源

padlet



<https://tiny.one/ai-rich>

sli.do



<https://tiny.one/ai-rich2>

科技力X雙語X跨域教學示例

from zero to error hero

顏永進 / 1229

版本1: 大綱

如果當時我懂的投資...

- 1) Investing - What and Why
- 2) Investing - How
- 3) 理財達人團體賽
- 4) Get Hands Dirty: DEMO
- 5) Get Your Hands Dirty
- 6) 分組報告

如果AI能幫我投資...

- 1) What can AI do?
- 2) An introduction to AI
- 3) CNN實作練習
- 4) Let's Get Greedy
- 5) Let's Be More Realistic

3. 理財達人團體賽 (Financial Talent Team Competition)

- 背景
 - 假設從六年前開始養成每月定期投資的習慣，現在的你損益情形如何？
 - 體驗經歷疫情與戰爭的投資成果
- 活動內容
 - 教師介紹幾種常見的投資商品
 - 示範如何由網路蒐集所需資訊
 - 示範如何以試算表計算六年投報率
 - 示範如何將資料視覺化
 - 學生實作、挑選投資商品、計算損益、繪製圖表

目標

- 成為賺最多的人，或
- 成為賠最多的人

Competition Rules

1. 投資期限：<2016-01-01 Fri> to <2021-12-31 Fri>
2. 分三種投資項目，每個項目每月投入1000元
3. 三種投資項目中需包含至少一檔台股
4. 股票配股配息併入下月購買資金，只要基於真實數據均可列入計算(各股配息資訊需自行搜尋)
5. 假設所有股票均提供零股買賣，且零股可以小數點買進(現實中股票最低買賣單位為1股)

Competition Rules

6. 假設零股交易價格與一般交易相同(現實中零股買賣價格不同於市價)
7. 選擇其他投資商品如基金、期貨、選擇權、政府公債者需請自行研究獲利計算規則
8. 六年中只能於每月 **第一個交易日** 買進投資商品 (不可短進短出)
9. 不考慮手續費、證交稅等因素
10. 只能選擇 **可持續交易六年** 之投資商品

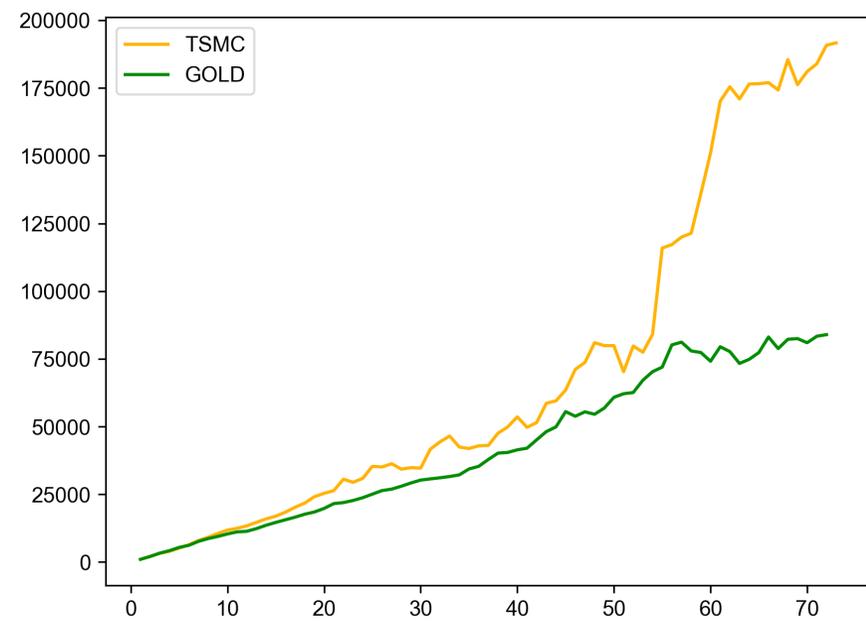
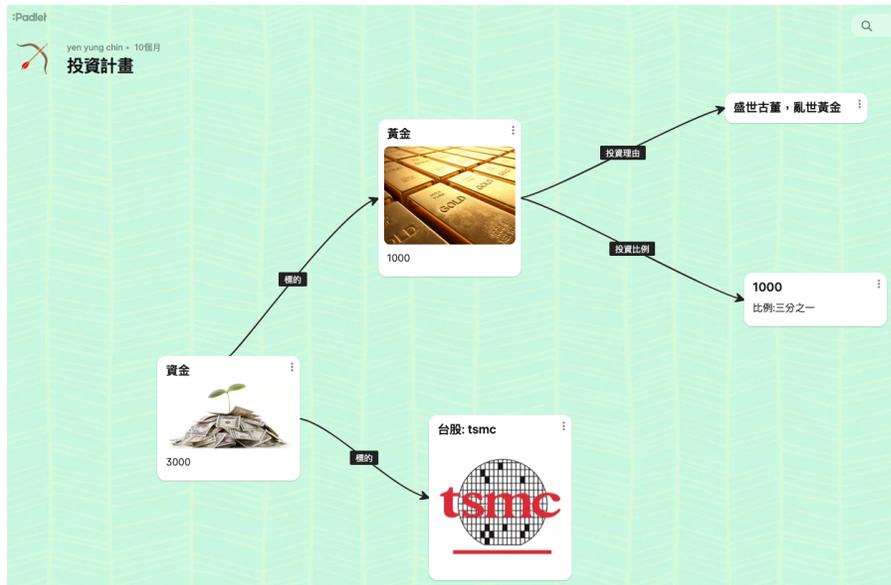
Group Tasks

- 小組討論投資項目
- 收集六年中每月資訊
- 製表
- 計算每項獲利
- 合併組員數據計算全部獲利
- 繪圖

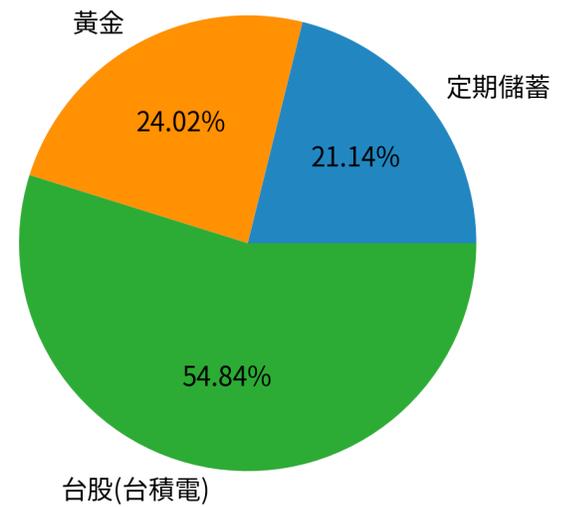
4. Get Hands Dirty: DEMO

DEMO

分組報告



總金額: 349482, 投報率: 61.80%



	A	B	C	D	E	F	G	H	I
1	月份	日期	時間	買入	賣出	資金	購入單位	累計單位	累計財富
2	1	0105-01-04	09:03:23	1,121	1,133	1000	0.8921	0.8921	1010.7047
3	2	0105-02-01	09:02:23	1,198	1,213	1000	0.8347	1.7268	2094.5904
4	3	0105-03-01	09:01:45	1,316	1,330	1000	0.7599	2.4867	3307.2626
5	4	0105-04-01	09:02:03	1,272	1,286	1000	0.7862	3.2728	4208.8557
6	5	0105-05-03	09:01:44	1,330	1,344	1000	0.7519	4.0247	5409.2060
7	6	0105-06-01	09:01:43	1,270	1,283	1000	0.7874	4.8121	6173.9351
8	7	0105-07-01	09:03:45	1,362	1,377	1000	0.7342	5.5463	7637.2865
9	8	0105-08-01	09:03:24	1,368	1,382	1000	0.7310	6.2773	8675.2520
10	9	0105-09-02	09:03:45	1,331	1,345	1000	0.7513	7.0286	9453.5097
11	10	0105-10-03	09:04:24	1,319	1,333	1000	0.7582	7.7868	10379.7802
12	11	0105-11-01	09:03:25	1,287	1,301	1000	0.7770	8.5638	11141.4812
13	12	0105-12-01	09:03:44	1,195	1,208	1000	0.8368	9.4006	11355.9281
14	13	0106-01-03	09:03:25	1,192	1,206	1000	0.8389	10.2395	12348.8719
15	14	0106-02-02	09:03:24	1,216	1,229	1000	0.8224	11.0619	13595.0718
16	15	0106-03-01	09:03:24	1,220	1,232	1000	0.8197	11.8816	14638.0936
17	16	0106-04-05	09:03:42	1,220	1,229	1000	0.8197	12.7012	15609.8259
18	17	0106-05-02	09:03:25	1,212	1,225	1000	0.8251	13.5263	16569.7470
19	18	0106-06-01	09:03:21	1,219	1,232	1000	0.8203	14.3467	17675.0958
20	19	0106-07-03	09:03:25	1,204	1,218	1000	0.8306	15.1772	18485.8703

3. 監督式學習

- 上課投影片 張貼日期
- [課堂練習]迴歸 截止日期：3月10日
- 決策樹課堂作業 截止日期：3月27日

2. 背景知識

- 上課投影片 張貼日期
- 資料預處理 張貼日期
- 相關係數測驗 截止日期：2月25日
- [個人作業]波士頓房價分析 截止日期

1. AI簡介

- 上課投影片 張貼日期
- Chapter 1 作業 張貼日期

6. 小組專題實作

新增作業後，學生就會看到這個主題

5. 深度學習

- [講義]深度學習 上次編輯時間：4月28日
- [講義]CNN實作教學 上次編輯時間：4月28日
- CNN實作#1 上次編輯時間：4月21日
- CNN實作#2 上次編輯時間：4月28日
- [講義]當AI遇上股票 上次編輯時間：4月28日
- [作業]當AI遇上股票-CNN-1 張貼日期：4月28日
- [小組作業]當AI遇上股票 上次編輯時間：5月26日
- 理財講座回饋單 張貼日期：4月26日

4. 非監督式學習

- 階層式分群的實作 截止日期：3月31日 晚上11:59

3) 版本3: 新興科技計畫

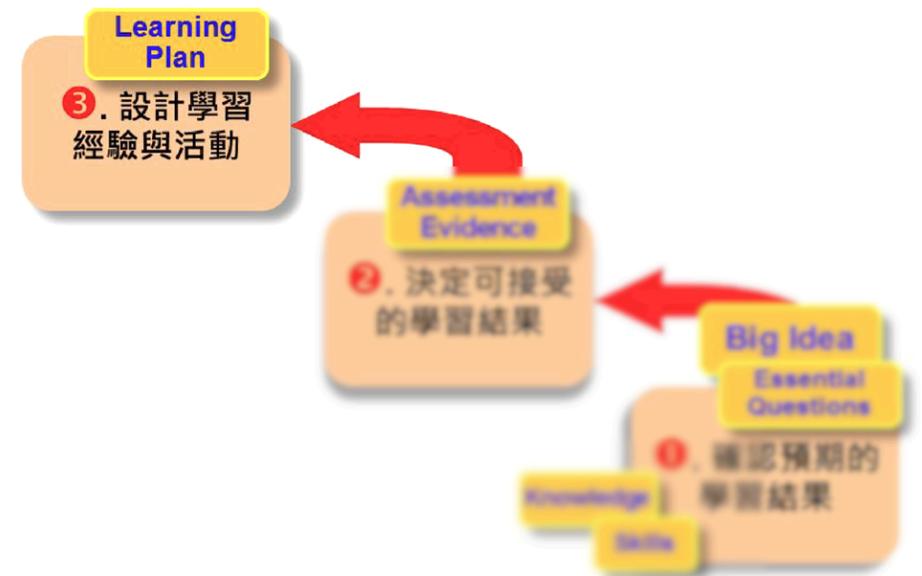
1) 主題樹連結：人工智慧預測股市動向為重心，為主題樹中的進階端課程。



3) 版本3: 新興科技計畫

• 階段3: 學習活動

1. AI基本概念: 各種常見的AI學習類型及基本原理。
2. AI模型的應用練習
3. 投資與理財(校內家政科教師)
4. 股票漲跌影響因素分析(外聘講師)
5. CNN, RNN, LSTM
6. 簡介及股價預測實作
7. 資料處理: 淨化、擴增



臺南一中 當AI遇上股票 (進階)

教學活動內容	時間
單元一 投資與理財	200分鐘
單元二 AI簡介	200分鐘
單元三 監督式學習	200分鐘
單元四 非監督式學習	100分鐘
單元五 深度學習	100分鐘
單元六 股價預測實作	400分鐘
單元七 專題實作	400分鐘

臺南一中 當AI遇上股票（進階）

- 教案大綱與內容：
- 單元一 投資與理財：瞭解何謂投資及可行手段，利用數據來說明投資理財觀念的重要性，並分組實作競賽。

教學活動內容	時間
介紹投資的方式，了解為什麼需要投資，並以常見投資商品、六年投報率、視覺化圖表等所需資料進行實作競賽。	200分鐘

臺南一中 當AI遇上股票 (進階)

- 教案大綱與內容：
- 單元二 AI簡介：認識及瞭解AI基本概念，學習各種常見的AI學習類型，並簡介神經網路運作原理。

教學活動內容	時間
介紹人工智慧的起源、發展歷程、人工智慧的類型、應用現況以及未來可能發展方向。	200分鐘

臺南一中 當AI遇上股票（進階）

- 教案大綱與內容：
- 單元三 監督式學習：藉由圖片分類辨識，引領學生認識特徵值、分類器基本觀念，**瞭解三種分類器之運作原理及相關應用**（最短距離分類器、KNN分類器、決策樹分類器）

教學活動內容	時間
活動主旨：藉由圖片分類辨識，引領學生認識特徵值、分類器基本觀念，進而瞭解三種分類器之運作原理及相關應用（最短距離分類器、KNN分類器、決策樹分類器）。	200分鐘

臺南一中 當AI遇上股票（進階）

- 教案大綱與內容：
- 單元四 非監督式學習：瞭解非監督式學習之分群問題，**瞭解兩種不同的分群演算法**（k-means演算法、階層式分群演算法）

教學活動內容	時間
藉由前一章節監督式學習帶入非監督式學習之概念，並比較其差異性。爾後引領學生瞭解非監督式學習之分群問題，並進而瞭解兩種不同的分群演算法（k-means演算法、階層式分群演算法）。	100分鐘

臺南一中 當AI遇上股票（進階）

- 教案大綱與內容：
- 單元五 深度學習：深度學習的概念，將待解問題數學化。

教學活動內容	時間
<p>探討深度學習的概念，結合類神經網路的概念，在訓練電腦進行分類概念的學習時，運用了分層次抽象的想法，這個想法認為更高層次的概念可以從低層次的概念學習到，所以會在輸入與輸出層中增加隱藏層的層數達到這一目的。而各個層次的運作機制是主要學習重點。</p>	100分鐘

臺南一中 當AI遇上股票（進階）

- 教案大綱與內容：
- 單元六 股價預測實作：CNN基本架構、運作原理及實作，股票漲跌影響因素分析，RNN/LSTM簡介及股價預測實作

教學活動內容	時間
綜合前節CNN之概念，帶領學生從頭開始實作神經網路之重要步驟，並提供學生作業實作，如：挑選投資商品、計算損益、繪製圖表等。	400分鐘

臺南一中 當AI遇上股票（進階）

- 教案大綱與內容：
- 單元七 專題實作：期末專題分組實作。

教學活動內容	時間
分組合作完成AI相關專題成品	400分鐘

4) 課程內容

4. 課程簡介

- 1) 設計理念
- 2) 教學活動/目標
- 3) 跨域學習內容
- 4) 課程評量方式
- 5) 教材內容(簡略版本)



設計理念

- 天下攘攘，皆為利往。天下熙熙，皆為利來。《六韜》 《史記·貨殖列傳》
- 倉廩實而知禮節，衣食足而知榮辱。《管子·牧民》
- 用錢能解決的都不是問題，問題是沒錢。《俗話》
- 高雄發大財... --

教學活動/目標

1. Way to Become Rich

- 了解何謂投資及可行手段
- 利用試算表模擬六年間固定投資特定目標的獲利，利用數據來說明投資理財觀念的重要性。

2. Can AI Make Everyone Rich?

- 簡介神經網路運作原理
- 實作CNN模型，了解不同設計架構對於CNN模型的影響
- 實際測試以神經網路預測股價，從中學習AI運作原理及其侷限性。

跨域學習內容

- 資訊科技
 - 資T-IV-2: 資訊科技應用專題
 - 運c-V-3: 能整合適當的資訊科技與他人合作完成專題製作。
 - 運p-IV-1: 能選用適當的資訊科技組織思維，並進行有效的表達。
 - ※運r-V-1: 能將問題以運算形式呈現。
- 綜合活動領域家政科
 - 家Ca-IV-1: 個人與家庭生活的金錢及時間管理。
 - 家3a-V-2: 涵養財金素養，整合消費能力，妥善規劃個人財務生活

評量

1. Way to Become Rich

- 投資目標挑選依據
- 六年投資報酬率

2. Can AI Make Everyone Rich?

- CNN模型圖片辨識準確率
- 股票預測模型準確率

1. WAY TO BECOME RICH

- 1) Investing - What and Why
- 2) Investing - How
- 3) 理財達人團體賽
- 4) Get Hands Dirty: DEMO
- 5) Get Your Hands Dirty
- 6) 分組報告

課程說明

1. 學完這門課，你
 - 不會變成投資達人
 - 不會變有錢
 - 不會變成AI專家
2. 但是你會
 - 了解投資理財的重要性
 - 也許會開始行動
 - 可能會覺得AI很有趣，將來會花時間去研究它

2. Investing - How

- 1) 股票(Stock)
- 2) 基金(Fund)
- 3) ETF(Exchange Trade Funds)
- 4) 債券(Bond)
- 5) 外匯(FX, Forex, Foreign Exchange)
- 6) 期貨(Futures)
- 7) 黃金(Gold investment)
- 8) 儲蓄(Saving)

1) 股票(Stock)

- 一間公司為了籌措資金，把公司其中一部分的所有權讓出來的憑證。
- 公司為什麼要發行股票： 缺錢
- 為什麼會缺錢
 - 經營不善
 - 擴大規模
- 缺錢怎麼辦
 - 找銀行借錢-要付利息
 - 發行股票，找韭菜人投資

股票要怎麼賺錢？

1) 股票

股票賺錢方式

- 賣掉股票賺價差
- 領取股息 (股票股利→配股 或 現金股利→配息)

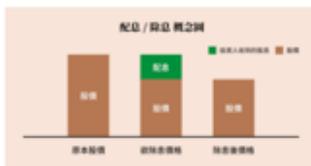
假設股票如果配1元股利，代表每1股可以分到1元的現金，但配息時公司股價也會減少1元。

其實，領取股息就是左手換右手，並不是投資賺錢的來源。

股票怎麼賺錢

1) 股票

- 除息
- 就是配發現金股利
- 除息後股價 = 除息前股價 - 現金股利(元)
- 台積電股價100元，每股配3元現金股利，除息後股價 = $100 - 3 = 97$ 元，



圖表來源: [HTTPS://YALECHEN.COM/US-STOCK-DIVIDEND/](https://yalechen.com/us-stock-dividend/)

股票怎麼賺錢

1) 股票

- 除權
- 就是發放股票股利
- 除權後股價 = 除權前股價 / (1 + 股票股利(元) / 10)
- 台積電股價100元，要配 5元股票股利，相當於每 1股配 0.5股股票股利，
除權後股價 = $100 / (1+5/10) = 66.67$ 元
- 除權前，你有1000股台積電，股價100元，市值10萬元。
- 配股後，你有1500股台積電，股價66.67元，市值10萬。

美股：現金股利(Cash Dividend)

1) 股票

- 美股各股資訊查詢: <https://seekingalpha.com/>

美國股市	配息頻率	2020年配息 (1股/美元)
蘋果公司 APPLE	每季	0.8075
微軟公司 Microsoft Corp	每季	2.09
英特爾股份有限公司 Intel Corp	每季	1.32
QQQ Invesco QQQ ETF	每季	1.7365
DIA SPDR Dow Jones Industrial Average ETF	每月	5.7274
ARKK ARK Next Generation Internet ETF	每季	1.8948

股票配息

- 每股 30% 股息 → 股息稅
- 剩下 70% 股息 → 匯投資者帳戶

圖表來源: [HTTPS://YALECHEN.COM/US-STOCK-DIVIDEND/](https://yalechen.com/us-stock-dividend/)

美股：股票分割(Stock Split)

1) 股票

表2：蘋果公司歷年股票分割記錄表

日期	分割內容	分割後累積股數
1987年6月	1股分為2股	2
2000年6月	1股分為2股	4
2005年2月	1股分為2股	8
2014年6月	1股分為7股	56
2020年8月	1股分為4股	224

來源：什麼是股票分割？以蘋果與特斯拉為例/余家榮（效率理財王）·理財作家

1) 股票



來源：什麼是股票分割？以蘋果與特斯拉為例/余家榮（效率理財王）·理財作家

美股要怎麼買？

1) 股票

方法一：海外券商開戶

美股券商下單手續費較低，以出入金費用、電匯費為主要成本。適合在意交易手續費的人。

方法二：權委託

透過國內的券商去海外幫你下單買進股票或ETF，適合不放心錢放海外的人使用。

1. Investing - What and Why

1. What is Investing? Investing 101: Easy Peasy Finance for Kids and Beginners
2. Inflation Definition for Kids
3. 通貨膨脹:為什麼需要投資

2. Investing - How

- 1) 股票(Stock)
- 2) 基金(Fund)
- 3) ETF(Exchange Trade Funds)
- 4) 債券(Bond)
- 5) 外匯(FX, Forex, Foreign Exchange)
- 6) 期貨(Futures)
- 7) 黃金(Gold investment)
- 8) 儲蓄(Saving)

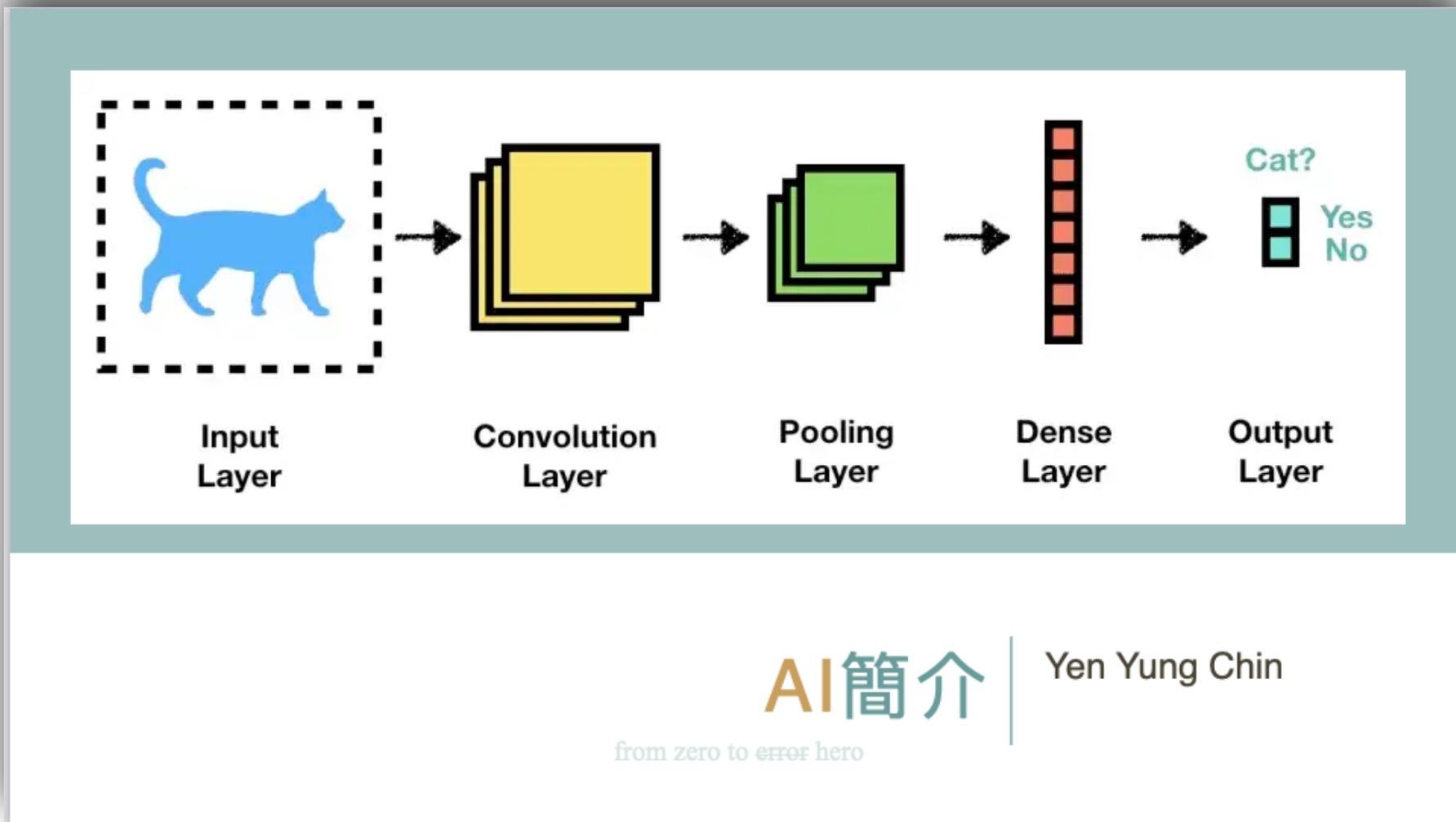
3. 理財達人團體賽 (Financial Talent Team Competition)

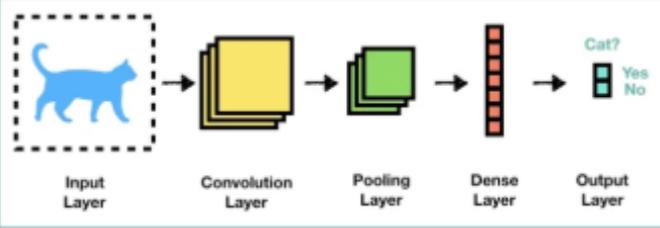
- 背景
 - 假設從六年前開始養成每月定期投資的習慣，現在的你損益情形如何？
 - 體驗經歷疫情與戰爭的投資成果
- 活動內容
 - 教師介紹幾種常見的投資商品
 - 示範如何由網路蒐集所需資訊
 - 示範如何以試算表計算六年投報率
 - 示範如何將資料視覺化
 - 學生實作、挑選投資商品、計算損益、繪製圖表

CAN AI MAKE EVERYONE RICH?

1. An introduction to AI
2. 監督/非監督/強化/深度學習
3. CNN架構及運作原理: 深度學習神經網路
4. CNN實作股價預測
5. Let's Get Greedy: RNN, LSTM
6. Let's Be More Realistic

1. An Introduction to AI





AI簡介 | Yen Yung Chin

From zero to error here

1

課程大綱

- 概念簡介
- 各主題簡單理論/實作
- 期末專題(期中前提出主題)
- 期末專題發表(線上直播、開放教師、同學觀看、接受線上提問)

2

AI做為研究主題的優勢與劣勢

- 優勢
 1. AI的學習門檻低，只要能寫程式，幾個月就能上手
 2. AI做為黑盒子，只玩應用算是做專題的捷徑，不懂也能用
- 劣勢
 1. 沒有深入的研究，只能玩皮毛
 2. 要精通很困難

3

朋友以為我在做



4

我媽以為我在做



5

鄰居以為我在做



6

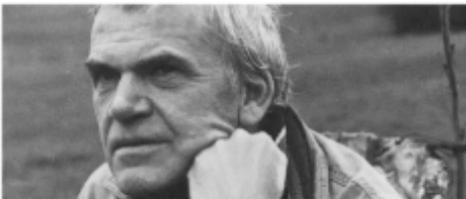
其實我在做



當我說我在研究深度學習

關於思考的兩種反應

- 文青：Man thinks; God laughs.



ooo

關於思考的兩種反應

- 理工人：Can machine think?



ooo

CAN A MACHINE THINK?

Vol. LIX. No. 226. [October, 1950]

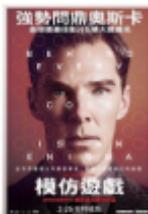
MIND

A QUARTERLY REVIEW
OF
PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND INTELLIGENCE

By A. M. TURING

I. The Imitation Game.
I propose to consider the question, "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think". The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words "machine" and "think" are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, "Can machines think?" is to be found in a statistical sense such as the following. — But this is



- 1950 A. M. Turing

10

A. M. TURING

- 人工智慧之父
- 於1950年發表了一篇在人工智慧領域深具里程碑意義的論文 (Computing machinery and intelligence)。該文旨在探討「機器能否思考」的問題，並提出一套評估機器是否具備人類智慧的方式：圖靈測試 (Turing Test)
- 圖靈認為與其寫程式來模擬成人的心智，不如寫程式去模擬兒童的心智並給予適當的教育，讓這個程式兒童能成長為成人。
- 他同時也是二戰中協助英軍破解納粹密碼的戰爭英雄
- 2021年起他的肖像也出現在面值50元的英鎊，與之競爭這個肖像位置的還有物理學家霍金、英國首相柴契爾夫人、科學家富蘭克林。

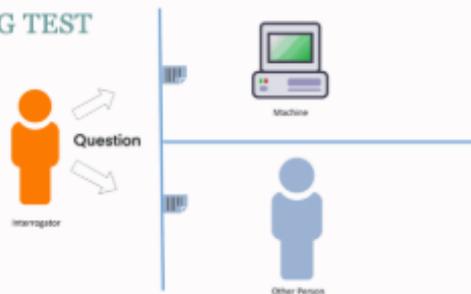
11

模仿遊戲



12

TURING TEST



圖片來源: <https://biturl2335.medium.com/a-brief-history-of-the-turing-test-77c9488897>

13

AI DEMO

- [chatGPT](#)
- [Dall E 2](#)
- [Beethoven X: The AI Project](#)
- [Teachable Machine](#)

14

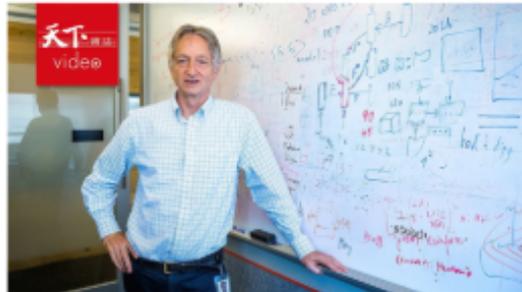
CHATGPT

15

HISTORY OF AI DEVELOPMENT



HINTON



TYPES OF AI

人工智慧 (Artificial Intelligence)

- 強人工智慧
- 弱人工智慧

機器學習 (Machine Learning)

- 監督式學習
- 非監督式學習
- 半監督式學習
- 強化學習

深度學習 (Deep Learning)

- CNN
- RNN
- DNN

1950's 1960's 1970's 1980's 1990's 2000's 2010's

傳統程式設計V.S.機器學習



19

AI在各領域的應用

- 人工智慧在台灣：產業轉型的契機與挑戰講AI在各個領域的實際應用
- 作者：陳昇璋, 溫怡玲
- 出版社：天下雜誌



20

AI選修: 背景知識

- AI於製造業的應用: 瑕疵檢測
 - 某紡織廠在織完一卷布後就由生產線同仁以肉眼檢查是否織的平整、織線有無扭曲變形。
 - 在料染色後，再由另一批同仁以肉眼檢測是否染整均勻。
 - 人工瑕疵檢測配有自動光學檢測(Automatic Optical Inspection, AOI)設備進行，但是仍屬於第二波AI技術，靠的是人類的規則
 - 導入深度學習後，不僅準確率更高，檢測速度更快。
 - 人員目測的瑕疵漏網率為5%，AI為0.01%；人員每天目測30萬張，一台十萬左右的電腦每日可測1,440萬張，效率提升40倍。



21

AI選修: 背景知識

- AI於製造業的應用: 自動流程控制
 - 在生產過程中，產品品質、良率及生產速度會受到許多參數影響：馬達轉速、輪軸速度、電流、電壓、環境變數（風速、溫度、濕度），需要調整的參數可能多達上千。
 - 數個參數若未精準調好，會造成生產效率的低落及良率的降低，而且調整參數後可能要再等四、五個小時才會知道結果是否符合預期目標。
 - 以機器學習來學習設備參數、環境、生產條件、目標之間的關聯性，當設備、環境、生產條件或目標其中一項改變時，就可以透過模型找出最佳參數，大幅縮短因試誤而浪費的時間及生產成本。
 - 在化工製程中，透過機器學習來模擬參數的調控，可以在對的時間提示製程工程師調整參數，最高可以將良率由原先的六成提升到98%，改善幅度超過五成。



22

AI選修: 背景知識

- AI於製造業的應用: 預測性維護
 - 生產設備隨時有可能會故障，有些設備的失效成本很高，若在運轉狀態中壞掉，將導致線上所有原料及半成品的消耗，造成良率降低。
 - 預測性維護率涉到訊號處理，例如要預測馬達是否將要故障，可以透過馬達的電壓、轉速、震動及聲音來判別；要預測機械手臂是否已有故障徵兆，除了上述訊號，還可以參考手臂的行程順暢度、夾具的穩定度、夾角的精確度。
 - 在某空氣壓縮機廠的合作中，藉由蒐集資料及建立深度學習模型，可以準確預測一小時後的空氣壓縮機溫度，可以在壓縮機過熱前，提早預知並爭取一個小時的緩衝時間，減少其工作量，防止過熱損壞而影響產線運作。



23

AI選修: 背景知識

- AI於製造業的應用: 原料組合最佳化
 - 一張印刷電路板上有數百電子元件，每個元件有不同供應商、等級、價格，如何挑選才能在成本與價值(良率、穩定性)間找到平衡？
 - 以染整業為例，客戶要在特定布料上染出指定顏色並指定固色性及染色性，目前的作法是先以打色模擬軟體計算第一次染料配方，由老師傅實驗、再反覆修正配方、實驗。平均要經歷三到七天的打色嘗試。
 - 以深度學習建立模型來描述布料、目標顏色及染料溫度間的關係，可以將第一次打色的成功率由原本的七成(軟體模擬加師傅經驗調整)提升到九成五



24

找題目前題

情境

1. 先確定AI優勢



專題研究

找題目前題

專題研究

2. 問對問題
 - 你從沒有答案的問題，AI也不會，例如假設性的問題。
 - 套用運算思維的解題策略：掌握問題的可運算性。
3. 跨域程度決定題目的CP值
 1. 防手震設計A
 2. 防手震設計B

防手震設計A



專題研究

000

防手震設計B

Smart Spoon



28

题目的靈感來源

專題研究

- 從生活經驗著手
 - 香港中文大學的研究：學生相關性(Student Relevance)在學習AI時非常重要。(學哪個主題時不是這樣?)
 - 生活經驗: AI店長、學生坐姿
- 跨域:
 - 學生自己專長的領域
 - 有興趣的事物-漫畫

29

人工智能教育高中營隊

成果



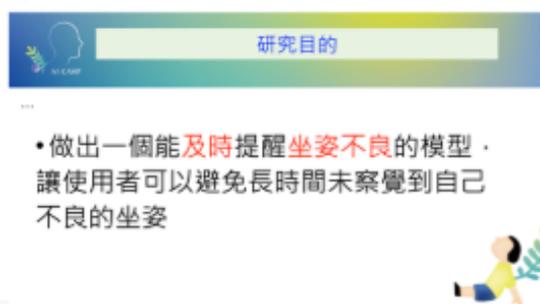
30

成果



31

成果



32

成果



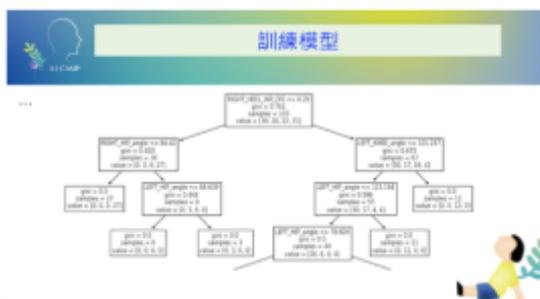
33

成果



人工智能教育高中營隊

成果



人工智能教育高中營隊



000

題目的靈感來源

- 3. 科展作品
 - [歷屆科展](#)
- 4. 各項相關競賽
 - [AI CUP 教育部全國大專校院人工智慧競賽](#)
 - [AIdca](#)
 - [TBrain](#)
 - [kaggle](#)

37

題目的靈感來源

- 5. [Papers With Code](#)
- 6. 商業套件社群
 - 樹莓派
 - Arduino
 - JetBot/Jetson
 - [Jetson Community Projects](#)
 - Google ML Kit
 - [Machine learning for mobile developers](#)

38

幫我設計一個將AI應用在環境保育的專題

7. chatGPT

以下是一種使用AI應用於環境保育的方案：

1. 資料收集：使用遙測技術（例如衛星影像）收集關於地球表面的資料，並使用機器學習算法分析資料以識別環境問題。
2. 環境問題偵測：利用資料收集的結果，識別和定位環境問題，例如森林砍伐、污染、土地開發等。
3. 偵測預測：使用預測模型預測未來的環境變化情況，並識別可能的影響。
4. 建議與行動方案：根據預測結果，提出保育環境的建議和行動方案。
5. 檢驗與評估：使用遙測數據定期評估方案的成效，並調整方案以提高效率。

這個方案可以通過自動化和數據分析，更有效地識別和解決環境問題，並提高環保效率。

39

如何利用AI來協助店員推薦顧客最喜歡的商品？

7. chatGPT

利用AI協助店員推薦顧客最喜歡的商品的方法有幾種：

1. 客戶偏好分析：通過分析顧客的歷史購買數據、瀏覽數據等，確定顧客的喜好，并向他們推薦相關的商品。
2. 圖像識別：利用圖像識別技術，讓店員掃描顧客手中的商品，並根據該商品相關的信息推薦其他相關商品。
3. 自然語言處理：利用自然語言處理技術，讓店員與顧客進行聊天，根據顧客的詢問和需求，推薦合適的商品。

這些技術都可以幫助店員更好地了解顧客的需求，並向他們提供更加精準的商品推薦，從而提高顧客滿意度和銷售額。

40

專題研究-歷屆成果

41

中華民國第 62 屆中小學科學展覽會
作品說明書

國中組 生活與應用科學(一)科

第二名

032809

以新建構之基神經網路對靜脈擴張攝影
執行多發性硬化症之判讀

學校名稱：新北市立中和國民中學

作者：

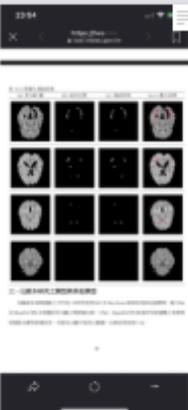
國二 呂凱天

指導老師：

陳志宏

張敏華

關鍵字：基神經網路、靜脈擴張攝影、多發性
硬化症



一張圖看出科展的現況

42

歷年專題成果

- 2019
 - 基於GAN技術進行語音轉換技術探討
- 2020
 - 智能訪「暴」- 以AI偵測影片中的不良行為
 - 利用百貨公司內商家風格識別進行室內定位研究
 - 藉醫病資料庫進行特徵提取，以人工智慧協助腎臟基因變異之致病預測
- 2021
 - 利用深度學習與影像處理技術改善黑白影片色彩化
- 2022
 - 外文漫畫之自動翻譯及製圖研究

專題研究

基於GAN技術進行語音轉換技術探討

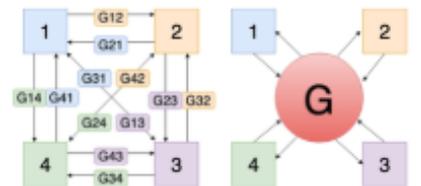
壹、研究動機

語音轉換(Voice Conversion)係指將一個人的聲音特色嵌入一句話語中，一般可應用在聲音沙啞、口腔損傷等發聲障礙者的聲音復原，抑或是非母語的發聲矯正、唱歌音調調整等。基於對相關文獻的探討，我們發現 GAN 是一個十分適合進行聲音轉換的工具。

生成對抗網路(GAN)的概念在 2014 年被提出，由於運用生成對抗網路(GAN)所生成出來的圖像有極好的成效，所以被視為深度學習的一大突破。到目前為止，生成對抗網路(GAN)大多數都用於生成圖片或是影片等等。但其實在類似的概念下，也能利用生成對抗網路(GAN)進行語音轉換，因此我們想利用 GAN 進行語音轉換，並設計實驗探討說話人數量對轉換成果之影響。

專題研究

基於GAN技術進行語音轉換技術探討



圖一：StarGAN 示意

如圖一所示，在一共有 4 個說話者的狀況下，左圖為一般的一對一模式共需 12 個生成器 (G_i)，右圖則為 StarGAN 所需的生成器，由圖一可看到一般一對一模式所需的更多生成器在 StarGAN 中僅需要一個。

專題研究

中華民國第 62 屆中小學科學展覽會
作品說明書

國中組 生活與應用科學(一)科

第三名

032809

以新建構之卷積神經網路對腦部核磁共振影像
執行多發性硬化症之判讀

學校名稱：新北市立永和國民中學

作者： 國二 呂顥天	指導老師： 鍾兆晉 陳敏華
---------------	---------------------

關鍵詞：卷積神經網路、腦部核磁共振影像、多發性
硬化症

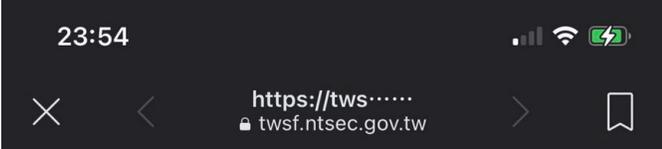
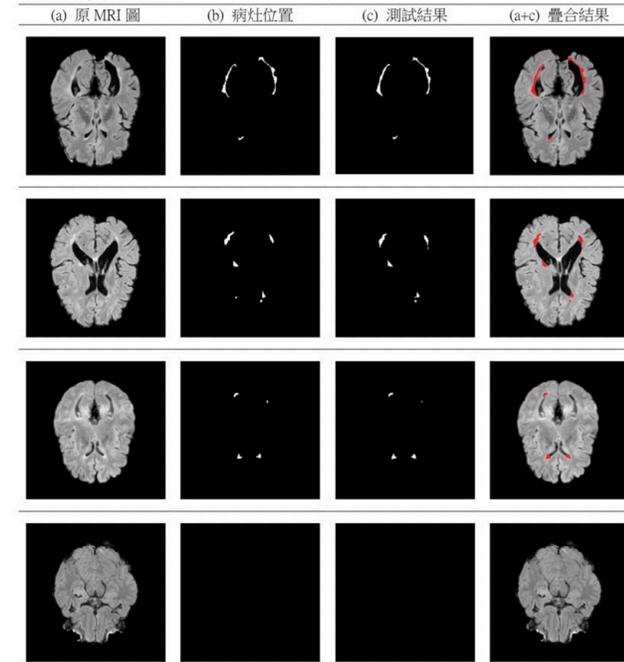
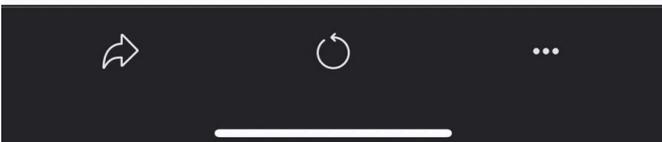


表 4-11 視覺化測試結果



三、比較本研究之模型與其他模型

為驗證本神經網路之可行性，本研究使用 IoU 及 Dice Score 兩相似度評估指標與一般 UNet 及 Mask R-CNN 及相關研究文獻之模型做比較。UNet、Mask R-CNN 與本研究新建構之卷積神經網路為實際訓練結果，其餘為文獻中提供之數據，比較結果如表 4-12。



基於GAN技術進行語音轉換技術探討

陸、討論

一、loss 函數之曲線

由上面說話者個數的 loss 函數圖可以看出，當說話者個數越多的時候，loss 函數的曲線趨於平順，在訓練的過程中，loss 比較不會有突然間的大幅變化，以這樣的結果看來，我們可以推測說話者個數有助於模型的穩定。

二、轉換之語音品質

從轉換結果可發現，當 speaker 數量愈多時，品質並沒有隨之下降，因此可見 StarGAN 的多對多語音轉換極為成功，比起原本 GAN 模型多對多時需要的眾多生成器及判別器，StarGAN 利用分類器僅需一個生成器及判別器，加快了訓練時間。

46

基於GAN技術進行語音轉換技術探討

專題研究

柒、結論

利用 StarGAN 進行聲音轉換是快速且有效的，訓練時只跑了 100 個 Epoch，便能產生逼近目標的聲音，且配合 GPU，四個 speaker 訓練時間不到一小時，為一個快速轉換多對多語言的模型，即，可以在不損失音質品質的前提下大幅提升語音轉換的目的。

47

智能防「暴」- 以AI偵測影片中的不良行為

專題研究

- 利用機器學習訓練出模型，以網路上的影片為輸入，輸出該段影片當中各個時間點含有暴力的機率。
- 分析了使用不同的程式架構對於模型預測正確率的影響 (CNN, LSTN, GRU等)，結果顯示CNN(3D)的模型有最高的正確率。
- 本研究所研究的防暴力模型除了應用於網站應片審核，也可易運用在監視器等即時監控，來達到及時發現暴力行為的效果。

48

智能防「暴」- 以AI偵測影片中的不良行為

肆、研究過程與方法

一、前置作業

- (一) 使用python自製爬蟲程式爬取Youtube上長度介於60秒到1000秒沒設有年齡限制的隨機影片5500部和設有年齡限制的相關影片5500部。
- (二) 以人工尋找是否含不良行為的影片各50部

49

智能防「暴」- 以AI偵測影片中的不良行為

專題研究

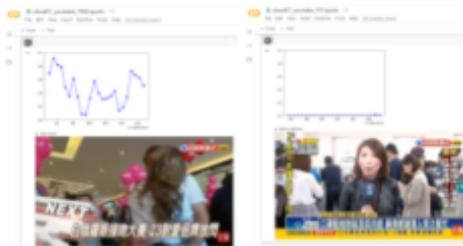
• 實驗設計

- 模型1 輸入→CNN(2D)卷積神經網路→全聯結→輸出
- 模型2 輸入→CNN(2D)卷積神經網路→全聯結→LSTM(序列)遞歸神經網路→全聯結→輸出
- 模型3 輸入→CNN(2D)卷積神經網路→全聯結→GRU(karas)(序列)遞歸神經網路→全聯結→輸出
- 模型4 輸入→CNN(2D)卷積神經網路→CNN(3D)卷積神經網路→全聯結→輸出

50

智能防「暴」- 以AI偵測影片中的不良行為

專題研究



模型3

模型4

51

利用百貨公司內商家風格識別進行室內定位研究

- 以實作室內定位方式為動機，利用圖片辨識與文字辨識模型，探討結合兩種模型在室內以環境或場景圖片的風格進行辨識與定位的成效。
- 以研究文字辨識與使用之演算法對於整體預測結果之影響。結合兩種模型，我們將研究以LineBot的形式，與日常場景進行連結。

專題研究

利用百貨公司內商家風格識別進行室內定位研究

專題研究

• 研究目的

- 以 EAST Text Detector 結合 Tesseract，製作文字辨識模型，並探討以下幾點問題：
 - 文字辨識在以手機拍攝之日常生活場景中的準確率。
 - LCS演算法的使用對文字辨識之影響。
 - 文字辨識在隨機取景下，辨識出商家之比率。
- 將「圖片辨識模型」結合「文字辨識模型」，製作成一個方便使用的 LineBot。

利用百貨公司內商家風格識別進行室內定位研究

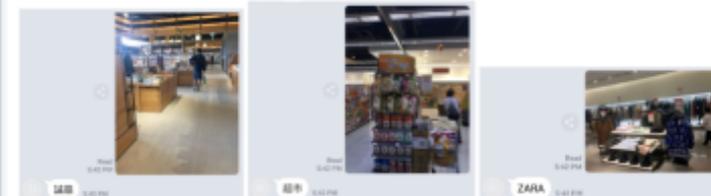


圖35、LineBot(一)

圖36、LineBot(二)

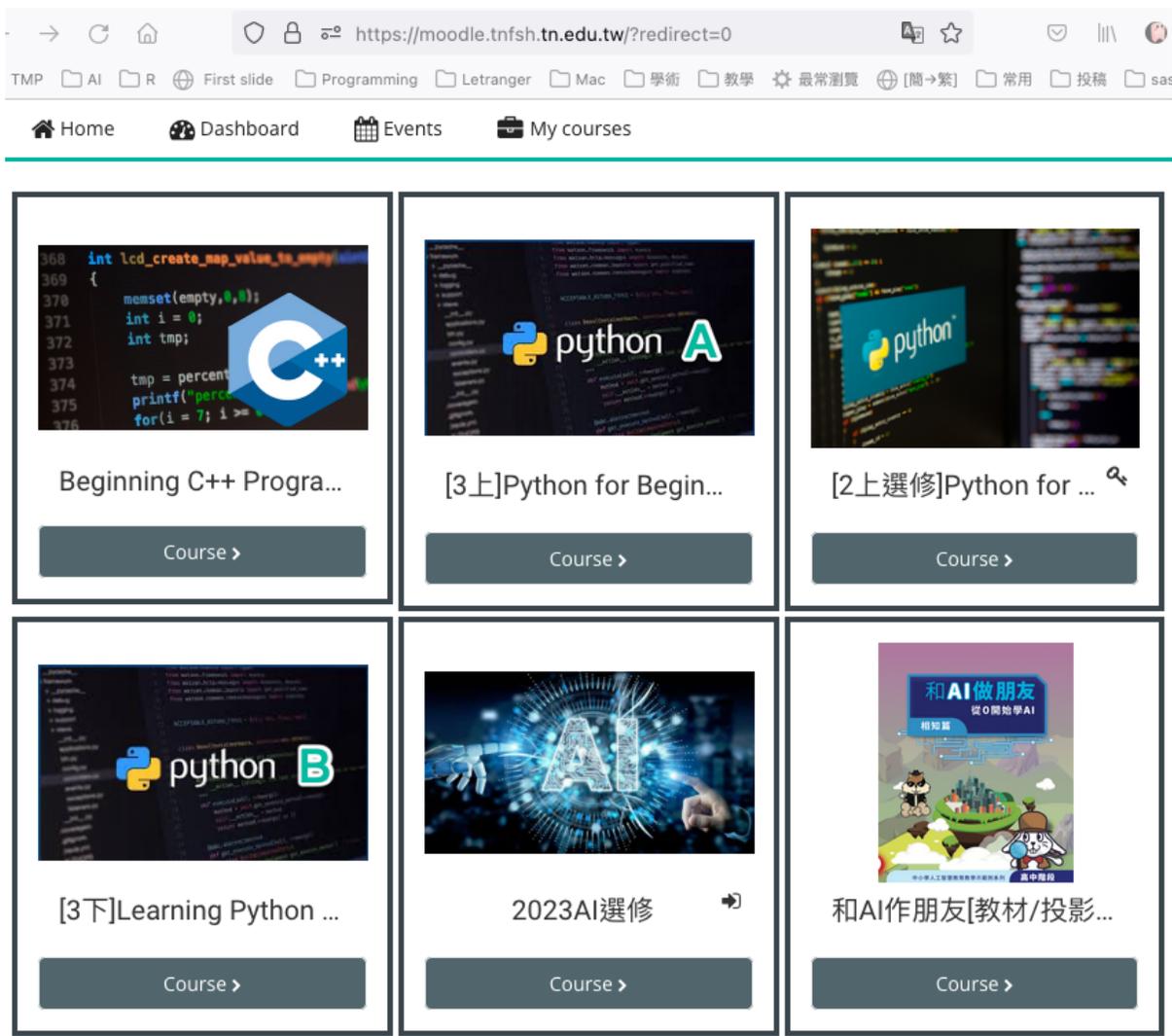
圖37、LineBot(三)

2. 監督/非監督/強化/深度學習



第一章 人工智慧簡介	03	4-3 資料探勘的相關名詞	67
1-1 思考如何避免思考	03	4-4 Apriori演算法-探勘常見項目集	68
1-2 人工智慧能聰明到什麼程度	03	4-5 關聯式法則	73
1-3 人工智慧的繁花盛開-人工智慧廣為應用於生活上的各領域	04	4-6 循序樣式探勘	75
1-4 起伏不斷的人工智慧發展歷程	06	4-7 SPAM演算法	76
1-5 人工智慧的強弱之分	09		
1-6 人工智慧與遊戲	10	第五章 監督式學習	85
1-7 人工智慧與機器學習	12	5-1 監督式學習簡介	85
		5-2 最短距離分類器	88
第二章 背景知識	15	5-3 KNN分類器	91
2-1 資料收集	15	5-4 決策樹	93
2-2 資料整理與儲存	18	5-5 隨機森林	106
2-3 特徵選擇	21		
2-4 特徵距離的計算	23	第六章 非監督式學習	110
2-5 資料的標準化	26	6-1 非監督式學習簡介	110
2-6 資料集分割	28	6-2 K-means演算法	111
2-7 樹狀結構	31	6-3 階層式分群法	117
第三章 搜尋演算法	49	第七章 增強式學習	126
3-1 搜尋是什麼?	49	7-1 增強式學習簡介	126
3-2 如何搜尋?	49	7-2 由代理人例子來了解如何進行最佳行動	127
3-3 當資料變得複雜時怎麼辦?	51	7-3 Q-學習	138
3-4 運用搜尋演算法處理複雜問題	55		
3-5 推銷員問題	60	第八章 深度學習	146
		8-1 類神經網路	146
第四章 資料探勘	64	8-2 深度學習簡介	154
4-1 什麼是「資料探勘」?	64	8-3 如何將待解問題數學化	156

教材/投影片: moodle.tnfsh.tn.edu.tw



教材/教案

- [和AI做朋友-相知篇\(教材\)](#)
- [和AI做朋友-相知篇\(教案\)](#)

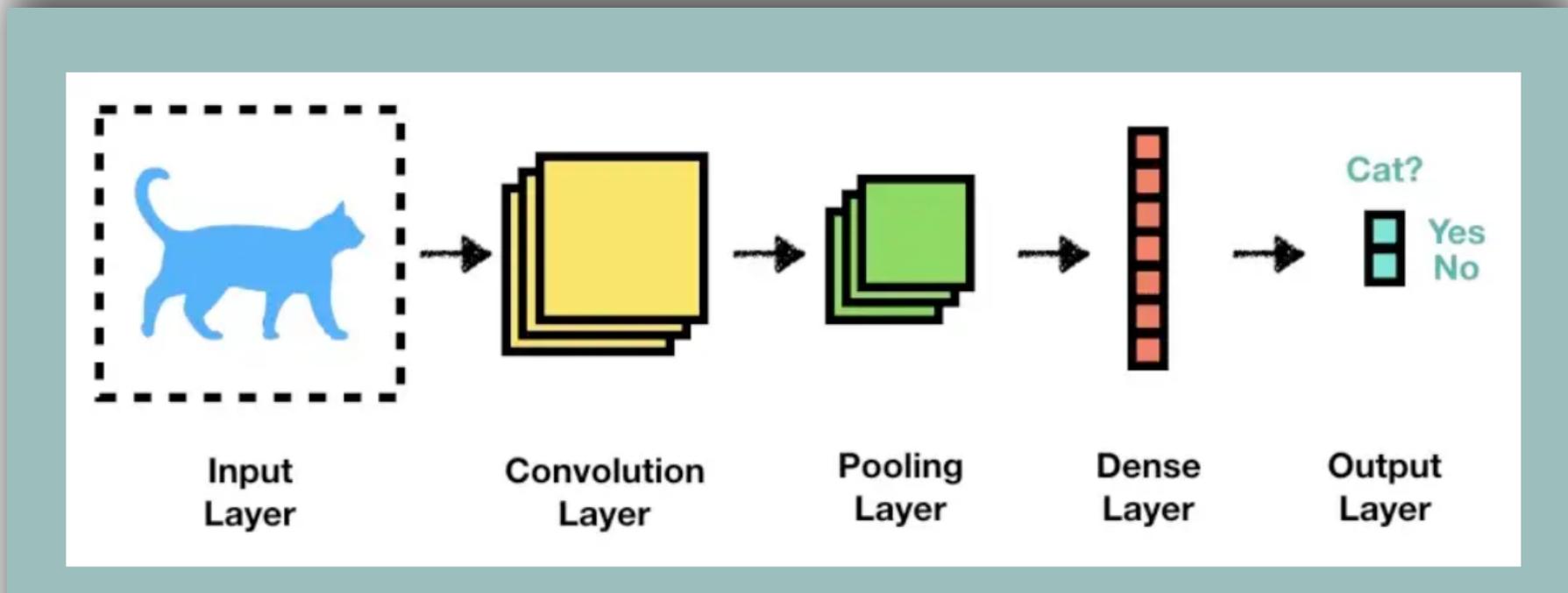
人工智慧簡介

- [EP01 人工智慧簡介-新手村大冒險](#)
- [\[任務2\]航班資訊](#)
- [分組座位表](#)
- [各國國家名稱與國旗](#)

背景知識

- [EP02 背景知識-行前準備](#)
- [Chap 02](#)
- [大數據跨界201909高中徐副主任](#)
- [特徵距離計算學習單](#)

3. CNN架構及運作原理



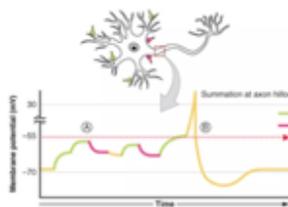
AI: 類神經網路/CNN

Yen Yung Chin

from zero to error hero

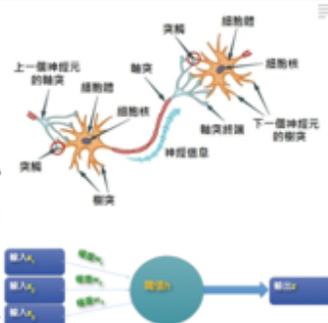
神經元

- 在接收到所有來自前方的訊息後，若該神經元引發的新動作電位超過閾值，就會再把訊息往後傳遞。
- 右圖中的電位就在B點衝過閾值，故可以將訊息再往後傳遞
- 類神經網路希望以電腦來模擬這個大腦處理訊息的架構。

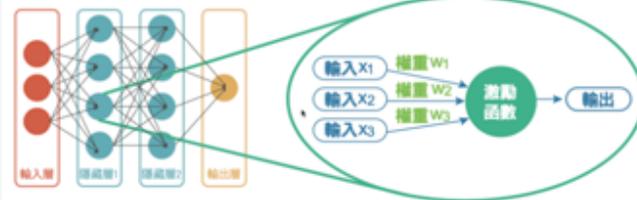


神經元

- 電腦科學家以數學模型來模擬神經網路的運作機制
- 作法：
 - 以數值來表示來自前方神經元的訊息
 - 為不同神經元傳過來的值給不同權重值、偏差值
 - 將傳入的數值與權重相乘、加總，加上偏差值
 - 將計算出來的結果交由激活函數重新計算後再傳給後方的神經元



神經元



卷積層

- 首先輸入一張圖片，一張圖片會由許多像素表示，以下是一個以「矩陣」形式呈現的概念圖：



3	1	2	5	8	4
4	5	2	6	1	3
1	3	2	1	4	6
2	5	8	5	4	2
2	4	2	1	3	5
4	5	1	1	3	2

卷積層

- 卷積層就是利用「卷積核」對原圖進行「卷積運算」，如同一個過濾器，藉此來對原始影像萃取特徵。
- 下圖表示原圖的第一區塊與一個卷積核做卷積運算後的矩陣，並將該矩陣所有的元素值加總：

$$\begin{bmatrix} 3 & 1 & 2 & 5 & 8 & 4 \\ 4 & 5 & 2 & 6 & 1 & 3 \\ 1 & 3 & 2 & 1 & 4 & 6 \\ 2 & 5 & 8 & 5 & 4 & 2 \\ 2 & 4 & 2 & 1 & 3 & 5 \\ 4 & 5 & 1 & 1 & 3 & 2 \end{bmatrix} \circ \begin{bmatrix} -2 & 3 & 1 \\ 2 & 2 & -3 \\ 1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} -6 & 3 & 2 \\ 8 & 10 & -6 \\ 1 & -3 & -2 \end{bmatrix}$$

矩陣元素值加總結果 7

卷積層

$$\begin{bmatrix} 3 & 1 & 2 & 5 & 8 & 4 \\ 4 & 5 & 2 & 6 & 1 & 3 \\ 1 & 3 & 2 & 1 & 4 & 6 \\ 2 & 5 & 8 & 5 & 4 & 2 \\ 2 & 4 & 2 & 1 & 3 & -5 \\ 4 & 5 & 1 & 1 & 3 & 2 \end{bmatrix} \circ \begin{bmatrix} -2 & 3 & 1 \\ 2 & 2 & -3 \\ 1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} -6 & 3 & 2 \\ -2 & 6 & 5 \\ 10 & 4 & -18 \\ 3 & -2 & -1 \\ -4 & 15 & 8 \\ 4 & 12 & -3 \\ 2 & -1 & -4 \\ -10 & 24 & 4 \\ 12 & 2 & -9 \\ 1 & -4 & -6 \end{bmatrix}$$

矩陣元素值加總結果 7
矩陣元素值加總結果 5
矩陣元素值加總結果 29
矩陣元素值加總結果 14

卷積層

$$\begin{bmatrix} 3 & 1 & 2 & 5 & 8 & 4 \\ 4 & 5 & 2 & 6 & 1 & 3 \\ 1 & 3 & 2 & 1 & 4 & 6 \\ 2 & 5 & 8 & 5 & 4 & 2 \\ 2 & 4 & 2 & 1 & 3 & 5 \\ 4 & 5 & 1 & 1 & 3 & 2 \end{bmatrix} \circ \begin{bmatrix} -2 & 3 & 1 \\ 2 & 2 & -3 \\ 1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} -8 & 15 & 2 \\ 2 & 6 & -6 \\ 2 & -5 & -8 \\ -10 & 6 & 6 \\ 6 & 4 & -3 \\ 5 & -8 & -5 \\ -4 & 18 & 1 \\ 4 & 2 & -12 \\ 8 & -5 & -4 \\ -12 & 3 & 3 \\ 2 & 8 & -18 \\ 5 & -4 & -2 \end{bmatrix}$$

矩陣元素值加總結果 0
矩陣元素值加總結果 1
矩陣元素值加總結果 8
矩陣元素值加總結果 -15

卷積層

$$\begin{bmatrix} 3 & 1 & 2 & 5 & 8 & 4 \\ 4 & 5 & 2 & 6 & 1 & 3 \\ 1 & 3 & 2 & 1 & 4 & 6 \\ 2 & 5 & 8 & 5 & 4 & 2 \\ 2 & 4 & 2 & 1 & 3 & 5 \\ 4 & 5 & 1 & 1 & 3 & 2 \end{bmatrix} \circ \begin{bmatrix} -2 & 3 & 1 \\ 2 & 2 & -3 \\ 1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} -2 & 9 & 2 \\ 4 & 10 & -24 \\ 2 & -4 & -2 \\ -6 & 6 & 1 \\ 10 & 16 & -15 \\ 4 & -2 & -1 \\ -4 & 3 & 4 \\ 16 & 10 & -12 \\ 2 & -1 & -3 \\ -2 & 12 & 6 \\ 10 & 8 & -6 \\ 1 & -3 & -5 \end{bmatrix}$$

矩陣元素值加總結果 -5
矩陣元素值加總結果 13
矩陣元素值加總結果 15
矩陣元素值加總結果 21

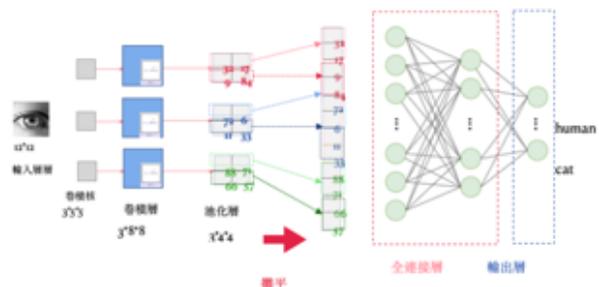
卷積層

$$\begin{bmatrix} 3 & 1 & 2 & 5 & 8 & 4 \\ 4 & 5 & 2 & 6 & 1 & 3 \\ 1 & 3 & 2 & 1 & 4 & 6 \\ 2 & 5 & 8 & 5 & 4 & 2 \\ 2 & 4 & 2 & 1 & 3 & 5 \\ 4 & 5 & 1 & 1 & 3 & 2 \end{bmatrix} \circ \begin{bmatrix} -2 & 3 & 1 \\ 2 & 2 & -3 \\ 1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} -4 & 15 & 8 \\ 4 & 8 & -6 \\ 4 & -5 & -1 \\ -10 & 24 & 5 \\ 8 & 4 & -3 \\ 5 & -1 & -1 \\ -16 & 15 & 4 \\ 4 & 2 & -9 \\ 1 & -1 & -3 \\ -10 & 12 & 2 \\ 2 & 6 & -15 \\ 1 & -3 & -2 \end{bmatrix}$$

矩陣元素值加總結果 23
矩陣元素值加總結果 31
矩陣元素值加總結果 -3
矩陣元素值加總結果 -7

2. CNN 架構與運作原理

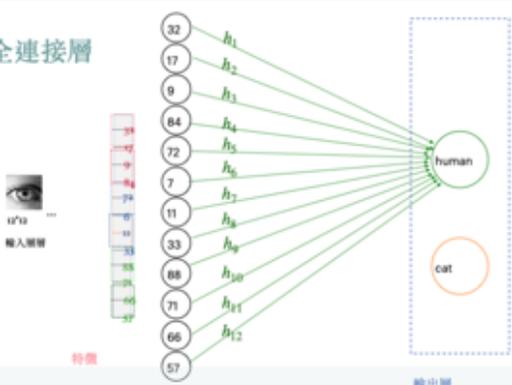
全連接層



67

2. CNN 架構與運作原理

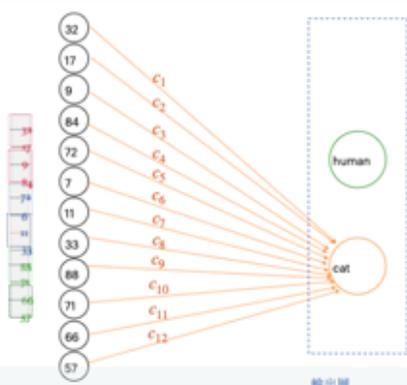
全連接層



68

2. CNN 架構與運作原理

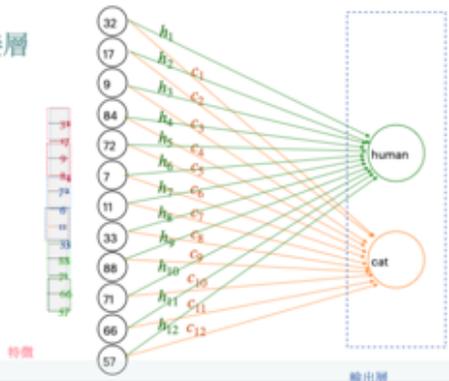
全連接層



69

2. CNN 架構與運作原理

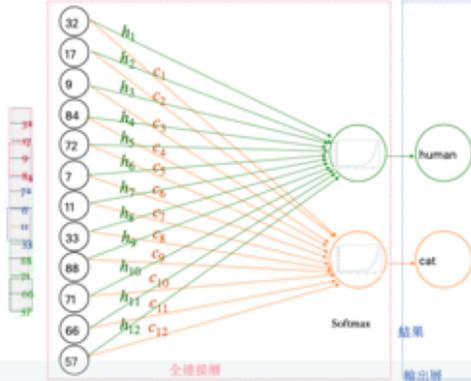
全連接層



70

2. CNN 架構與運作原理

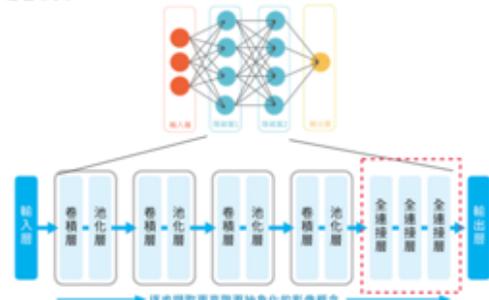
全連接層



71

2. CNN 架構與運作原理

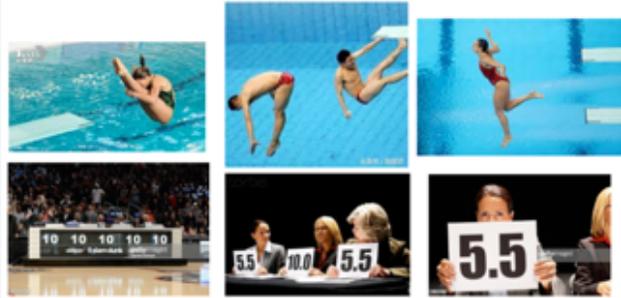
全連接層



72

2. CNN 架構與運作原理

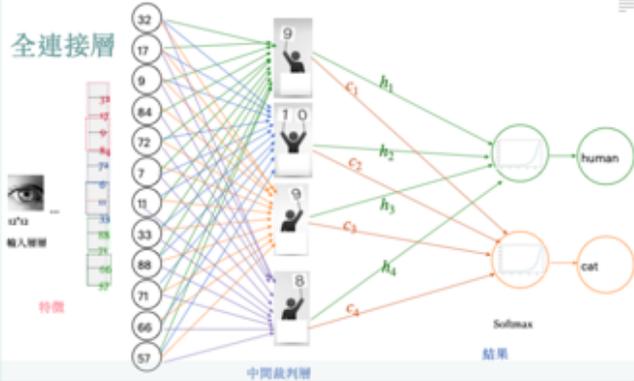
全連接層



73

2. CNN 架構與運作原理

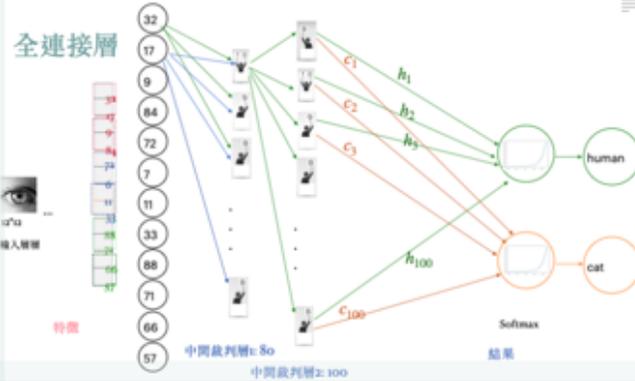
全連接層



74

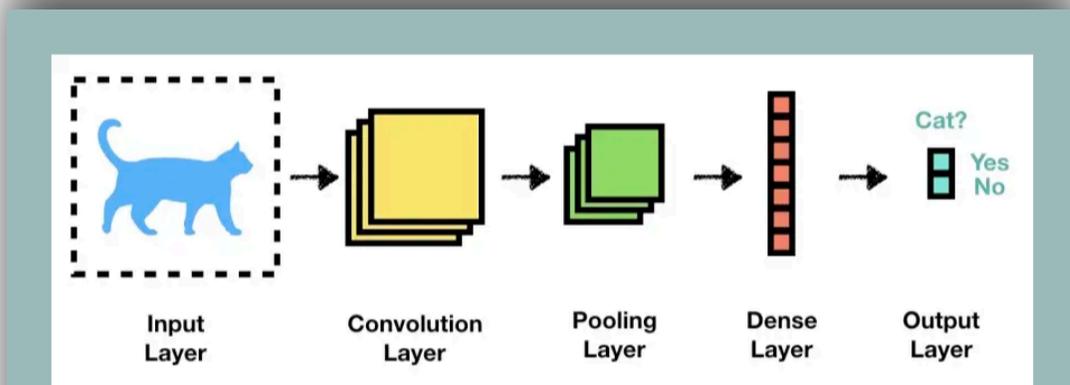
2. CNN 架構與運作原理

全連接層



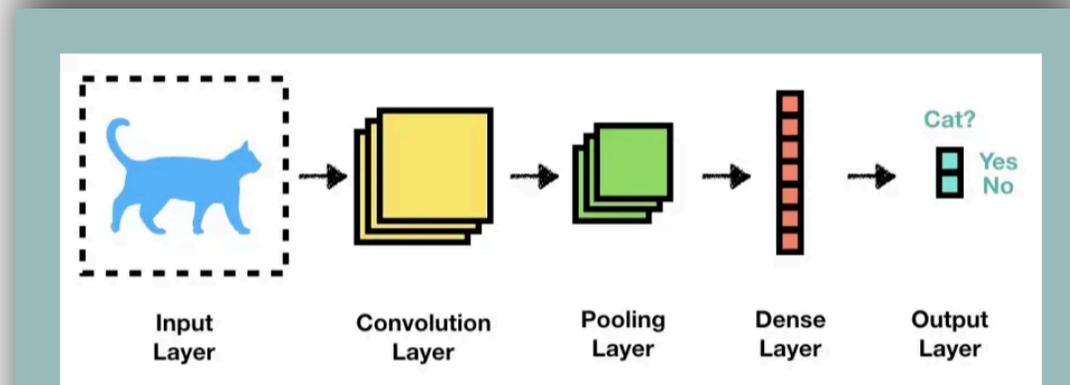
75

3. CNN實作



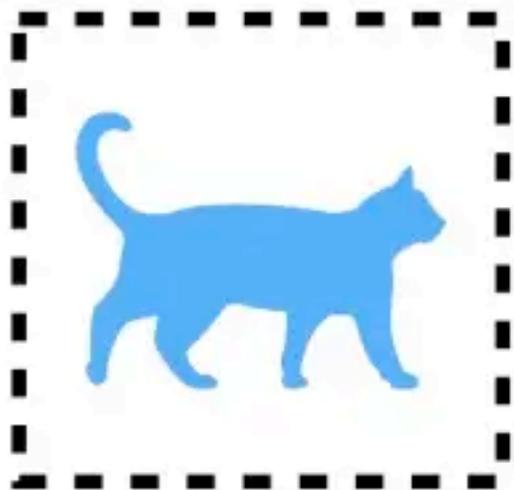
CNN模型實作1 | 顏永進 / 1229

from zero to error hero

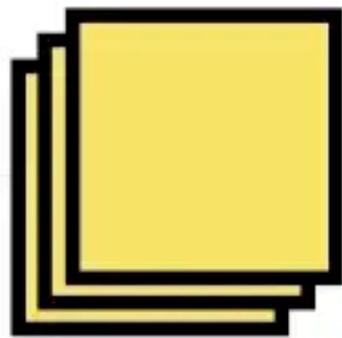


CNN模型實作2 | 顏永進 / 1229

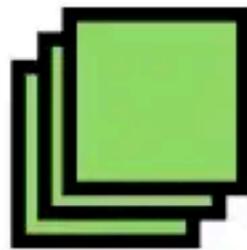
from zero to error hero



Input Layer



Convolution Layer



Pooling Layer



Dense Layer



Output Layer

CNN模型實作1

from zero to error hero

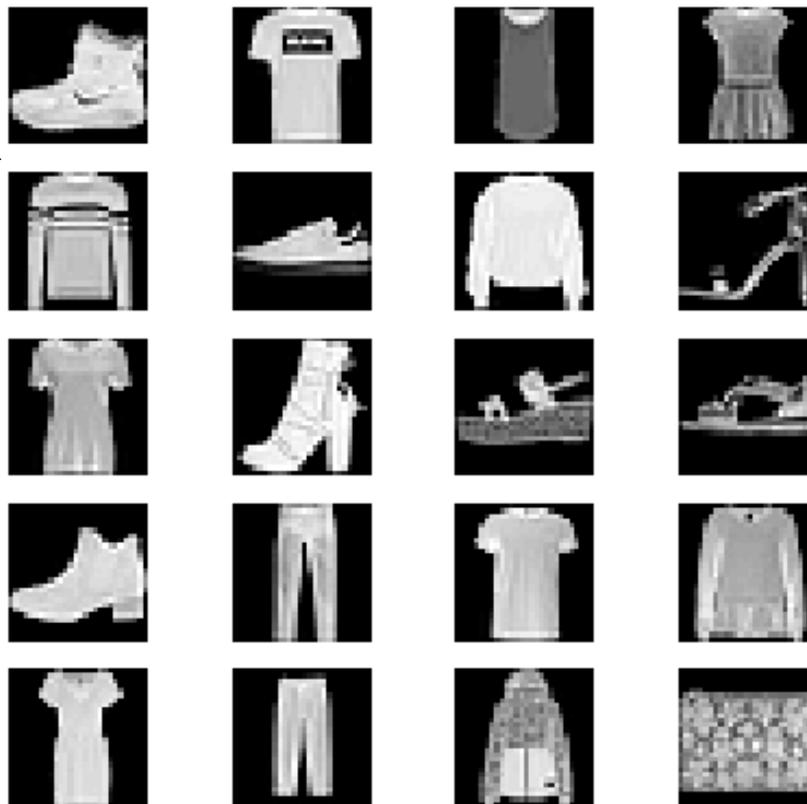
顏永進 / 1229

CONTENT

1. 任務說明/資料集簡介
2. 下載資料集/分割資料集
3. 觀察資料集內容
4. 將資料標準化/歸一化
5. 建立模型
6. 編譯模型
7. 訓練模型
8. 測試模型效能

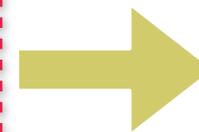
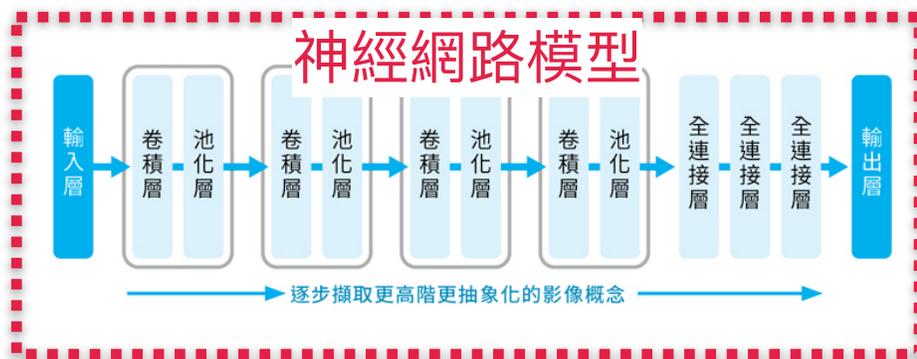
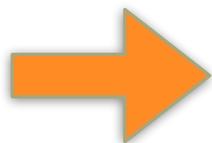
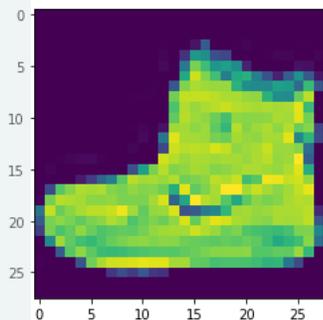
1. 資料集

- Fashion-MNIST
- 7萬張10種類別的商品的正面圖片
- 6萬張訓練集
- 1萬張測試集
- 28x28灰階圖片



分類	描述
0	T-shirt/top (T恤)
1	Trouser (褲子)
2	Pullover (套頭毛衣)
3	Dress (裙子)
4	Coat (外套)
5	Sandal (涼鞋)
6	Shirt (襯衫)
7	Sneaker (運動鞋)
8	Bag (包包)
9	Ankle boot (短靴)

1. 任務說明



Ankle boot / 短靴



2. 下載資料集/分割資料集

```
1 import tensorflow as tf
2
3 # 從網路下載資料集
4 (x_train, y_train), (x_test, y_test) = tf.keras.datasets.fashion_mnist.load_data()
5 print(type(x_train))
6 print("訓練集圖片矩陣形狀(維度):", x_train.shape)
7 print("訓練集標籤矩陣形狀(維度):", y_train.shape)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
29515/29515 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880 [=====] - 1s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 [=====] - 0s 0us/step
<class 'numpy.ndarray'>
訓練集圖片矩陣形狀(維度): (60000, 28, 28)
訓練集標籤矩陣形狀(維度): (60000,)
```

3. 觀察資料集內容



```
print("第一張圖的矩陣形狀:", x_train[0].shape)  
print("第一張圖的矩陣內容:", x_train[0])
```

第一張圖的矩陣形狀: (28, 28)

第一張圖的矩陣內容:

```
0 0 0 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 13 73 0 0 1 4 0 0 0 0 1 1 0  
0 0 0 0 0 0 0 0 0 0 0 0 3 0 36 136 127 62 54 0 0 0 1 3 4 0 0 3  
0 0 0 0 0 0 0 0 0 0 0 0 6 0 102 204 176 134 144 123 23 0 0 0 0 12 10 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 155 236 207 178 107 156 161 109 64 23 77 130 72 15  
0 0 0 0 0 0 0 0 0 0 0 1 0 69 207 223 218 216 216 163 127 121 122 146 141 88 172 66  
0 0 0 0 0 0 0 0 0 1 1 1 0 200 232 232 233 229 223 223 215 213 164 127 123 196 229 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 183 225 216 223 228 235 227 224 222 224 221 223 245 173 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 193 228 218 213 198 180 212 210 211 213 223 220 243 202 0  
0 0 0 0 0 0 0 0 0 1 3 0 12 219 220 212 218 192 169 227 208 218 224 212 226 197 209 52  
0 0 0 0 0 0 0 0 0 0 6 0 99 244 222 220 218 203 198 221 215 213 222 220 245 119 167 56  
0 0 0 0 0 0 0 0 0 4 0 0 55 236 228 230 228 240 232 213 218 223 234 217 217 209 92 0  
0 0 1 4 6 7 2 0 0 0 0 0 237 226 217 223 222 219 222 221 216 223 229 215 218 255 77 0  
0 3 0 0 0 0 0 0 0 62 145 204 228 207 213 221 218 208 211 218 224 223 219 215 224 244 159 0  
0 0 0 0 18 44 82 107 189 228 220 222 217 226 200 205 211 230 224 234 176 188 250 248 233 238 215 0  
0 57 187 208 224 221 224 208 204 214 208 209 200 159 245 193 206 223 255 255 221 234 221 211 220 232 246 0  
3 202 228 224 221 211 211 214 205 205 205 220 240 80 150 255 229 221 188 154 191 210 204 209 222 228 225 0  
98 233 198 210 222 229 229 234 249 220 194 215 217 241 65 73 106 117 168 219 221 215 217 223 223 224 229 29  
75 204 212 204 193 205 211 225 216 185 197 206 198 213 240 195 227 245 239 223 218 212 209 222 220 221 230 67  
48 203 183 194 213 197 185 190 194 192 202 214 219 221 220 236 225 216 199 206 186 181 177 172 181 205 206 115  
0 122 219 193 179 171 183 196 204 210 213 207 211 210 200 196 194 191 195 191 198 192 176 156 167 177 210 92  
0 0 74 189 212 191 175 172 175 181 185 188 189 188 193 198 204 209 210 210 211 188 188 194 192 216 170 0  
2 0 0 0 66 200 222 237 239 242 246 243 244 221 220 193 191 179 182 182 181 176 166 168 99 58 0 0  
0 0 0 0 0 0 0 40 61 44 72 41 35 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

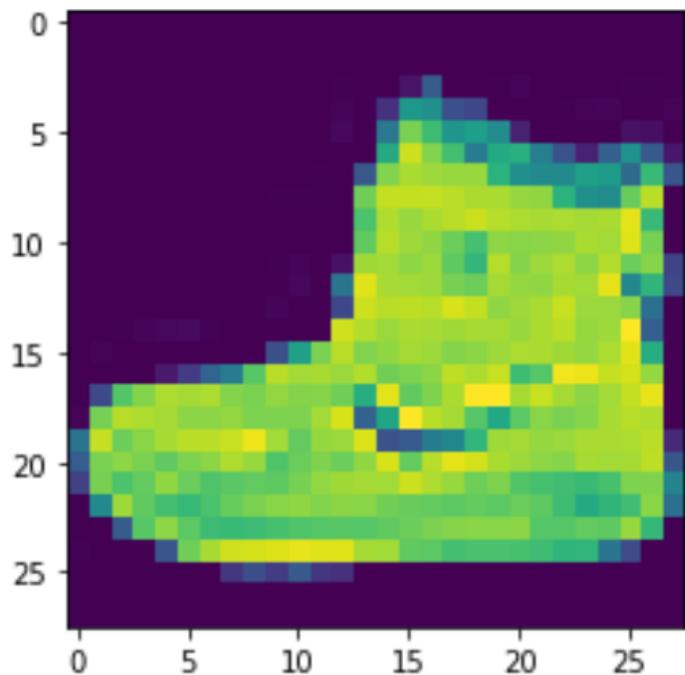
3. 觀察資料集內容

```
▶ import matplotlib.pyplot as plt

print('訓練集中第1張圖的實際分類為:', y_train[0])
plt.imshow(x_train[0])
```

訓練集中第1張圖的實際分類為： 9

<matplotlib.image.AxesImage at 0x7f6ddc8ca8b0>

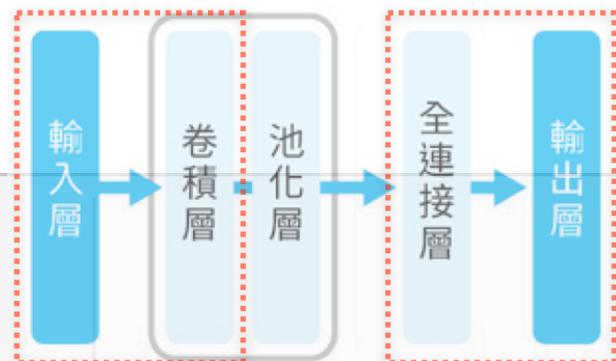


分類	描述
0	T-shirt/top (T恤)
1	Trouser (褲子)
2	Pullover (套頭毛衣)
3	Dress (裙子)
4	Coat (外套)
5	Sandal (涼鞋)
6	Shirt (襯衫)
7	Sneaker (運動鞋)
8	Bag (包包)
9	Ankle boot (短靴)

5. 建立模型

一個非常基本的卷積神經網路模型:

- 第一層為卷積層，有32個神經元，使用2X2的卷積核，啟動函數則是常見的relu，這層同時也負責接收資料，在這裡需將資料的形狀明確定義清楚。
- 第二層為池化層，使用2X2的池化核
- 將資料送至輸出層之前先對資料進行攤平處理
- 最後一層為輸出層，也是全連接層(Dense)，可以看出這層有10個神經元，用來儲存資料分為是哪一個類別的機率值



```
1 #建構模型
2 model = tf.keras.Sequential()
3 #輸入層/卷積層
4 model.add(tf.keras.layers.Conv2D(filters=32, kernel_size=2, activation='relu', input_shape=(28,28,1)))
5 #池化層
6 model.add(tf.keras.layers.MaxPooling2D(pool_size=2))
7 #攤平
8 model.add(tf.keras.layers.Flatten())
9 #全連接層/輸出層
10 model.add(tf.keras.layers.Dense(10, activation='softmax'))
```

在CoLab上執行這段程式碼並不會有輸出，我們可以藉由summary函式來查看模型的基本架構

```
▶ model.summary()
```

```
↳ Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 27, 27, 32)	160
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 32)	0
flatten_1 (Flatten)	(None, 5408)	0
dense_1 (Dense)	(None, 10)	54090

```
=====  
Total params: 54,250  
Trainable params: 54,250  
Non-trainable params: 0  
=====
```

5. 建立模型

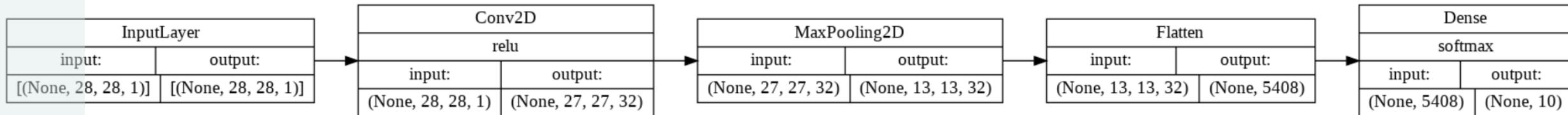
5. 建立模型

TensorFlow提供的模型架構圖

```
[ ] tf.keras.utils.plot_model(  
    model,  
    to_file='model-1.png',  
    show_shapes=True,  
    show_dtype=None,  
    show_layer_names=None,  
    rankdir='LR',  
    expand_nested=True,  
    dpi=96,  
    layer_range=None,  
    show_layer_activations=True  
)
```

▶ #建構模型

```
model = tf.keras.Sequential()  
#輸入層/卷積層  
model.add(tf.keras.layers.Conv2D(filters=32, kernel_size=2,  
                                  activation='relu', input_shape=(28,28,1)))  
#池化層  
model.add(tf.keras.layers.MaxPooling2D(pool_size=2))  
#攤平  
model.add(tf.keras.layers.Flatten())  
#全連接層/輸出層  
model.add(tf.keras.layers.Dense(10, activation='softmax'))
```



6. 編譯模型

Keras用compile函式來指定這三個方法：

- **loss**: 即損失函數，比較模型的預測結果與真實答案的差異，這個數值越小越好。此次任務為分類預測，而且資料的標籤尚未進行one-hot-encoding轉換(留在實作2)，所以選擇sparse_categorical_crossentropy做為loss function。
- **optimizer**: 即優化器，在深度學習中負責找出所有神經元中各參數的最佳值的方法，也就是能讓模型預測結果最準確的方法。和損失函數一樣，可用的optimizer方法也有很多，這裡我們直接選擇最常用的adam。
- **metrics**: 即效能指標，這是評估模型整體效能的標準，這個數值越高，表示模型的預測效能越好。我們使用最常用的accuracy，也就是模型正確預測的機率。

```
[ ] model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

7. 訓練模型

Keras使用fit()函式來進行訓練，訓練時要宣告以下兩個參數值：

- epochs: 所有的資料全都被用來訓練模型一次稱為1個epoch，如果一個資料集很小（100張），那麼把這100張圖全部丟進模型訓練就完成了一個epoch的訓練。
- batch_size: 由於訓練的資料集一般數量都極為龐大，不太可能一口氣全部丟進模型中去訓練，因為電腦的記憶體畢竟有限，batch_size就是用來指定每次要丟入模型中去訓練的資料數量。每一次以一個batch的資料進行訓練稱為一次迭代(iteration)，模型每完成一次迭代的訓練，就會修正下一次訓練參數的方向。

```
[ ] model.fit(x_train, y_train, batch_size=100, epochs=5)
```

```
Epoch 1/5  
600/600 [=====] - 2s 3ms/step - loss: 0.5359 - accuracy: 0.8157  
Epoch 2/5  
600/600 [=====] - 2s 3ms/step - loss: 0.3724 - accuracy: 0.8719  
Epoch 3/5  
600/600 [=====] - 2s 3ms/step - loss: 0.3351 - accuracy: 0.8840  
Epoch 4/5  
600/600 [=====] - 2s 3ms/step - loss: 0.3108 - accuracy: 0.8907  
Epoch 5/5  
600/600 [=====] - 2s 3ms/step - loss: 0.2953 - accuracy: 0.8965  
<keras.callbacks.History at 0x7fdf0e233160>
```

8. 測試模型效能

使用keras的predict()函式來進行預測的第一張。predict()預測的對像是整個測試集的資料

```
▶ pred = model.predict(x_test)[0]  
print(pred)
```

```
313/313 [=====] - 2s 6ms/step  
[6.6730004e-06 9.0090134e-07 6.3061011e-06 2.8054474e-06 2.7422097e-05  
1.1541011e-02 2.6281763e-05 3.8374770e-02 2.4585463e-03 9.4755542e-01]
```

- 結果是10個小數，為模型判斷這張圖為10個分類的可能性(以科學記號表示)，總和為1。
- 此為softmax函數的輸出樣貌，看的出來最後一個類別的數值最大(0.99)。
- 答案y_test[0]輸出，結果發現為9，表示這次的預測是正確的。

```
[22] print(y_test[0])
```

```
9
```

8. 測試模型效能

`evaluate()`會傳回`loss`與`accuracy`兩個值:

- `loss`是在`compile`模型時設定的損失函數計算得來的損失值，`loss`越低，模型越好。它不是一個百分比，而是一次`epoch`中訓練集樣本或驗證集樣本所有`error`的和。通過損失函數的計算，從而更新我們的模型參數，目標是為了減小最佳化誤差(Optimization error)。
- `accuracy`則是模型在資料集上的預測精準率，目標是對模型的效果進行評量。`accuracy`是當模型的參數學習得到並固定後來計算得到的。通過比較真實標籤和預測標籤來確定的0-1損失。它是一個百分比值。

```
▶ score = model.evaluate(x_test, y_test)
print('loss:', score[0])
print('accuracy:', score[1])
```

```
313/313 [=====] - 3s 8ms/step - loss: 0.3172 - accuracy: 0.8852
loss: 0.3171987235546112
accuracy: 0.885200023651123
```

這個模型夠好嗎？

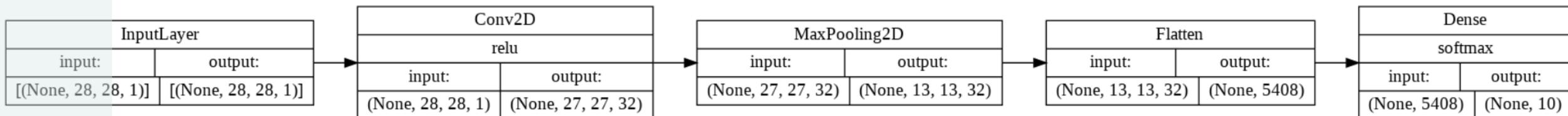
8. 測試模型效能

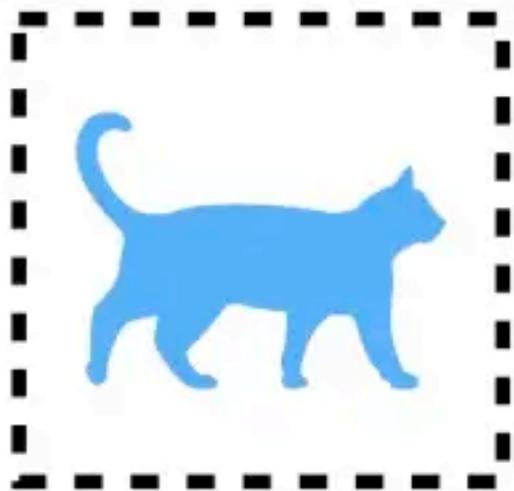
```
loss: 0.3171987235546112  
accuracy: 0.885200023651123
```



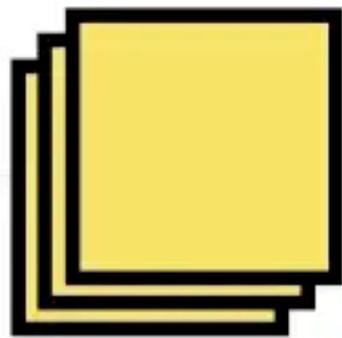
#建構模型

```
model = tf.keras.Sequential()  
#輸入層/卷積層  
model.add(tf.keras.layers.Conv2D(filters=32, kernel_size=2,  
                                  activation='relu', input_shape=(28,28,1)))  
  
#池化層  
model.add(tf.keras.layers.MaxPooling2D(pool_size=2))  
  
#攤平  
model.add(tf.keras.layers.Flatten())  
  
#全連接層/輸出層  
model.add(tf.keras.layers.Dense(10, activation='softmax'))
```

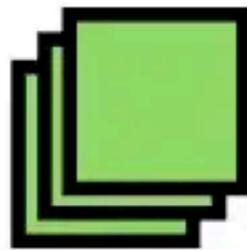




Input Layer



Convolution Layer



Pooling Layer



Dense Layer



Output Layer

CNN模型實作2

from zero to error hero

顏永進 / 1229

CONTENT

1. 進一步查看效能
2. 轉換label資料格式
3. 修改模型架構
4. 編譯、訓練新模型
5. 再次評估模型效能
6. 模型的建置流程

1. 任務說明

實作1:

```
[ ] model.fit(x_train, y_train, batch_size=100, epochs=5)

Epoch 1/5
600/600 [=====] - 2s 3ms/step - loss: 0.5359 - accuracy: 0.8157
Epoch 2/5
600/600 [=====] - 2s 3ms/step - loss: 0.3724 - accuracy: 0.8719
Epoch 3/5
600/600 [=====] - 2s 3ms/step - loss: 0.3351 - accuracy: 0.8840
Epoch 4/5
600/600 [=====] - 2s 3ms/step - loss: 0.3108 - accuracy: 0.8907
Epoch 5/5
600/600 [=====] - 2s 3ms/step - loss: 0.2953 - accuracy: 0.8965
<keras.callbacks.History at 0x7fdf0e233160>
```

實作2:

```
1 history = model.fit(x_train, y_train, batch_size=100, epochs=50, validation_data=(x_test, y_test))
```

1. 進一步查看效能

```
1 history = model.fit(x_train, y_train, batch_size=100, epochs=50, validation_data=(x_test, y_test))
```

類神經網路實作2.ipynb ☆

檔案 編輯 檢視畫面 插入 執行階段 工具 說明 已儲存所有變更

+ 程式碼 + 文字

此次評估結果，本模型的損失值(loss)約為0.35、精準率(accuracy)約為0.88。

-----實作2-----

8. 進一步查看效能

訓練模型只是整個模型建構的一部份，後續的模型調校以提高模型效能才是我們的目的。為了更清楚模型的效能，我們將epoch的次數增加到50次，並利用history()函式將每個epoch的效能記錄下來。

```
[24] 1 history = model.fit(x_train, y_train, batch_size=100, epochs=50, validation_data=(x_test, y_test))
```

```
Epoch 1/50  
600/600 [=====] - 25s 41ms/step - loss: 0.2763 - accuracy: 0.9019 - val_loss: 0.3092 - val_accu  
Epoch 2/50  
600/600 [=====] - 25s 41ms/step - loss: 0.2669 - accuracy: 0.9044 - val_loss: 0.3069 - val_accu  
Epoch 3/50  
600/600 [=====] - 25s 41ms/step - loss: 0.2581 - accuracy: 0.9082 - val_loss: 0.3051 - val_accu  
Epoch 4/50  
600/600 [=====] - 25s 42ms/step - loss: 0.2498 - accuracy: 0.9113 - val_loss: 0.3055 - val_accu  
Epoch 5/50  
600/600 [=====] - 26s 43ms/step - loss: 0.2431 - accuracy: 0.9129 - val_loss: 0.2982 - val accu
```

al_accuracy: 0.8885

al_accuracy: 0.8909

al_accuracy: 0.8868

al_accuracy: 0.8881

al_accuracy: 0.8997

al_accuracy: 0.8992

al_accuracy: 0.8952

al_accuracy: 0.8974

al_accuracy: 0.8967

1. 進一步查看效能

- 將每次的損失值(loss)由歷史記錄中取出

```
▶ 1 print(len(history.history['loss']))  
2 print(len(history.history['val_loss']))
```

```
↳ 50  
50
```

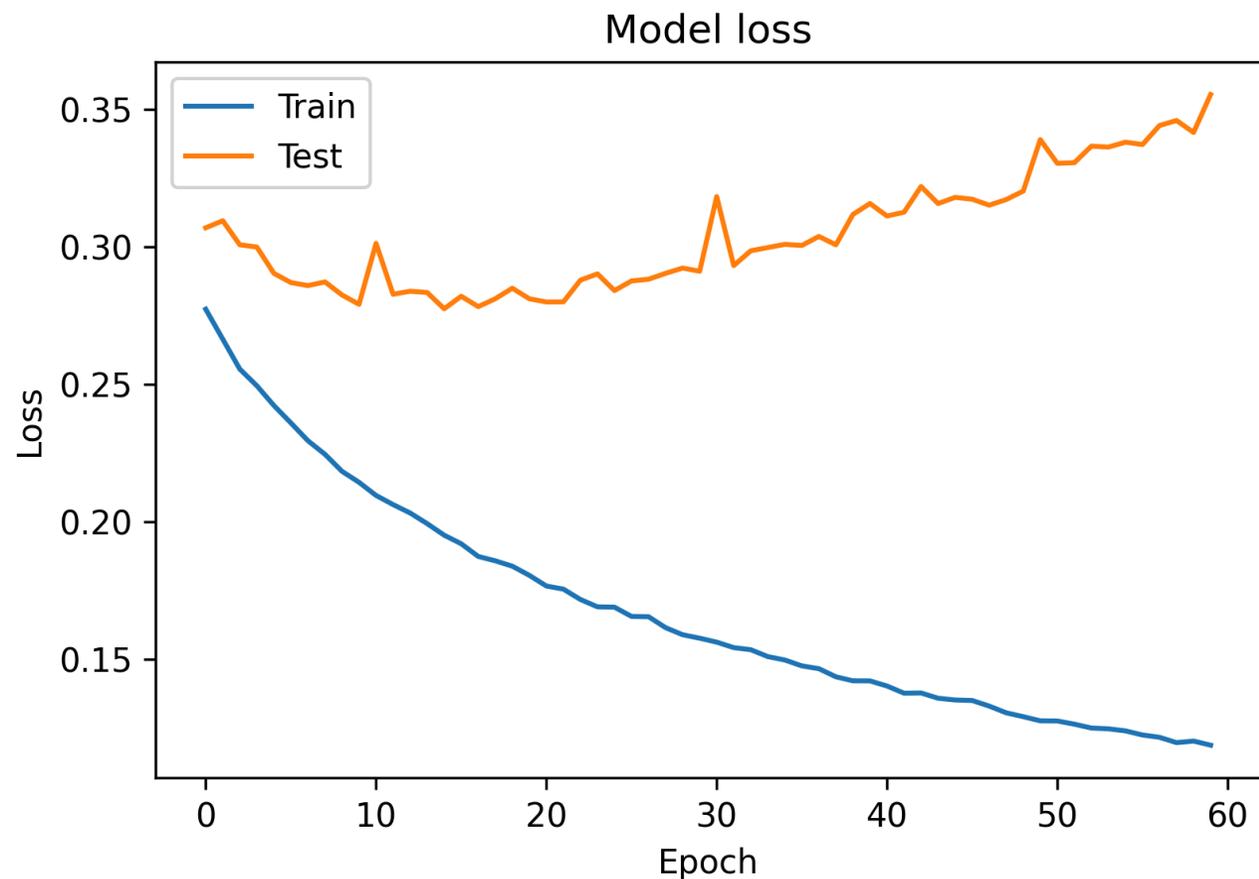
```
1 print('train loss:', history.history['loss'])  
2 print('test loss:', history.history['val_loss'])
```

```
train loss: [0.2763459086418152, 0.2668665647506714, 0.258055716753006, 0.24981632828712463, 0.24306733906269073, 0.23676112294  
test loss: [0.3091515004634857, 0.3068563640117645, 0.30511489510536194, 0.3054684102535248, 0.2982345223426819, 0.287497490644
```

1. 進一步查看效能

- 將每次的損失值(loss)由歷史記錄中取出，畫成折線圖

```
1 #取出每次的訓練資料損失值
2 plt.plot(history.history['loss'])
3 #取出每次的測試資料損失值
4 plt.plot(history.history['val_loss'])
5 #加上標題
6 plt.title('Model loss')
7 #加上y軸標籤
8 plt.ylabel('Loss')
9 #加上x軸標籤
10 plt.xlabel('Epoch')
11 #顯示折線的名稱
12 plt.legend(['Train', 'Test'])
13 #顯示折線圖
14 plt.show()
```



2. 轉換LABEL資料格式

- 在資料集中以數字0~9來表示10種不同的分類，這種表示法隱含了一些問題。
- 以數字來表示分類，這會讓模型誤會這些分類是連續、有序的，也就是說，第2類比第1類大，第3類到第5類的距離等於第5類到第7類的距離。
- 解決這個問題的方式是使用one-hot encoding(獨熱編碼)，方法是使用N個位元來表示N個類別。此資料集為例，原本的的類別和轉換過後的分別如下：

- 0: 0000000001

- 1: 0000000010

- 2: 0000000100

- ...

- 9: 1000000000

2. 轉換LABEL資料格式

- Keras提供的to_categorical()函式可以讓我們簡單快速的對資料集中的分類進行one-hot encoding轉換，從下例中也可以看到原始資料與轉換後資料的比對。

```
1 from tensorflow.keras.utils import to_categorical
2 print('第一張圖的類別(轉換前):', y_train[0])
3 # 將label進行one-hot encoding轉換
4 y_train = to_categorical(y_train)
5 y_test = to_categorical(y_test)
6 print('第一張圖的類別(轉換後):', y_train[0])
7 # 後面要將model.compile的loss改為'categorical_crossentropy'
```

第一張圖的類別(轉換前): 9

第一張圖的類別(轉換後): [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]

3. 修改模型架構

```
1 model = tf.keras.Sequential()
2 # 輸入層/第1個卷積層
3 model.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, padding='same', activation='relu', input_shape=(28,28,1)))
4 # 第1個池化層
5 model.add(tf.keras.layers.MaxPooling2D(pool_size=2))
6 # 第1個Dropout層
7 model.add(tf.keras.layers.Dropout(0.5))
8 # 第2個卷積層
9 model.add(tf.keras.layers.Conv2D(filters=64, kernel_size=5, padding='same', activation='relu'))
10 # 第2個池化層
11 model.add(tf.keras.layers.MaxPooling2D(pool_size=2))
12 # 第2個Dropout層
13 model.add(tf.keras.layers.Dropout(0.5))
14 # 攤平
15 model.add(tf.keras.layers.Flatten())
16 # 第一個全連接層
17 model.add(tf.keras.layers.Dense(128, activation='relu'))
18 # 第3個Dropout層
19 model.add(tf.keras.layers.Dropout(0.5))
20 # 第2個全連接層/輸出層
21 model.add(tf.keras.layers.Dense(10, activation='softmax'))
```

3. 修改模型架構

- 查看新模型架構

1 # 查看模型結構

2 model.summary()

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 28, 28, 32)	320
max_pooling2d_3 (MaxPooling 2D)	(None, 14, 14, 32)	0
dropout_3 (Dropout)	(None, 14, 14, 32)	0
conv2d_4 (Conv2D)	(None, 14, 14, 64)	51264
max_pooling2d_4 (MaxPooling 2D)	(None, 7, 7, 64)	0
dropout_4 (Dropout)	(None, 7, 7, 64)	0
flatten_2 (Flatten)	(None, 3136)	0
dense_3 (Dense)	(None, 128)	401536
dropout_5 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 10)	1290

Total params: 454,410
Trainable params: 454,410
Non-trainable params: 0

- 實作1模型架構

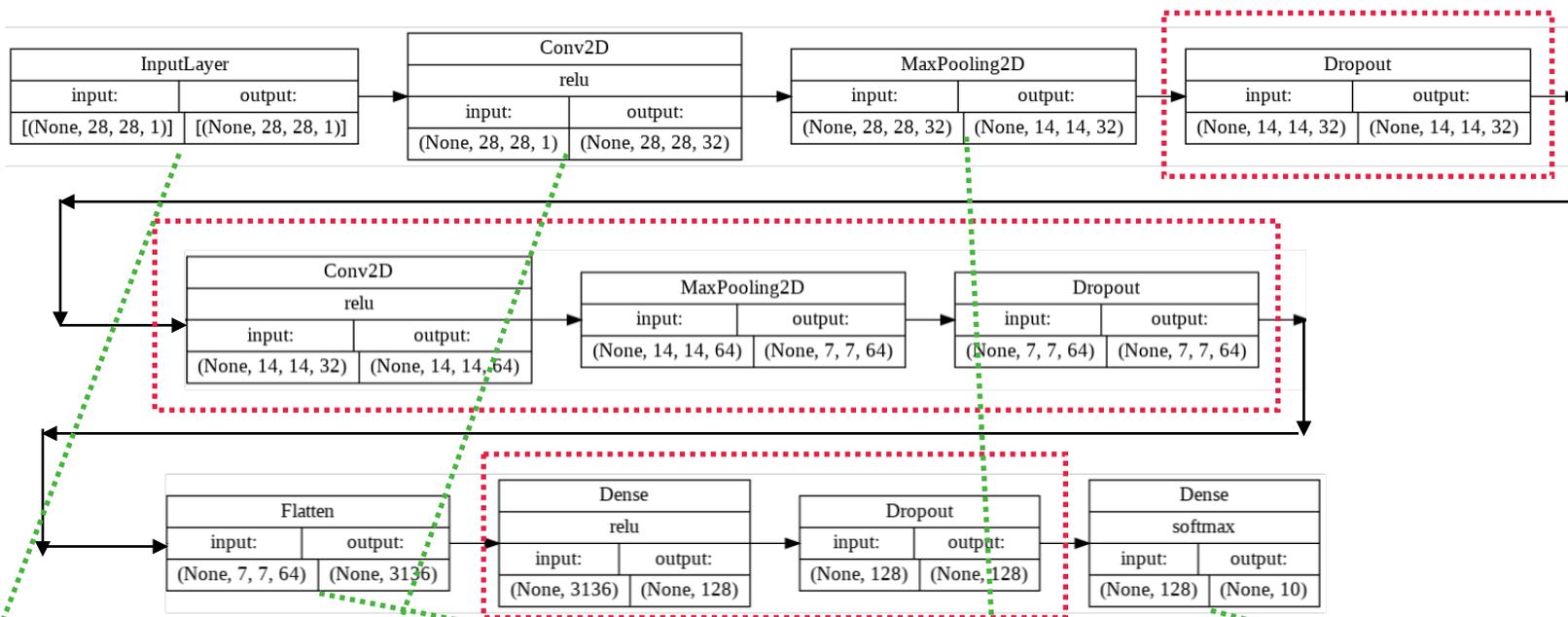
Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 27, 27, 32)	160
max_pooling2d_1 (MaxPooling 2D)	(None, 13, 13, 32)	0
flatten_1 (Flatten)	(None, 5408)	0
dense_1 (Dense)	(None, 10)	54090

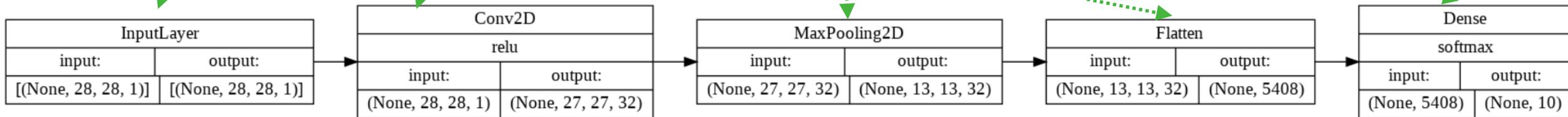
Total params: 54,250
Trainable params: 54,250
Non-trainable params: 0

3. 修改模型架構

- 以tf.keras.utils.plot_model()函式查看實作2的圖形化模型架構



- 實作1模型架構



4. 編譯、訓練新模型

- 對這個修改過後的模型進行編譯及訓練

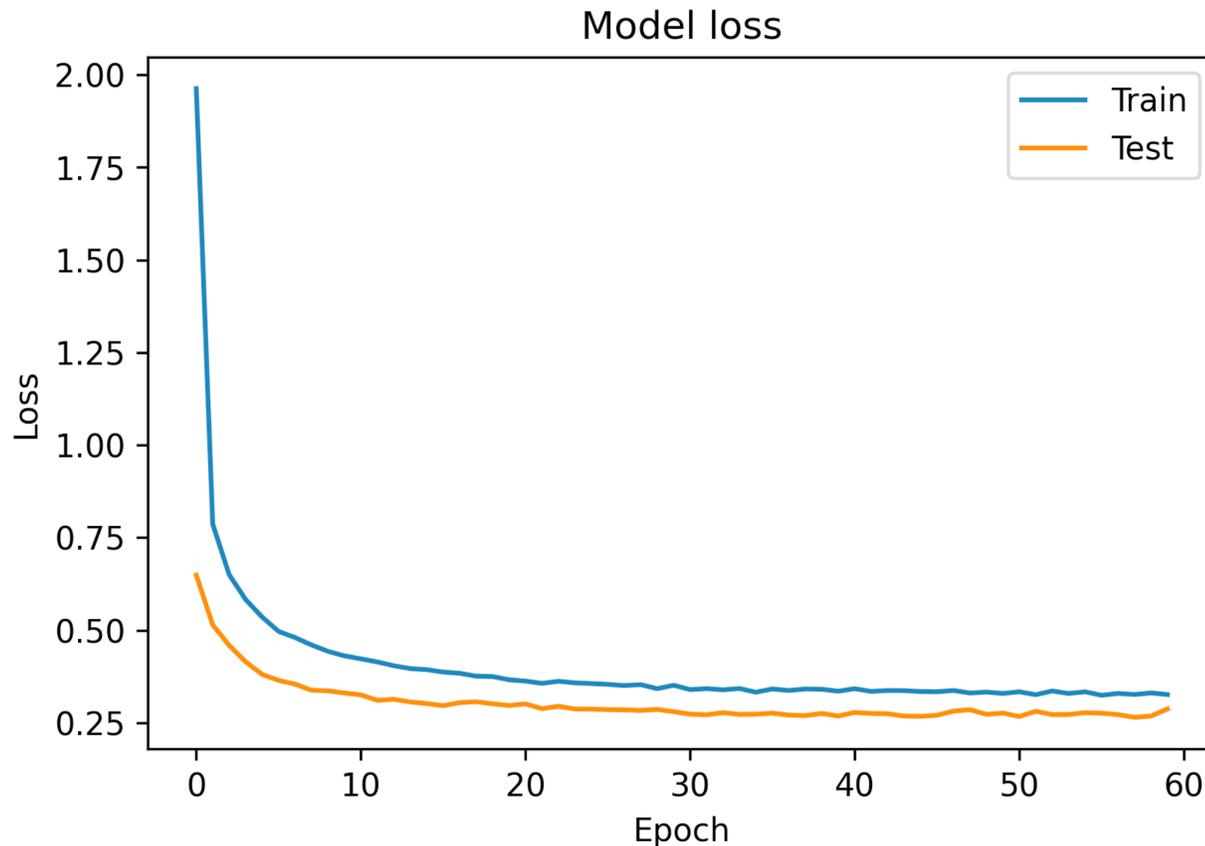
```
1 model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
2 history = model.fit(x_train, y_train, batch_size=100, epochs=50, validation_data=(x_test, y_test))
```

```
Epoch 1/50
600/600 [=====] - 164s 271ms/step - loss: 0.6860 - accuracy: 0.7502 - val_loss: 0.4218 - val_accuracy: 0.8489
Epoch 2/50
600/600 [=====] - 156s 259ms/step - loss: 0.4516 - accuracy: 0.8352 - val_loss: 0.3566 - val_accuracy: 0.8715
Epoch 3/50
600/600 [=====] - 157s 261ms/step - loss: 0.4010 - accuracy: 0.8544 - val_loss: 0.3183 - val_accuracy: 0.8836
Epoch 4/50
600/600 [=====] - 156s 261ms/step - loss: 0.3730 - accuracy: 0.8658 - val_loss: 0.3008 - val_accuracy: 0.8894
Epoch 5/50
600/600 [=====] - 156s 260ms/step - loss: 0.3500 - accuracy: 0.8727 - val_loss: 0.2846 - val_accuracy: 0.8944
600/600 [=====] - 155s 258ms/step - loss: 0.2290 - accuracy: 0.9143 - val_loss: 0.2083 - val_accuracy: 0.9234
Epoch 49/50
600/600 [=====] - 156s 259ms/step - loss: 0.2287 - accuracy: 0.9150 - val_loss: 0.2081 - val_accuracy: 0.9216
Epoch 50/50
600/600 [=====] - 155s 259ms/step - loss: 0.2272 - accuracy: 0.9152 - val_loss: 0.2035 - val_accuracy: 0.9255
```

5. 再次評估模型效能

- 完成訓練後再將各個epoch的loss變化畫成折線圖，由圖中的折線變化可以看出，不但整體loss降低，模型也沒有再出現過擬合的現象

```
1 import matplotlib.pyplot as plt
2 #train loss
3 plt.plot(history.history['loss'])
4 #test loss
5 plt.plot(history.history['val_loss'])
6 #標題
7 plt.title('Model loss')
8 #y軸標籤
9 plt.ylabel('Loss')
10 #x軸標籤
11 plt.xlabel('Epoch')
12 #顯示折線的名稱
13 plt.legend(['Train', 'Test'])
14 #顯示折線圖
15 plt.show()
```



5. 再次評估模型效能

- 最後輸出模型的整體loss與accuracy

```
1: score = model.evaluate(x_test, y_test)
2: print('loss:', score[0])
3: print('accuracy:', score[1])
```

```
313/313 [=====] - 3s 6ms/step - loss: 0.2091 - accuracy: 0.9236
loss: 0.2090652883052826
accuracy: 0.9236000180244446
```

- 實作1中模型的整體loss與accuracy

```
313/313 [=====] - 3s 8ms/step - loss: 0.3172 - accuracy: 0.8852
loss: 0.3171987235546112
accuracy: 0.885200023651123
```

4. Let's Get Greedy

當AI遇上股票



TABLE OF CONTENTS

- | | | |
|----------------------------|----------------------------|---------------------------|
| 1. CNN-1: 以AI預測股價-隔日
漲跌 | 2. CNN-2: 以AI預測股價-隔日
漲跌 | 3. LSTM: 以AI預測股價-隔日
漲跌 |
| 1.1. 安裝相關套件 | 2.1. 安裝相關套件 | 3.1. 安裝相關套件 |
| 1.2. 下載股價資訊 | 2.2. 下載股價資訊 | 3.2. 下載股價資訊 |
| 1.3. 觀察原始資料/日K圖 | 2.3. 觀察原始資料/日K圖 | 3.3. 觀察原始資料/日K圖 |
| 1.4. 將資料標準化 | 2.4. 將資料標準化 | 3.4. 將資料標準化 |
| 1.5. 建立、分割資料 | 2.5. 建立、分割資料 | 3.5. 建立、分割資料 |
| 1.6. 建立、編譯、訓練模型 | 2.6. 建立、編譯、訓練模型 | 3.6. 建立、編譯、訓練模型 |
| 1.7. 性能測試 | 2.7. 性能測試 | 3.7. 性能測試 |
| 1.8. 能怎麼胡搞 | 2.8. 能怎麼胡搞 | 3.8. 能怎麼胡搞 |

5. Let's Be More Realistic

- 模型不準確的原因
 1. 時間拉長、影響因素太多：金融風暴、戰爭、疫情
 2. 人為拉抬放空
- Solution:
 1. 加入其他影響股價的特徵資料
 2. 不直接測股價、只測漲跌機率

5) 專題成果發表

1) padlet: 2023AI選修

2) YouTube: 線上成發

5. 課程體驗：CNN實作股價預測

- 網址：<https://tinyurl.com/ai-stock>
- 登入Google Colab
- 開啟分享的ipynb

1. CNN-1：以AI預測股價-隔日漲跌

當AI遇上股票-CNN-1.ipynb

- 儲存自己的副本



5. 課程體驗：CNN實作股價預測

Date	2013-10-21 00:00:00+08:00	2013-10-22 00:00:00+08:00	2013-10-23 00:00:00+08:00	2013-10-24 00:00:00+08:00	2013-10-25 00:00:00+08:00	2013-10-28 00:00:00+08:00	2013-10-29 00:00:00+08:00	2013-10-30 00:00:00+08:00	2013-10-31 00:00:00+08:00	2013-11-01 00:00:00+08:00
Close	82.231735	83.353065	82.979301	82.231735	79.989059	81.857941	81.857941	82.231735	81.857941	81.857941

x_data[0]

y_data[0]

Date	2013-10-21 00:00:00+08:00	2013-10-22 00:00:00+08:00	2013-10-23 00:00:00+08:00	2013-10-24 00:00:00+08:00	2013-10-25 00:00:00+08:00	2013-10-28 00:00:00+08:00	2013-10-29 00:00:00+08:00	2013-10-30 00:00:00+08:00	2013-10-31 00:00:00+08:00	2013-11-01 00:00:00+08:00
Close	82.231735	83.353065	82.979301	82.231735	79.989059	81.857941	81.857941	82.231735	81.857941	81.857941

x_data[1]

y_data[1]

Date	2013-10-21 00:00:00+08:00	2013-10-22 00:00:00+08:00	2013-10-23 00:00:00+08:00	2013-10-24 00:00:00+08:00	2013-10-25 00:00:00+08:00	2013-10-28 00:00:00+08:00	2013-10-29 00:00:00+08:00	2013-10-30 00:00:00+08:00	2013-10-31 00:00:00+08:00	2013-11-01 00:00:00+08:00
Close	82.231735	83.353065	82.979301	82.231735	79.989059	81.857941	81.857941	82.231735	81.857941	81.857941

x_data[2]

y_data[2]

6. Q&A

