台灣人工智慧學校學習心得報告

陳彥欽

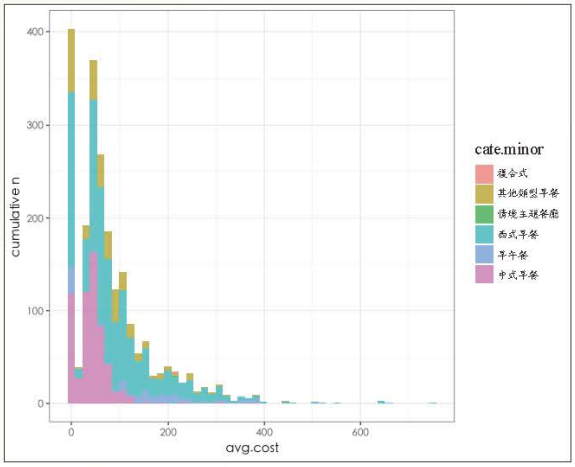
前言

有鑒於人工智慧在近年蓬勃發展，尤其 2018 年為台灣人工智慧元年，本行為了培養可以因應未來發展的人才，於 2017 年 12 月派作者參加台灣人工智慧學校－技術領袖培訓班第一期的考試，考試必須考五科，分別為微積分、線性代數、機率、統計與程式設計，通過這五門科目的考試後才可以進入學校就讀。很幸運地我考上了，學校上課時間為 2018 年 1 月 28 日～4 月 28 日為期三個月，台灣人工智慧學校由財團法人科技生態發展公益基金會及台灣資料科學協會主辦，中央研究院資訊科學研究所及中央研究院資訊科技創新研究中心協辦。學校將其定義為產業 AI 化的軍校，希望在短短三個月內將各領域的頂尖人才，培養成可以快速為台灣產業升級的人才，為了把一些學校學到的知識與心得帶回給本行的同仁，我開始撰寫了這篇報告。

而人工智慧對人的重要性就猶如火與電的發明那樣重要，中研院的 Project Theta Team 已經在之前幫助過不少台灣公司進行產業 AI 化，但是因為台灣每年產出的 AI 人才不到百人，所以才催生了台灣人工智慧學校的誕生，希望以一年培養 7,500 人的速度，讓台灣缺 AI 人才的問題不再是台灣發展 AI 的阻礙。

機率、統計與 R 語言

圖 1-1 為 R 語言產生視覺化圖片



資料來源：台灣人工智慧學校教學教材

R 語言算是相當高階的語言，與 Java 不同，例如資料處理用在 R 兩張 Table(表格) 合併，R 可以簡單完成，但是一般來說在 Java 要透過下 SQL 來控制資料庫才可以達到。換言之 R 可以直接在程式裡處理資料，Java 則需透過與資料庫連結才可以處理大量資料，所以助教說他們處理資料有時候用 R，寫 AI 程式用 Python，另外 R 除了處理資料以外，它的強項是資料視覺化，如圖 1 － 1 就是用 R 跑出來的圖，一般來說 R 的圖會比 Python 的漂亮許多。

統計課教了非常多統計上的公式與理論，在課堂上常用到的名詞有 mean（平均數）、median（中位數）、Standard Deviation（標準差）、Variance（變異數）、Outliers（離群值）、Linear Regression（線性迴歸），以下將簡單介紹這些名詞的意義：

1. mean（平均數）：即所有的數值加起來的值除以數值的總數。

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

2. median（中位數）：即所有數取中間的那一位數例如：1、2、3、4、5，中位數為 3。

3. Standard Deviation（標準差）：通常用來代表數的離散程度，值越大代表資料越離散。

$$SD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

μ為平均值(x)。

4. Variance（變異數）：即為標準差的平方，通常用來代表數的離散程度，值越大代表資料越離散。

5. Outliers（離群值）：例如年薪大家大約都在 20 萬到 100 萬之間，但是有一個數年薪 3 億，這個 3 億的數即為離群值。

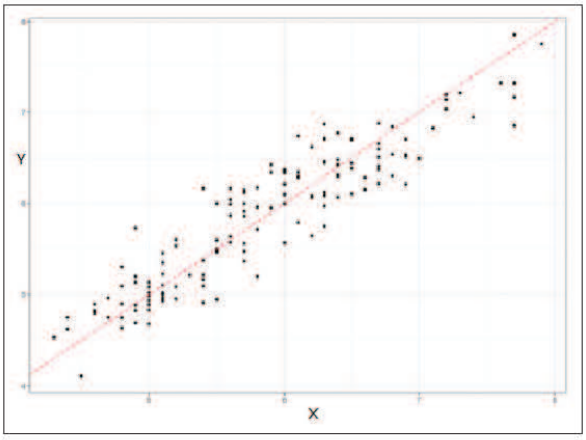
6. Linear Regression（線性迴歸）：這個比較複雜，簡單來說就是用 X 值預測 Y 值，母體資料可以找到一條線性的方程式來預測 Y，一般來說此觀念可以用在 Machine Learninig 房價預測上。

在 AI 的路上我們必須知道資料對機器學習（Machine Learning）相當重要，以 Machine Learning 舉例通常要有足夠的訓練資料才可以訓練（Train）出一個好的模型（Model）來預測房價。所以資料的預先處理以及清洗相當重要，常常資料過來的時候已經有缺失值（Missing Value）或不對的值，例如房價預測問題某一個特徵（feature），屋齡出現了年月日的資料，例如屋齡：10 變成屋齡：2017/03 這樣錯誤的資料，缺失值的處理通常可以補 0 或是用 mean 來補。另外統計也有一些辦法如 K-means clustering 或是補值為 median 也可以。另外處理不對的值其實也很簡單，把 Training Data（訓練資料）輸入演算法如 RandomForest 中，Train 的時候有 String Python 會顯示錯，把資料稍微處理一下就可以了。

另外說明一下離群值，以我們作業房價預測為例，離群值有時候必須從 Training Data（訓練資料）刪除，因為它會影響房價預測回歸問題的準確性。因為我們要預測的是一般條件下的房價，不想讓豪宅價格進入母數中，因此在做分析時，將一般房價與豪宅房價的數群分開處理，這樣模型（Model）預測才會準確。

而我所說的 Model 如果是線性的回歸線，假設房價預測只有一個特徵值（feature）X，那麼 Model 可能像圖 1－2 一樣，然後預測的房值為 Y，但是會影響房價的 feature 很多，例如屋齡、附近的交通設施數、建材等等，所以多維的線性回歸圖通常是畫不出來的。

圖 1－2



資料來源：台灣人工智慧學校教學教材

機器學習概論與演算法

現今人工智慧最有名的兩個名詞莫過於機器學習（Machine Learning）及深度學習（Deep Learning）。深度學習其實是機器學習的一個分支，而機器學習又是人工智慧的一個分支。機器學習可分為監督式學習（Supervised Learning）與非監督式學習（Unsupervised Learning），這兩個學習的解釋如下：

1. 監督式學習（Supervised Learning）：
訓練 Model 的時候必須先告訴機器這張圖片是狗還是貓，也就是有 Label（標籤）的資料，通常會是一張圖片配一個 Label，這個 Label 會標記是狗或是貓。

2. 非監督式學習（Unsupervised Learning）：訓練 Model 的時候沒有 Label，機器不知道是狗是貓，機器會透過大量的資料自己分類出不同的類別。一般現今的影像辨識例如有名的 Yolo v3 就是監督式學習，但是人工智慧領域有名的 GAN（對抗生成式網路）則是非監督式學習，那麼一般我們在做 Machine Learning 有個重要的圖如圖 2－1。

圖 2－1

資料來源：台灣人工智慧學校教學教材

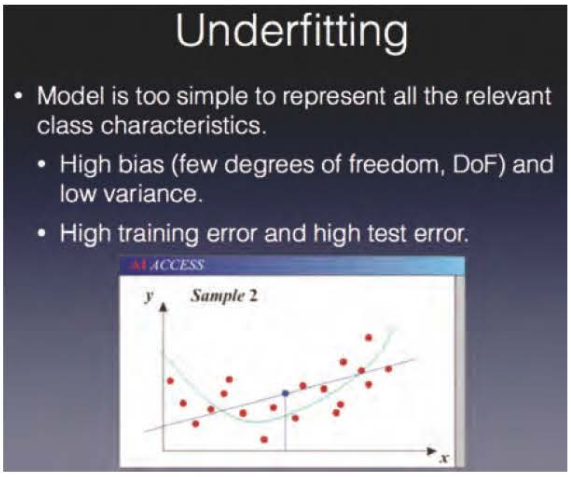
這兩個多月的學習與競賽與作業用到的大多是監督式學習的回歸（Regression）與分類（Classification）問題。當然非監督式學習也有例如 K-means Clustering 演算法[註1] 即是群集問題，通常是要預測什麼價格會回歸問題，例如房價預測。不過最近很流行的股市預測，因為跟時間序列比較有關係，所以它是用 RNN（遞迴式類神經網路）來做股市預測。另外分類（Classification）問題，例如手寫辨識（數字 0~9，總共 10 個類別）或是我們參加的 Tbrain 趨勢科技無檔案的惡意程式辨識[註2] 問題，辨認此檔案是否為惡意檔案（其為二元分類問題就是答案是是或否）。

老師在上課時解釋相當多與統計學相關的機器學習演算法，例如：DescisionTree、RandomForest、SVM、PCA、Adaboost、XGboost，我們最後建立機器學習模型都是用 XGboost，因為很多競賽的第一名或金牌都是用它來建立模型。當然參數也要調一下，一般我都是調 n_estimator=100（用幾棵樹做學習），max_depth=10（樹的高度）。另外評估模型好壞與否有兩種比較常用的評分方式，一是 AUC（越高 Model 越好），另一是 RMSE（越低 Model 越好）。此外還有兩個名詞要特別介紹，一為 Overfitting(過度擬合)，另一為 Underfitting（好像沒中文翻譯），兩者都是在做實驗時會碰到的現象，通常 Overfitting 會發生在訓練資料過少或 feature（特徵值）過多的時候，解釋如下方三張圖（圖 2－2，圖 2－3，圖 2－4），Overfitting & Underfitting 為圖中那條藍色的就是 Model 的曲線。

圖 2 － 2

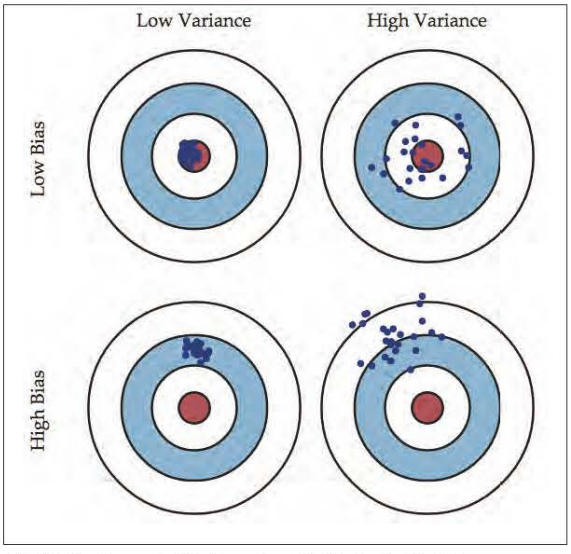
資料來源：台灣人工智慧學校教學教材

圖 2 － 3



資料來源：台灣人工智慧學校教學教材

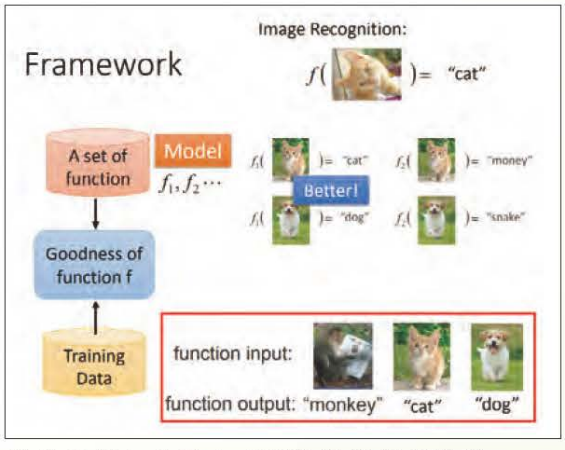
圖 2 － 4



資料來源：台灣人工智慧學校教學教材

接著介紹什麼是 Machine Learning（簡稱 ML），ML 有三大步驟：Step 1（Define a set of function）：先定義一組 functions（可能有成千上萬無窮無盡）。Step 2（Goodness of function）：讓機器學習判斷什麼是對什麼是錯，什麼是狗什麼是貓的過程。Step 3（Pick the best function）：取用最好的 function。這三個步驟好比把一隻大象塞進冰箱。接著如圖 2 － 5 所示，在尋找 function 的過程中，譬如 f1（輸入狗的圖片）= 狗，f1（輸入貓的圖片）= 貓，f2（輸入狗的圖片）= 猴子，f2（輸入貓的圖片）= 鳥，這樣看來，f1 可能是一個比較對的 function。當然也有可能 f1（輸入鳥的圖片）= 雞，這樣它也可能就會判斷錯誤。所以 ML 的整個過程，就是經由大量 Training Data 學習最後，挑出一個最好的 function f*，這也是我整篇心得所談到人工智慧的 Model。

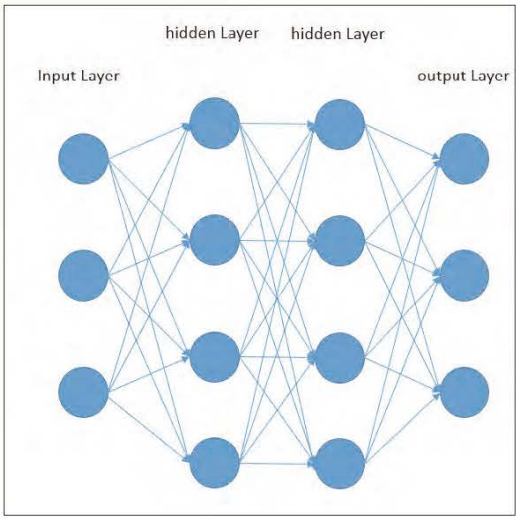
圖 2 － 5



資料來源：台灣人工智慧學校教學教材

深度學習理論 DNN&CNN

圖 3 － 1

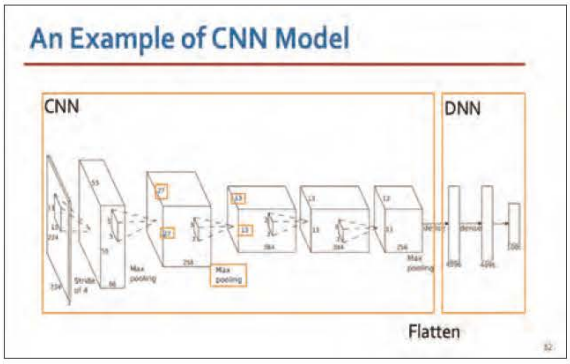


資料來源：彰銀辦事員陳彥欽

現今人工智慧有突破除了 GPU 速度提升以外還有深度學習的發明，在 ImageNet 資料庫圖像辨識（辨識 1,000 種類別）比賽當中，微軟的 152 層 Residual Net 俗稱 ResNet，對於圖片的辨識率首度超越人類。那麼何謂深度，其實就是深度類神經網路（Deep Neural Network=DNN）的深度，例如圖 3 － 1 的 Hidden Layer 越多越好。科學家做了很多實驗後，發現類神經網路越深圖片辨識度越高。理論上越多層 NN 可以學到的圖片 Feature 越多，因此辨識度越好。但是到底要幾層以及每一層要多少 Neuron 呢？（以圖 3 － 1 來看，一個圓圈叫一個 Neuron，越往右越深 =Deep，越往上跟下增長越胖）老師上課有說明，這其實都是人主導設計，但是可以經由實驗來決定每層要幾個 Neuron，總共要幾層。但是為什麼需要 GPU 呢？因為 Deep Learning 在學習（或訓練）時，會用到非常多的矩陣（Matrix）運算。

什麼是捲積類神經網路（Convolutional Neuron Network）？其實可以說是 DNN 的變形，如圖 3 － 2。原本圖像辨識 DNN 是將一張圖 Flattern 成一維向量，輸入 DNN 的 Input Layer，但是 CNN 是直接輸入一張例如 200x200x3（x3 代表 RGB），然後用一個 3x3 的 kernel 去掃（或者說內積這張圖片）這張圖片，因為是用影像處理的方式來處理，所以找出來的 feature 會比 DNN 更好，辨識度也越高。

圖 3 － 2



資料來源：台灣人工智慧學校教學教材

另外介紹 DNN 裡面常用名詞：

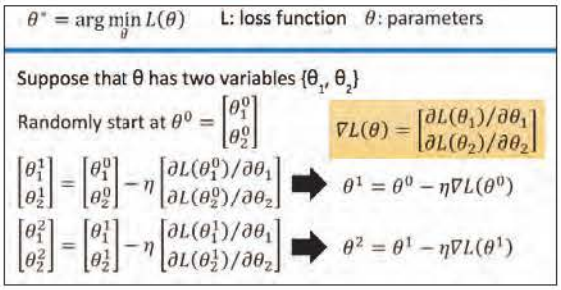
1. Loss Function（損失函數）：用來估量你模型的預測值 f（x）與真實值 Y 的不一致程度。
2. Activation Function：用來讓 DNN 找出非線性關係的函數。

3. Gradient Descent（梯度下降法）：用來求 loss function 最佳解的演算法。

4. Back Propagation：用來求 Gradient Descent 的 Gradient 的快速方法。

5. Gradient Descent（梯度下降法）重要公式：（見圖 3 － 3）

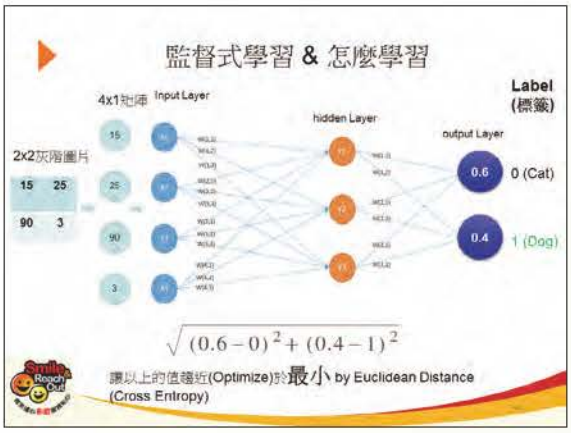
圖 3 － 3



資料來源：台灣人工智慧學校教學教材

接下介紹深度學習是怎麼學習，首先由圖 3 － 4 作一解釋，左邊為一張 2x2 的灰階圖片，先將其拉成一維 4x1 的矩陣，最後輸入事先設計好的類神經網路。但是在做矩陣運算前，圖片中的 w（1~n,1~n）=weight 值電腦會先預設初始值，經過第一次矩陣運算後，算出兩個結果 0.6 及 0.4。先前有說過監督式學習會先告訴機器這張圖片是什麼，下面這張圖就是在說這張 2x2 的灰階圖片其實是一張狗的圖片，標註的方式是用 0&1 表達，若是 [0,1] 代表狗，若是 [1,0] 代表貓，我們使用監督式學習訓練人工智慧 Model 時，大量訓練資料都會是一張圖片配一個 Label（標籤），用來標註這張圖片屬於哪個類別，用以辨別是狗或是貓。所以經由下面的 Euclidean Distance 公式可以算出一個值約 2.68 多，我們會希望這個值越小越好，因為假設在我們更新 weight 值以後，可以使輸出值為 0 與 1，代表著我們人工智慧 Model 可以精準預測這張圖片是 Dog（狗）了（也就是 Euclidean Distance 的值為 0）。不過這是最理想狀態，一般都只能將其做到最佳化，達到一個最小值而已，然而最佳化有賴於數學方法 Gradient Descent& Back Propagation 來達成。在最佳化的過程中，類神經網路那些線所代表的 weight 值會一直不段更新，直到找出 Gradient Descent 的 Local Minima 為止。簡言之，整個監督式學習－深度學習的過程就是一種最佳化的過程，而這個過程也稱之為讓機器一直學習或說是訓練。

圖 3 － 4



資料來源：彰銀辦事員陳彥欽

網路爬蟲 &DNN&CNN 實戰演練

網路爬蟲其實就是以寫程式的方式將我們人工智慧要的訓練資料抓下來，要訓練（Train）一個好的人工智慧 Model 常常仰賴大量的學習或者說訓練資料，要寫爬蟲程式通常要懂一些 HTML 與送 Request 的方式，其分成 GET 與 POST。要爬蟲前其實可以先從 Github 查看一些別人先寫好的 Sample Codes，另外若是要爬 Facebook 文章，其實 Facebook 有提供 API 供人爬蟲，爬蟲其實是一直對伺服器送出需求（request）的過程，所以有些網站被爬太多次後就不能爬了。

學校教的 DNN&CNN 程式設計都會用到 TensorFlow，那什麼是 Tensor（張量），張量是現代機器學習的基礎。它的核心是一個數據容器，DNN 會用到一維的 Tensor 即是一維的向量（Vector），CNN 會用到二維的 Tensor 即二維的圖片，那麼 Flow 其實就代表資料在類神經網路中流動的意思，這一部分助教的實作 Code 先從微分教起，再教到 Gradient Descent 與 Backpropagation 實作手刻程式，相當扎實！

學習過程中先用 DNN 做一個類神經網路來訓練一個人工智慧 DNN，可以辨識出手寫數字，接著是辨別辛普森家族的圖片分類器，但是 DNN 的 training accuracy 只達到 80%，所以必須用 CNN 精進，最後作業是用 CNN 做辛普森家族圖片辨識器，最後我的實驗結果是 training accuracy 達到 100%，但是不是 testing accuracy，因為 training 時 CNN 已看過辛普森家族所有圖片了所以 accuracy 可以達到 100%，但是如果載入這些訓練資料以外的資料，可能 accuracy 就不會那麼高，甚至有可能 overfitting，所以可以把 training 資料當作平時的作業或小考，testing 資料當作真正的段考或聯考。最後助教教一個大學教授常用叫 Keras，相較於 TensorFlow，Keras 程式好寫太多了，很多人期中考時都是用 Keras 通過測驗，當然我是兩種都用。圖 4-1 是用 TensorFlow 搭建類神經網路，圖 4-2 是用 Keras 搭建類神經網路，可以看得出來 Keras 好寫多了。建議新手先學 Keras 就好，其實 Keras 只是把 TensorFlow 的 Code 包好，但是它的缺點就是 Train 的速度比較慢，並且 Keras 可能會遇到它包的函式使用者不滿意時，就必須用 TensorFlow[註3] 自己撰寫程式了。

圖 4 － 1

```
conv_layer = tf.layers.conv2d(x_tensor, conv_num_outputs, kernel_size=conv_ksize, strides=conv_strides, activation=tf.nn.relu)
conv_layer = tf.layers.max_pooling2d(conv_layer, pool_size=pool_ksize, strides=pool_strides)
```

資料來源：台灣人工智慧學校教學教材

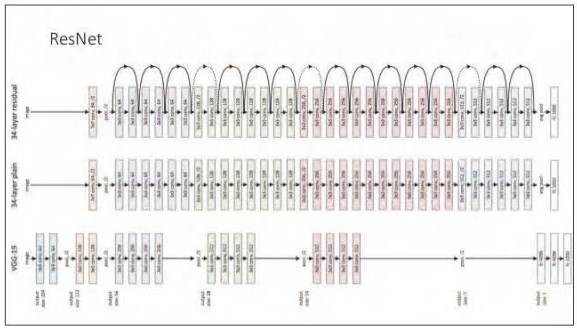
圖 4 － 2

```
model.add(Conv2D(128, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

資料來源：台灣人工智慧學校教學教材

轉移學習與實戰演練 & 趨勢科技 Kaggle 競賽

圖 5 － 1



資料來源：https：//arxiv.org/abs/1512.03385
且已取得原作者授權使用

轉移學習簡單來說就是一種類神經網路，利用別人 Train（訓練）好的類神經網路接到自己設計的類神經網路。例如要做一個貓狗分類器的類神經網路，其作法就是類神經網路前端可以使用最知名的 ResNet（如圖 5 － 1），後面應用 fully connected network 接續（自行設計的類神經網路），然後 output（輸出）兩個結果（因為只辨識狗跟貓）。老師上課舉了很多例子，其中我最喜歡的是照片風格轉換，如圖 5 － 2 把一張照片轉換成卡通風格，想想如果以後有這種技術，就可以減少動畫師很多工作時間，或許這也是未來 AI 的另一種應用。

圖 5 － 2

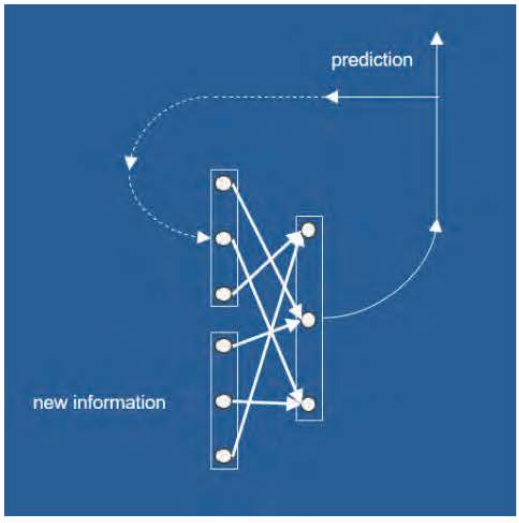
資料來源：台灣人工智慧學校教學教材

接著就是學校為期兩天的趨勢科技 Kaggle 競賽[註4]，其實總共比了三個禮拜。這個競賽的需求就是趨勢提供 8,000 萬筆某段時間使用者的 log 資料，內容大概就是使用者使用某種裝置，接著用某一個軟體開啟檔案，然後用了某趨勢科技的掃毒軟體掃這個檔案，包含其檔案開啟時間，整個比賽就是要用 Machine Learning 的方式來 Train 一個人工智慧的 Model，將測試資料輸入以後，辨識出某個檔案是否為惡意軟體。一開始我們團隊六個人討論後開始做 Feature Engineering，因為有時間我們可以將時間分成四個時區做成 feature，或是說某個檔案 ID 總開啟次數超過 20,000 次作為一個 feature，或是 ProductID 有 32 種，所以可以用 One Hot Encoding 的方式作成 32 種 feature，之後我也有試

過 Deep Learning，但是效果沒比較好，最後最好的成績是用傳統 Machine Learning 的方法，演算法用的是最強的 XGBoost，最後得到了 Private Leaderboard 第 68 名（總團隊共 424 隊－名次為前 16.5%）的成績。另外值得一提的是，全部有 8,000 萬筆資料，將所有的資料讀到 RAM 一次要 5 分鐘，且佔用 RAM 19.6GB（學校配給 48GB RAM），可見人工智慧要用到的資料量非常大，難怪有人稱人工智慧的人叫資料科學家（Data Scientist）。

RNN 遞迴式類神經網路（LSTM），NLP 自然語言處理

圖 6 － 1



資料來源：台灣人工智慧學校教學教材

RNN（遞迴式類神經網路）常用在股市預測，其實簡單來說就是用過去的資料來預測未來的事情（如圖 6 － 1），因為類神經網路太深時，傳統的 RNN 有梯度消失（Vanishing Gradient），Gradient 會有消失的問題。這邊是 wiki 的解釋：in some cases，the gradient will be vanishingly small，effectively preventing the weight from changing its value 的問題，所以近來會開始用一種新的方法叫 LSTM（Long Short-Term Memory）取代傳統的 RNN。目前聽到 RNN 大部分都是使用 LSTM，LSTM 的類神經網路（簡稱 NN）模型相當複雜，除了模型複雜其數學也很複雜。

另外還上了 NLP（Natural Language Processing= 自然語言處理），上了這門課一樣感覺它博大精深，現在有名的智慧音箱及聊天機器人（ChatBot）都跟 NLP 有關係，要做一個智慧音箱是相當困難，據我所知必須先懂語音識別（Speech Recognition），因為我們必須先把一般人講的語音（以科技大擂台的比賽為例，還需處理掉背景音效並轉成文字，然後再用中國做的斷詞工具 jieba 對文字作後處理。）圖 6 － 2 是一些 PTT 文章用 jieba 斷詞後的結果，後處理並沒有想像中簡單，因位還要做文字特徵工程（TF-IDF）[註 5]。

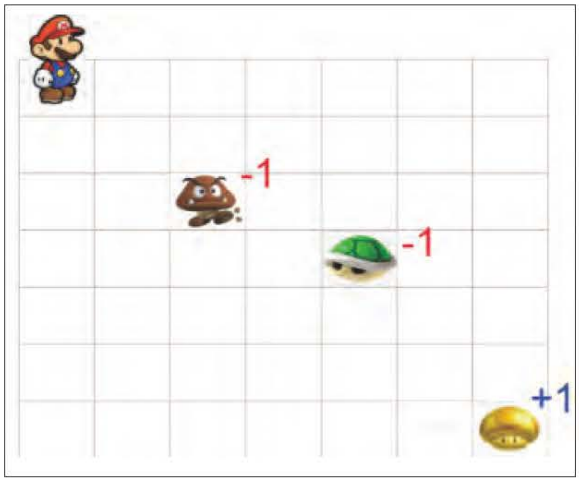
圖 6 － 2

```
[['韓瑜', '協志', '前要', '正', '女演員', '周子', '瑜', 'TWICE', '團裡裡面', '台灣', '人', '正', '兩個', '要當', '鄉民', '老婆', '選', '五樓', '真', '勇氣'], ['dear', 'all', '逢甲', '碟仙', '發生', '民國', '七十五年', '三月中', '事情', '一堆', '大學生', '玩', '碟仙', '後發', 'bbs', '成功', '預測', '921', '地震', '小弟', '預言', '都還沒', '出生', '後面', '說', '預言', '一百', '一十六年', '兩岸', '統一', '統一', '對岸', '對岸', '統一', '應該', '不用', '猜', '真的', '存在', '預言', '這種', '事情', '倒底', '116', '被統', '知道', '資料庫', '發文', '日期', '輕鬆', '改變', '拍照', '狀況', '下', '碟仙', '真的假', '有沒有', '科學', '經驗', '法則', '破解', '謠言', '真實', '八卦'], ['晚上', '好', '預備', '唱', '風雲', '山河', '動', '國軍', '早上', '唱', '有沒有', '相關', '八卦', 'Sent', 'from', 'JPTT', 'on', 'my', 'Xiaomi', 'Redmi', 'Note'], ['明天', '早起', '睡覺', '旁邊', 'Youtube', '眼睛', '壞掉', '有沒有', '方法', '早點', '睡', '掛', 'Sent', 'from', 'JPTT', 'on', 'my', 'HTC', 'D10i'], ['一段時間', '注意', 'LOL', '發現', '各大', 'LOL', '討論區', '人數', '明顯', '下降', '趨勢', '實在', '令人', '驚訝', '曾經', '一時', '遊戲', '霸主', '漸漸', '過氣', 'LOL', '確實', '撐', '久', '現在', '算', '厲害', '遊戲', '玩久', '會屬']]
```

資料來源：台灣人工智慧學校教學教材

Reinforcement Learning，CNN（Object Detection with YOLO）

圖 7 － 1



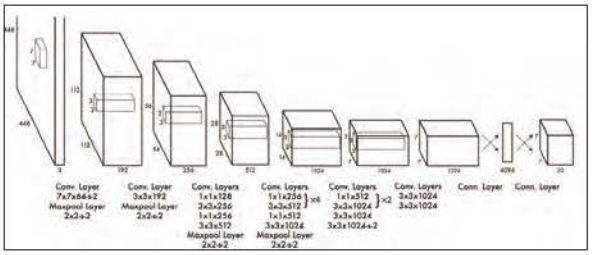
資料來源：台灣人工智慧學校教學教材

增強式學習簡單來說就是像圖 7 － 1 的馬利歐，他會嘗試往各個方向走，碰到妖怪扣一分，拿到金香菇得一分，走到一般格子得零分，電腦就會一直訓練跑這個二維地圖，直到馬利歐跑出一個 Q-Learninig 的 Table 就可以得到最高的分數，理論上叫做 Reward= 獎勵。人工智慧最後會訓練馬利歐在每次玩遊戲時都盡量走得分的路線，然後最後拿到金香菇過關。這邊的應用比較有名的是訓練人工智慧玩遊戲，不同於以前電動遊戲是事先寫好在遊戲程式碼內的人工智慧，增強式學習的 Actor-Critic 的方式是類似人在玩電動的角度去訓練出的人工智慧，目前有不少應用是將人工智慧套在 Atari 遊戲[註6] 上。當然也有人將增強式學習用在 3D 虛擬賽車遊戲上，那我的想法是目前自駕車也許就可以透過增強式學習＋ CNN with YOLO 達成，我實際有查了網路上的論文[註7]，確實是用增強式學習＋ CNN 但不是一定有用到 YOLO，另外 3D 世界虛擬開車的增強式學習影片[註8]，其實也是蠻厲害且有趣的。

接著要介紹 YOLO，目前最新的版本為 YOLO v3，它是一種物件偵測的人工智慧，圖 7 － 2 是其類神經網路的架構，原作者[註9] 寫道，we frame object detection as a re-gression problem to spatially separated bounding boxes and associated class probabilities，也就是說把物件偵測當作一種回歸預測問題，它的方法有三個步驟：（1） resizes the input image to 448 × 448、（2） runs a single convolutional net- work on the image、（3） thresholds the resulting detections by the model's confidence。

有興趣的話可以再去詳看原作者的論文，另外還有 YOLO 經典測試影片[註10]也非常厲害。先前提到的自駕車可以用 YOLO[註11]，我覺得雖然 YOLO 辨識度已經很好，但還是要考慮天候因素，因此我認為一般來說應該還要配合物體偵測雷達＋ YOLO ＋紅外線攝影機（夜間物體偵測），這樣自駕車才會比較完善安全，提供 nVidia 展示自家自駕車系統影片[註12]供參。至於增強式學習在自駕車的應用，簡單來說就是在訓練 Model 時，走沒有障礙的路得分比較多，撞到障礙物扣分，速度越快分數越高，當然判斷障礙物就是用 YOLO 物件偵測（Object Detection）。

圖 7 － 2



資料來源： https：//pjreddie.com 且已取得原作者授權使用

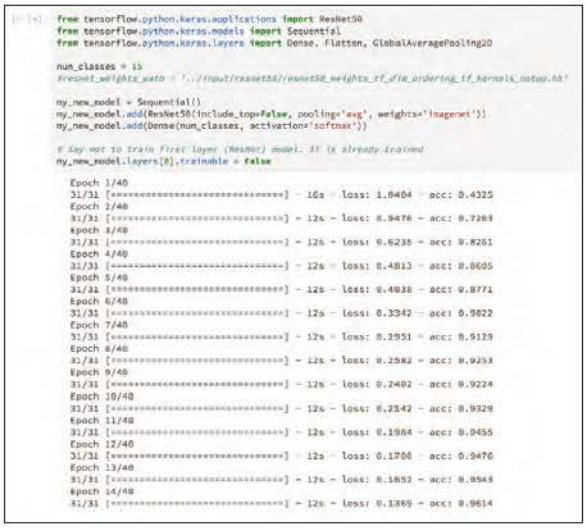
Generative Adversarial Network（生成對抗式網路，簡稱 GAN）

GAN 是一種印鈔機與驗鈔機概念的人工智慧類神經網路， GAN 輸入一個參數 z（通常是一維向量），印鈔機產出一張鈔票 （驗鈔機會事先看過真的鈔票），然後驗鈔機驗鈔，判斷結果如果是真的，整個網路就輸出鈔票，假的就打回去。整個網路就會變成印鈔機想盡辦法印出鈔票騙過驗鈔機，但驗鈔機也會進化（類神經網路經由 loss function 使用 Gradient Descent 方法的最佳化過程，當然一般來說 Optimizer 都用 Adam）找出印鈔機的弱點驗出假鈔，整個網路就這樣一直訓練學習成一個 GAN 的人工智慧類神經網路。雖說名字叫對抗生成網路但其實也可以看成是合作，就看你怎麼解讀。應用上有少女漫畫生成器[註13]或是可以把某個影片裡面的人臉換成你想要的臉，當然這讓我直接聯想到的是犯罪事情（犯罪人的臉換成你想要陷害的人的臉），但這對於電影產業應有其用處！提供網路上換臉影片[註14]供參，其實蠻誇張，這個換臉方法叫 Deepfake。

期中考 &IBM x 遠傳競賽

圖 9 － 1 為 keras 寫的 ResNet50 Transfer Learning 圖片辨識程式碼以及 Training 過程：

圖 9 － 1



```
from tensorflow.python.keras.applications import ResNet50
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense, Flatten, GlobalAveragePooling2D

num_classes = 15
resnet_weights_path = '../input/resnet50/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5'

my_new_model = Sequential()
my_new_model.add(ResNet50(include_top=False, pooling='avg', weights='imagenet'))
my_new_model.add(Dense(num_classes, activation='softmax'))

# Say not to train first layer (ResNet) model. It is already trained
my_new_model.layers[0].trainable = False
```

```
Epoch 1/40
31/31 [==============================] - 16s - loss: 1.8404 - acc: 0.4325
Epoch 2/40
31/31 [==============================] - 12s - loss: 0.9476 - acc: 0.7283
Epoch 3/40
31/31 [==============================] - 12s - loss: 0.6235 - acc: 0.8261
Epoch 4/40
31/31 [==============================] - 12s - loss: 0.4813 - acc: 0.8605
Epoch 5/40
31/31 [==============================] - 12s - loss: 0.4038 - acc: 0.8771
Epoch 6/40
31/31 [==============================] - 12s - loss: 0.3342 - acc: 0.9022
Epoch 7/40
31/31 [==============================] - 12s - loss: 0.2951 - acc: 0.9129
Epoch 8/40
31/31 [==============================] - 12s - loss: 0.2582 - acc: 0.9253
Epoch 9/40
31/31 [==============================] - 12s - loss: 0.2402 - acc: 0.9224
Epoch 10/40
31/31 [==============================] - 12s - loss: 0.2142 - acc: 0.9328
Epoch 11/40
31/31 [==============================] - 12s - loss: 0.1984 - acc: 0.9455
Epoch 12/40
31/31 [==============================] - 12s - loss: 0.1700 - acc: 0.9476
Epoch 13/40
31/31 [==============================] - 12s - loss: 0.1652 - acc: 0.9543
Epoch 14/40
31/31 [==============================] - 12s - loss: 0.1369 - acc: 0.9614
```

資料來源： 台灣人工智慧學校教學教材

期中考為兩天上機考，總共2題，一題是要預測某筆交易是否為詐欺，Training Data 有 250 萬筆，不過不用害怕擔心，因為 Training Data 多反而好，人工智慧最怕資料太少，Model Train 不起來。其實這題就是資料前處理，用 type：CASH-IN：存錢、CASH-OUT：提款、DEBIT：支付、PAYMENT：付款、TRANSFER：轉帳資料做 One Hot Encoding，這一題我大概花 1 小時就做出來了，成績如圖 9 － 2，通過考試 Baseline AUC：0.9。

第二題就比較困難，是用 CNN 方法的圖像辨識問題，有 2,000 多張圖片，設計一個 CNN 網路可以辨識出 15 種類別的類神經網路人工智慧。第一天我用 TensorFlow 實作，一開始比較難的是資料前處理，光處理資料要怎樣讀進來跟做 Data Augmentation（意即將圖片轉幾度，放大縮小等等的事情讓 Training Data 比較多樣性！）就花了很多時間。第一天晚上大概做到 AUC 0.3 ，Baseline 是 0.68888，後來去請教助教，他建議用 keras，因為 TensorFlow 太難了。我繼續嘗試了很多種方法例如 leakey relu 等等的，改變各種類神經網路的設計，最後達到 AUC 0.74857（如圖 9 － 3），總算是超過 baseline AUC 0.6888。不過因為看到有人 AUC 0.9 就去打聽他們怎麼做的，原來他們用微軟的 ResNet50 ＋ Fully Connected NN 來做，完全不是用傳統 CNN。最後我也花了點時間用 keras，實作出來果然可以達到 AUC 0.9 成績（如圖 9 － 4）。考完這次期中考，後來想到華為 Mate 10& 華碩 Zenfone 5 推出有 AI 照片場景辨識的功能，其實就是我們這次期中考的考題。我只要用 ResNet50（50 層版本）＋ Fully Connected NN output Layer 為 16 個來 Train，就可以做出相機 AI 16 種場景辨識的功能了。不過我個人覺得，這只是軟體，其實 ResNet 有 152 層，所以也要很強的硬體配備支援，要不然光照片輸入 GPU ＋ CPU 可能就要耗費很長的時間。

圖 9-2

資料來源：台灣人工智慧學校教學教材

圖 9-3

資料來源：台灣人工智慧學校教學教材

圖 9-4

資料來源：台灣人工智慧學校教學教材

緊接著是為期三天半的 IBM x 遠傳台灣人工智慧技術大賽，這個比賽主要是提供參賽者遠傳用戶某段時間某經緯度使用了那些 App、一天使用了多少的 3G/4G 網路資料、某一用戶某一天各地遠傳基地台所記錄的經緯度位置（三角定位法），以及 13 天多位使用者的移動最大距離等資料，最後要預測出這些使用者在 2018/03/16 這天的最大移動距離（遠傳給了 2018/03/03~2018/03/15 的資料）。像這種 Machine Learning 比賽，我們一開始要做資料前處理與特徵工程，例如在某一基地台定位的經緯度下，某一使用者使用 Facebook、Google、Line 的次數；其使用的作業系統為 iOS 或 Android；他經緯度的所在區域例如是新北市板橋區等等的資料，最後把 Training Data 載入 XGBoost 演算法去 Train Model，最後用 Testing Data 進 Model，預測出 2018/03/16 這天所有使用的最大移動距離。這部分其實有些複雜，不過最後的概念是，應用前一天的資料預測後一天的資料（所以 Testing Data 是 2018/03/15 的資料），有點 RNN 的概念，但我們沒有用 RNN 做，因為時間只有三天半。圖 9 － 5 為我們團隊「無所遁形」最後的排名，從 rmse（root mean square error）看得出來大家都很拼，分數都相當接近（另帶一提這比賽是自願參加，學校大概 60% 的人參加，每隊 4~6 人），複賽要前六名，雖然沒有進入複賽，但是個相當難得的經驗。我個人覺得其實回到工作崗位後要研究的東西還很多，大概也要花好長一段時間才能成氣候吧！

圖 9 － 5

台灣人工智慧學校

名次	隊伍名稱	提交次數	最佳分數(越低越佳)	最近上傳時間	入圍複賽
1	就讀遠傳，很有訊號！）	44	0.251065	2018-03-31 16:49:44	是
2	AIA_ALL4ONE	32	0.251799	2018-03-31 16:49:44	是
3	CCI	38	0.252128	2018-03-31 16:49:44	是
4	妙蛙種子~	33	0.25327	2018-03-31 16:49:44	是
5	AIA 吃喝玩樂為首要	51	0.253304	2018-03-31 16:49:44	是
6	AIA_PhiChiPi	51	0.254024	2018-03-31 16:49:44	是
7	傲!	39	0.254063	2018-03-31 16:49:44	
8	Northern Lights	43	0.254969	2018-03-31 16:49:44	
9	King\'s Pawn	34	0.2579	2018-03-31 16:49:44	
10	BES	26	0.258858	2018-03-31 16:49:44	
11	AIA_8743	20	0.258965	2018-03-31 16:49:44	
12	DD	16	0.259046	2018-03-31 16:49:44	
13	AIA_TOGO	24	0.259469	2018-03-31 16:49:44	
14	AIA-stickers	24	0.260009	2018-03-31 16:49:44	
15	JJ	32	0.260536	2018-03-31 16:49:44	
16	AIAIAIAIAIA	10	0.260598	2018-03-31 16:49:44	
17	無所遁形	26	0.261572	2018-03-31 16:49:44	

資料來源：台灣人工智慧學校教學教材

期末專題

最後一個月我們做的是成功大學測量所－房價預測，有別於我們先前實戰演練的作業，這次不像先前提供很多資料，這次成大提供了 2,600 多筆資料，我們自己還必須上實價登入網站下載資料。首先我們先定義這個問題應該是屬於回歸問題，所以大概會像以往一樣用 Python 的 XGBoost 或 RandomForest 演算法來建立人工智慧 Model。接著我們建立了一些 Features，大約 70 多種，然後把資料載入 XGBoost 演算法建模，最後產出的 R square（R2）值只有 0.55，跟其他組別 0.93 比較後，討論發現用每坪單價當 Label（Ground Truth）會有問題，必須把 Label 改成房屋總價，修改後 R2 值馬上跑到 0.83，至於為什麼後面

會解釋。我們知道 R2 值越高越好，代表房價預測越準確，先前有提到一個好的人工智慧 Model 有賴於大量的訓練資料，以上課經驗來看，2,000 多筆資料其實是不夠的，所以之後幾個星期我們便開始持續在實價登入網站下載台中市 2013Q1 ～ 2018 年 Q1 實價登入資料，大約 19 萬筆。另外為了要萃取 Features 出來，經由全國門牌地址定位服務，將房屋大概的住址轉成經緯度位置（TWD97 格式）， Features 大概有以下幾種：

物件基本屬性	物件所在地理位置	物件周邊機能
・建物型態 ・屋齡 ・坪數 ・現況格局 ・建材 ・有無車位	・行政區別 ・商業住宅面積 ・工業住宅面積 ・都市計畫區域* ・距離台鐵距離*	・便利商店 ・學區 ・公園 ・市場 ・停車場 ・其他…

經緯度的功用讓我舉個例子，例如說我們有一個房屋的經緯度位置，我們會以這個位置為中心點畫一個半徑約 500m~2,000m 的圓，然後算其圓內所涵蓋的商業區面積，而這個面積就會當做一個 Feature，或是說以這個位置為中心點畫一個半徑約 500m~2,000m 的圓，算出這個圓內有多少家便利商店或是小學。不過由於在處理這些 Feature 時，例如算出商業住宅面積這個 Feature，1 萬筆資料就要跑 8 個小時，團隊五個人分批跑，跑起來還是相當費時，所以隊長最後決定只跑三萬筆就好，也就是實價登入的前三萬筆拿來當作 Feature 的資料，最後經過團隊小組的討論 我們依序排除親屬間交易（1,166 筆）、農舍 / 倉庫 / 廠辦……（1,704 筆）、排除極值（602 筆）[例如價值過高的 Outlier 豪宅]，最後將資料刪減到只剩 26,528 筆。

再來介紹上述便利商店的資訊如何取得，其實是從成大提供的 ShapeFile 而來，ShapeFile 是一種儲存地理圖資的檔案格式，由數個檔案合成的：

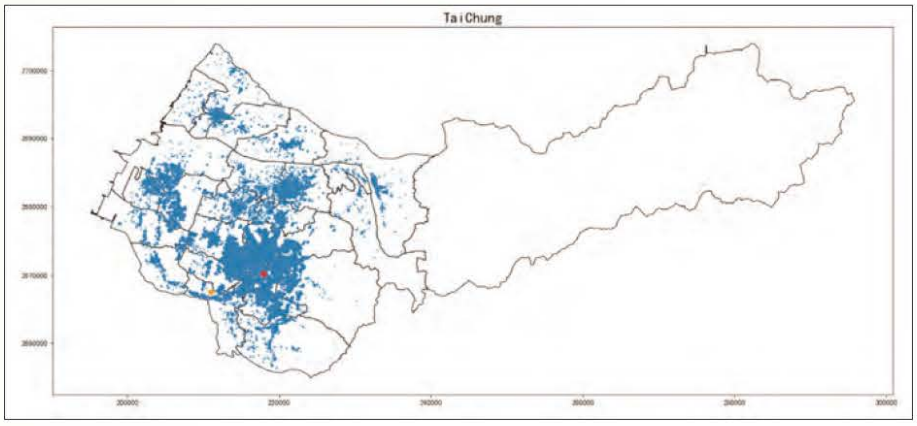
1. dbf（地理圖資的屬性資料）
2. shp（記錄地理圖資的點線面資訊）
3. shx（地理圖資的索引）

以下頁圖 10 － 1 為例，台中市週邊機能的點圖（圖中紅色為台鐵站位置，橘色為高鐵站的位置），經由上述 ShapeFile 資料我們可以計算出我們要用的 Features。

下頁圖 10 － 2 是台中市行政區的平均房價（單位：萬元 / 每平方公尺），可以看到西屯區跟南屯區最貴。

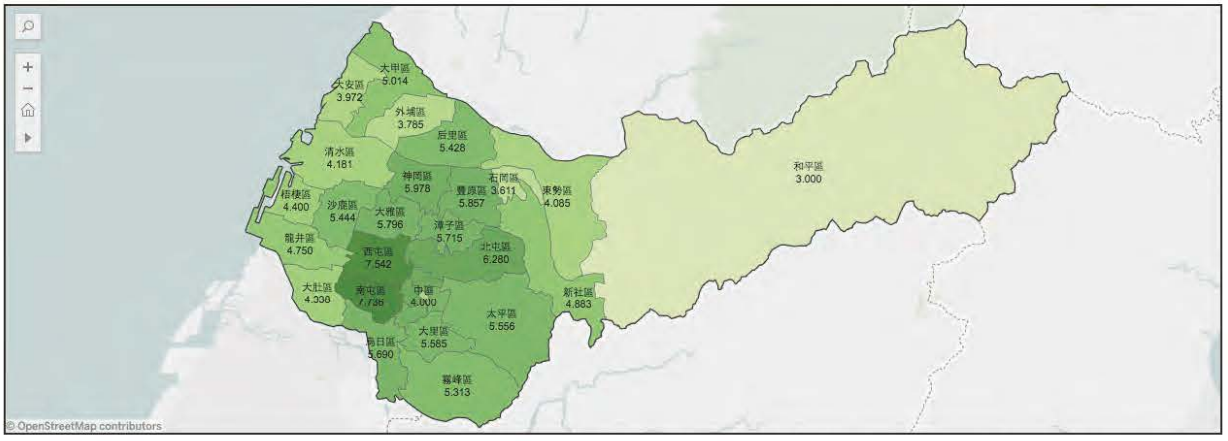
下頁圖 10 － 3 是台中七期跟五期重劃區對行政區房價的影響，由圖可見重劃區七期會拉高西屯跟南屯區整體房價。

圖 10 － 1



資料來源： 台灣人工智慧學校教學教材

圖 10 － 2

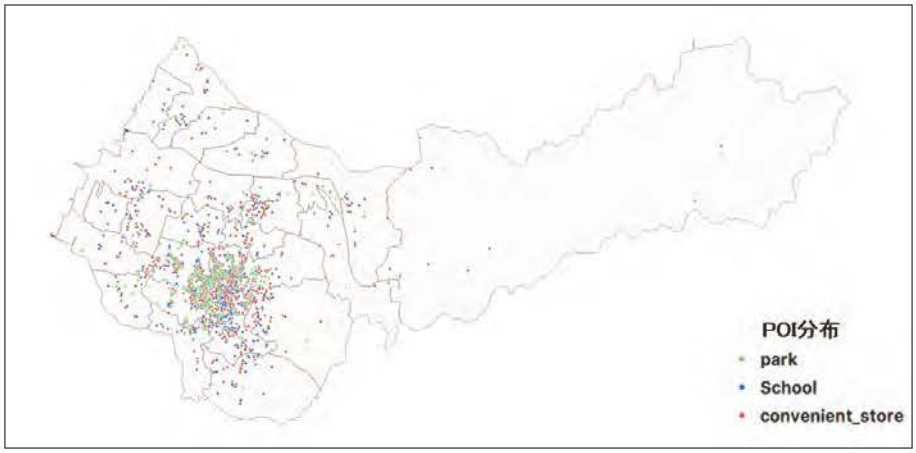


資料來源： 台灣人工智慧學校教學教材

圖 10 － 3

資料來源： 台灣人工智慧學校教學教材

圖 10 － 4



資料來源：台灣人工智慧學校教學教材

圖 10 － 4 顯示 POI（Point of Interest）就是跟生活機能有關的點如公園、學校、便利商店。

圖 10 － 5 為商業區與工業區的點分佈，由圖可見商業區比較分部在市中心，工業區分部在比較外圍。

最後團隊以圖 10 － 6 為基準做出 128 個 Features，並產生一個人工智慧 Model R2 值為 0.942（用 XGBoost 演算法），圖 10 － 7 為 Feature Importance 也就是特徵值重要性圖表。

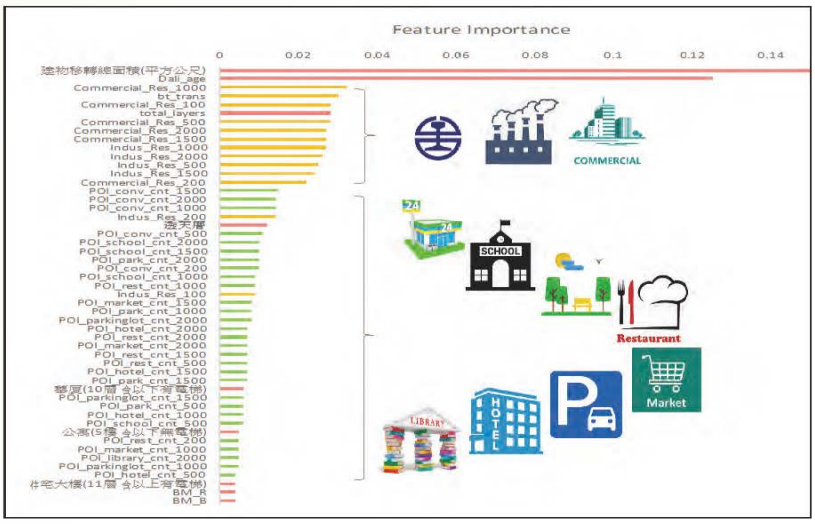
圖 10 － 5

資料來源：台灣人工智慧學校教學教材

圖 10 － 6

資料來源：台灣人工智慧學校教學教材

圖 10 － 7

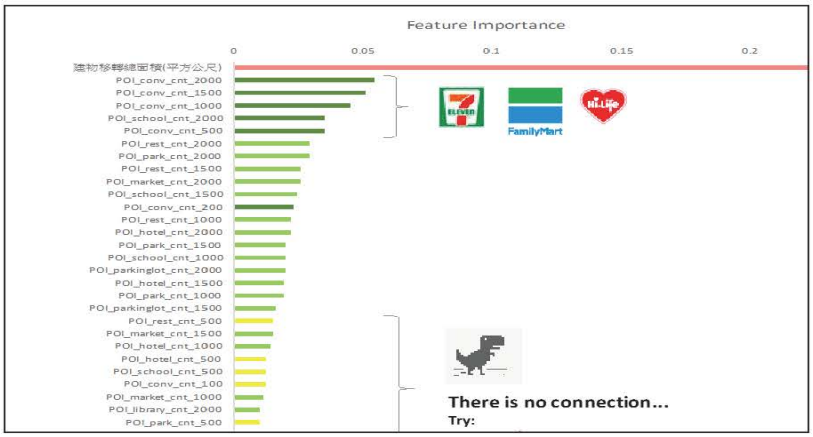


資料來源：台灣人工智慧學校教學教材

由圖 10 － 7 可見，會影響房價預測的最重要因素有兩個：房屋坪數和屋齡。接著就是台鐵離房子的距離，以及附近商業區工業區的涵蓋面積，最後的 Feature 才是附近的生活機能如加油站、公園、學校、超市、停車場等設施。

其實在做專題的過程中有先找助教跟成大討論過，成大說他們比較希望可以只看地裡因子，也就是 POI 生活機能，或是有關行政區對房價的預測能力，因此我們就依成大需求執行了一個只有地裡因子 Feature 的 Model（如圖 10 － 8）。最後看見比較重要的 Feature 只剩坪數與便利商店 2,000m/1,500m/1,000m 範圍內的店家數，然後 R2 值是 0.88（用 XGBoost 演算法），其他的 Feature Importance 值都很小不具代表性。

圖 10 － 8



資料來源：台灣人工智慧學校教學教材

最後解釋一下為什麼用總價來預測遠比用每坪單價來預測更為準確，我們後來想到的原因是單價有可能不準確，總價除以建物坪數與資料中的單價有不小的落差，所以才造成這樣的結果。

最後我們需要改進及建議的方向如圖 10 － 9：

圖 10 － 9

資料來源： 台灣人工智慧學校教學教材

結論

在接受三個月約 13 週的密集課程以後，獲益良多，覺得 AI 是門有趣的學問，但也是門大學問。三個月內學校教了相當大量的知識，有些知識原本可能是碩士班兩三年要學的東西，壓縮在三個月內教完，所以將台灣人工智慧學校定義為 AI 的軍校真是不為過。不過也因為東西多且複雜，我們學員都將老師與助教上課的投影片、影片或程式碼備份，當作未來複習之用。有些同學回到各自的工作崗位開始寫 AI 程式時，發覺兩個月前上課的東西已忘了，除了在學校 Line 群組提問之餘，也開始複習老師與助教之前的投影片，頓時有種學海無涯的感覺。終於可以理解為什麼在學校時，不管是在比賽或是上實習課，總是看到助教們在研讀最新的 AI 相關論文，因為 AI 相關知識是一直不斷地進步更新。幫我們上理論課的是大學教授或是上市公司 CEO，但是在實做上遇到問題時，在第一線給予我們協助的還是這些助教，我們真的打從心裡崇拜這些中研院的助教們。

此次學習相當豐富且精彩，雖然我的專題僅是幾萬筆資料的前處理與清理，與 Train 一個房價預測 Model，預測未來 5 年台中市房價，而這個 Model 目前雖只能達到輔助房產鑑價師的水準，不過也許將來有可能會取代他們。據我所知這種 AI 人工智慧的鑑價系統，某一同業三年前就開始研發，系統最近上線，上線前曾找台灣人工智慧學校做房價預測的三個團隊去為他們做簡報。上線系統[註15]我有去測試，特徵值只輸入屋齡、坪數、所在樓層及總樓層等 4 種特徵值，所以後來預測出來的房價雖然有在預期房價附近（我有去查我家地址實價登入的房價），但還是差了幾百萬，所以這樣的系統其實是不夠精準。目前看來這系統主要是讓民眾大約了解自己房子的約略價格，還不足

以取代鑑價師的工作。不過不準的原因其實應該是輸入的特徵值太少，之前我們 Train 的 Model 要輸入 128 種特徵值才可以達到精準預測，所以我想也許鑑價公司未來或許會需要 AI 人才協助粹取特徵值，那麼鑑價師的工作會不會因此被取代了呢？也許這是引進 AI 技術後會產生的另一個問題。

另外，中國大陸所發展的人工智慧已經可以考過他們國家醫科的高考了，這一點讓我相當震驚，因為這代表著在不久的將來，有可能是我跟機器人陳述我的病情（NLP 自然語言處理相關技術），它會自動判斷你是什麼病並開藥，到最後也許連醫生都會被完全取代。若真的是這樣，這個社會問題還蠻大，因為不少醫生都將會失業。

總之在未來的世代，AI 將會是一個相當重要的技能，因為之前去聽 SAP Taiwan Now 的年度盛會，有一位幫台積電建置大數據平臺的業界人士提到，以前都是學術界覺得 AI 會興起革命，這一次是產業界覺得 AI 興起會對這世界有重大影響。最後我要說的是，這三個月一口氣學習到太多新知識及技能，感覺還要花一段時間才逐一消化，真正提升自己的能力，融會貫通在自己的工作上。

參考資料

1. 微軟 ResNet 論文：https：//arxiv.org/pdf/1512.03385.pdf.
2. CNN ：https：//en.wikipedia.org/wiki/Convolutional_neural_network
3. 全國門牌地址定位服務 ：https：//www.tgos.tw/tgos/Web/Address/TGOS_Address.aspx
4. 內政部實價登入網站 ：http：//lvr.land.moi.gov.tw/homePage.action
5. XGBoost 論文 ：https：//arxiv.org/pdf/1603.02754.pdf
6. Backpropagation ：https：//mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/
7. Gradient Descent ：https：//blog.csdn.net/code_caq/article/details/79295335（建議參考台大電機李宏毅老師的 Youtube 影片會更懂）
8. nVidia 自駕車論文 ：
   a. https：//images.nvidia.com/content/tegra/automotive/images/2016/solutions/pdf/end-to-end-dl-using-px.pdf
   b. https：//devblogs.nvidia.com/deep-learning-self-driving-cars/
9. 台灣人工智慧學校 ：http：//aiacademy.tw/
10. R square（R2） ：https：//zh.wikipedia.org/wiki/%E5%86%B3%E5%AE%9A%E7%B3%BB%E6%95%B0
11. AUC ：https：//zh.wikipedia.org/wiki/ROC%E6%9B%B2%E7%BA%BF

12. RMSE ：https：//zh.wikipedia.org/wiki/%E5%9D%87%E6%96%B9%E6%A0%B9%E8%AF%AF%E5%B7%AE

13. TWD97 ： http：//www.sunriver.com.tw/grid_tm2.htm

14. Kaggle 競賽 ： https：//www.kaggle.com/

15. 中信房價試算：https：//ctbc-mortgage.com/tool/cal12.html

16. TensorFlow ： https：//www.TensorFlow.org/

17. K-Means Clustering ：https：//zh.wikipedia.org/wiki/K-%E5%B9%B3%E5%9D%87%E7%AE%97%E6%B3%95

18. RandomForest ： https：//en.wikipedia.org/wiki/Random_forest

19. 缺失值處理 ： https：//rpubs.com/skydome20/R-Note10-Missing_Value

－ 註 釋 －

1. K-means Clustering 演算法：此演算法會自動將母體資料自動分成 K 類，K 一般來設定為 1 or 3 or 4，設成 3 就會自動分成 3 群。

2. 惡意程式辨識 https：//tbrain.trendmicro.com.tw/Competitions/Details/1

3. 關於 Keras 與 TensorFlow 的文章提供給大家參考： https：//www.Hksilicon.com/articles/1277383 TensorFlow

4. 趨勢科技 Kaggle 競賽 https：//tbrain.trendmicro.com.tw/Competitions/Details/1

5. TF-IDF 是將一段文章比較重要的詞擷取出來藉以判斷這篇文章是怎麼樣的文章。

6. Atari 遊戲影片連結 https：//www.youtube.com/watch?v=V1eYniJ0Rnk

7. 3D 世界虛擬開車的增強式學習影片 https：//www.youtube.com/watch?v=pqGSublT02w

8. https：//web.stanford.edu/~anayebi/projects/CS_239_Final_Project_Writeup.pdf

9. 論文網址：https：//pjreddie.com/media/files/papers/yolo.pdf

10. YOLO 經典測試影片 https：//www.youtube.com/watch?v=VOC3huqHrss

11. 自駕車可以用 YOLO 網路影片 https：//www.youtube.com/watch?v=OksuVuNY5o0

12. nVidia 展示自家自駕車系統影片 https：//www.youtube.com/watch?v=0rc4RqYLtEU

13. 少女漫畫生成器 https：//make.girls.moe/#/

14. 換臉影片 https：//www.youtube.com/watch?v=jI6H-0YWkSc

15. 上線系統連結：https：//ctbc-mortgage.com/tool/cal12.html