|  |  |
| --- | --- |
| 大專新進人員訓練工作報告  　姓　　名： 王翔禾 單位：　台化工務部自控處  　報告類別：□輪班訓練　 ■基層實務工作訓練第 (二) 次報告  　訓練部門：台化工務部自控處智能專案組  　起迄日期：2022/01/01~2022/03/14    工作項目：太陽能案場發電效率統計自動化及ARO3廠去庚烷塔AI節  機介面建置  　報告項目：太陽能案場發電效率統計自動化及 | |
| 評 核 ( 評 語 ) | |
| (２) 一 級 主 管 | (１) 二 級 主 管 |
|  |  |
| (４) 經 營 主 管 | (３) 經 理 室 |
|  |  |

表號：P0002703 規格：A4

註：一.報告內容應包括訓練期間之 1.工作心得感想 2.所得之工作觀

念及精神 3.自我檢討 4.建議意見。

二.訓練部門主管應詳細評核「訓練工作報告」內容，並批註意見。

**目錄**

[一、 前言 1](#_Toc121238253)

[二、 節能優化AI人機介面 3](#_Toc121238254)

[2-1 系統架構 3](#_Toc121238255)

[2-2 Django 介紹 5](#_Toc121238256)

[2-3 PostgreSQL介紹 8](#_Toc121238257)

[2-4 Apache架設 10](#_Toc121238258)

[2-5 系統開發流程 12](#_Toc121238259)

[三、 AI機器學習介紹 15](#_Toc121238260)

[3-1 XGboost Model介紹 15](#_Toc121238261)

[3-2 Bagging 與 Boosting 比較 16](#_Toc121238262)

[3-3 XGBoost 優缺點比較 18](#_Toc121238263)

[3-4 C914 XGBoost AI Model 介紹 18](#_Toc121238264)

[四、 AI深度學習介紹 21](#_Toc121238265)

[4-1 AI神經網路介紹 21](#_Toc121238266)

[4-2 AI神經網路訓練 23](#_Toc121238267)

[4-2 ARO3\_3C8201 AI神經網路模型介紹 24](#_Toc121238268)

[五、 工作心得 25](#_Toc121238269)

前言

職目前承辦工作為「太陽能案場發電效率統計自動化」及「陽能案場發電效率統計分析報表自動化」。

「太陽能案場發電效率統計自動化」主要協助經理室專員將每日手動抓取太陽能案場資訊之程序自動化，每日共需抓取新港紡紗、紡紗南側、紡紗北側、複材南/北側、公用保養廠及苛化房、麥寮PTA及AROMA廠、寧波煤場等共17案場，抓取內容為當日發電量(kWh)、容因%、PR值、模組溫度等，每日彙總於Excel報表中，提供業主做營運分析與管理。

「太陽能案場發電效率統計分析報表自動化」主要將第一案所蒐集之資料，依照不同地區進行統計分析，目前分成四大區域分別為 新港、寧波、麥寮、福懋各區，每大區域可再細分成個小區域並依照當月累計「容因%」進行排序，可讓使用者一眼看出各案場發電效率優劣

此兩案旨在協助人員將每日所需進行的重複動作，以自動化程式替代人工操作，可謂人員節省大量時間，提升工作效率。

本季工作心得，以上述二案為主軸，詳如後續說明。

1. 太陽能案場發電效率統計自動化

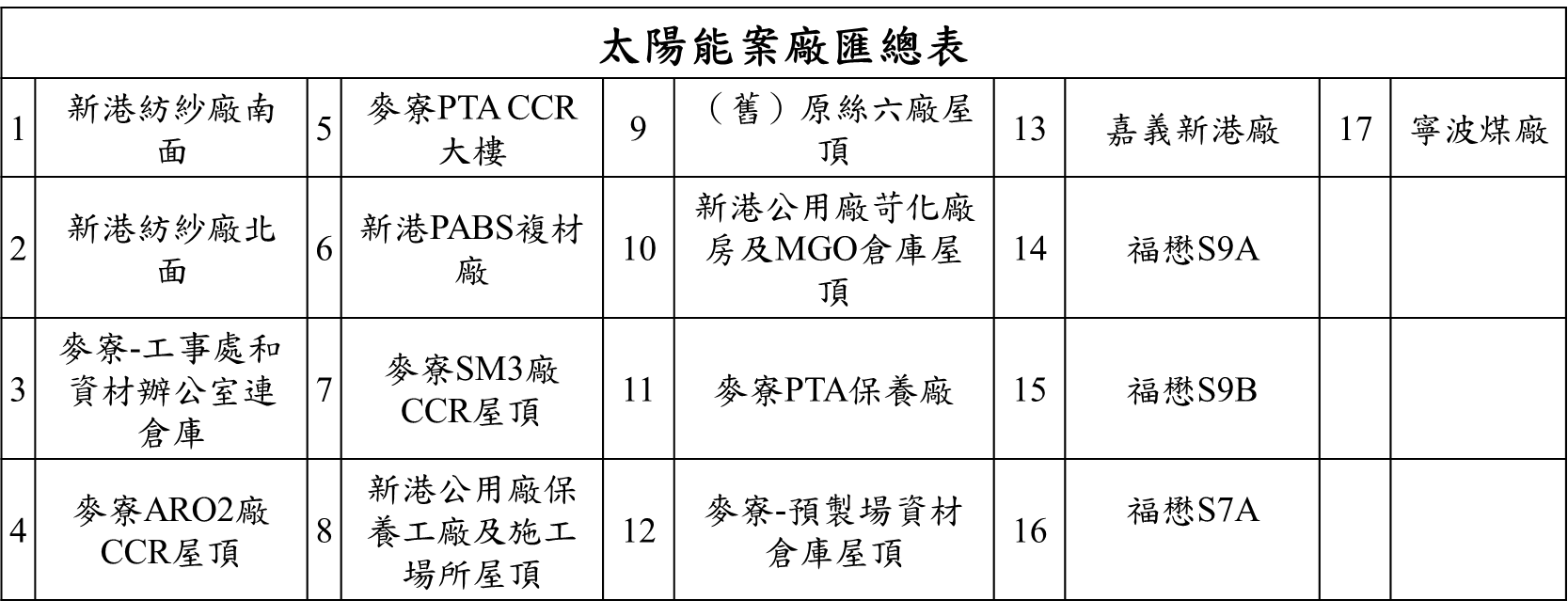
本案主要以爬蟲及API串接形式蒐集各太陽能電廠資料，於自動化改善前，使用者須每日於寧波、福懋、台化嘉義新港紡紗、盛齊綠電等四個不同網站平台共17案廠每日抄寫所需資料，資料蒐集過程繁瑣複雜，因此本案撰寫自動化程式每日自動抓取上述各案廠資料，經計算後每日自動產出報表，以利業主及時獲取所需資訊並減輕人工負擔，案廠列表如下表2-1所示。

表2-1太陽能案廠匯總表

2-1 系統架構

以下為本案工程開發時所用到的作業系統、使用的程式語言，如表2-2所示。系統架構圖如圖2-1所示，使用Python語言之第三方套件「Requests」，配合測試工具 Postman開發網頁爬蟲及API串接程式，接續以Python語言設計將下載之數據配合第三方套件「Openpyxl」匯總於每月Excel報表中，以供使用者查看。

|  |  |
| --- | --- |
| 作業系統 | Windows 10 |
| 程式語言、套件 | Python、Requests、Openpyxl |
| 測試工具 | Postman |

表2-1、系統環境

圖2-1、系統架構圖

2-2 Python網路爬蟲介紹

網路爬蟲 ( spider 或 web crawler )，是一種可以自動瀏覽全球資訊網的網路機器人，許多的搜尋入口網站 ( 例如 Google )，都會透過網路爬蟲收集網路上的各種資訊，進一步分析後成為使用者搜尋的資料，許多開發者也會自行開發不同的爬蟲程式，進行大數據收集與分析的動作。

而網頁爬蟲又可大致分為兩種分別為靜態網頁爬蟲及動態網頁爬蟲，靜態網站是指網站完成一個請求 ( request ) 與回應 ( response ) 後，用戶端即不再與伺服器有任何的交流，所有的互動都只與瀏覽器的網頁互動，資訊不會傳遞到後端伺服器，動態網站是指網站會依照使用者的行為不斷的與伺服器進行交流，常見的動態爬蟲需要進行「登入」的動作，以獲取使用者認證資訊。

通常動態網站爬蟲實作比較複雜，撰寫爬蟲程式時必須要知道網站需要使用者提供那些資訊，提供正確的資訊，才能取得所需要的資料 而靜態網站爬蟲比較容易實作，只要爬蟲已經閱讀完整份網頁，就可以取得這個網頁所有的資訊進行分析。

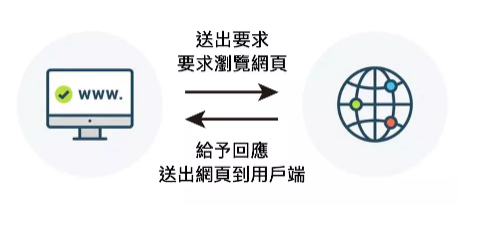
圖2-2、Python靜態網路爬蟲示意圖

圖2-3、Python動態網路爬蟲示意圖

本案所所撰寫的爬蟲程式簡單來說是一段可以自動化抓取網頁內容的程式，以瀏覽網頁為例，平常我們瀏覽網站，就是對遠方的伺服器提出一個請求(Request)，這時候目標伺服器就會收到資訊，並且回傳(Response)給你前端html, css, javascript等程式資訊，接著再由瀏覽器會把這些檔案組成渲染成所呈現之網頁，同樣的爬蟲原理就是，藉由你的爬蟲程式去進行Request，等待程式接收到Response後，解析出需要的資訊，進行更進一步應用。

簡易的架構如下，以下以本案為例，藉由初始網址(URL)，驗證登錄資訊(目標伺服器收到請求中夾帶的資訊例如帳號、密碼等，經伺服器確認使用者身份後會返回一組TOKEN或建立SESSION連線以作為使用者後續操作時的身分判斷)，接著發對目標網址發出Request請求，等待目標伺服器回應後，藉由網頁解析器獲取解析(網頁解析器能將伺服器傳送回來之原始碼轉換成能夠被理解閱讀之程式代碼，常用的有Python自带的html.parser、或Python 第三方套件beautifulsoup、Xpath等)，最後再寫入Excel報表中，其流程如下圖2-4 、圖2-5所示。

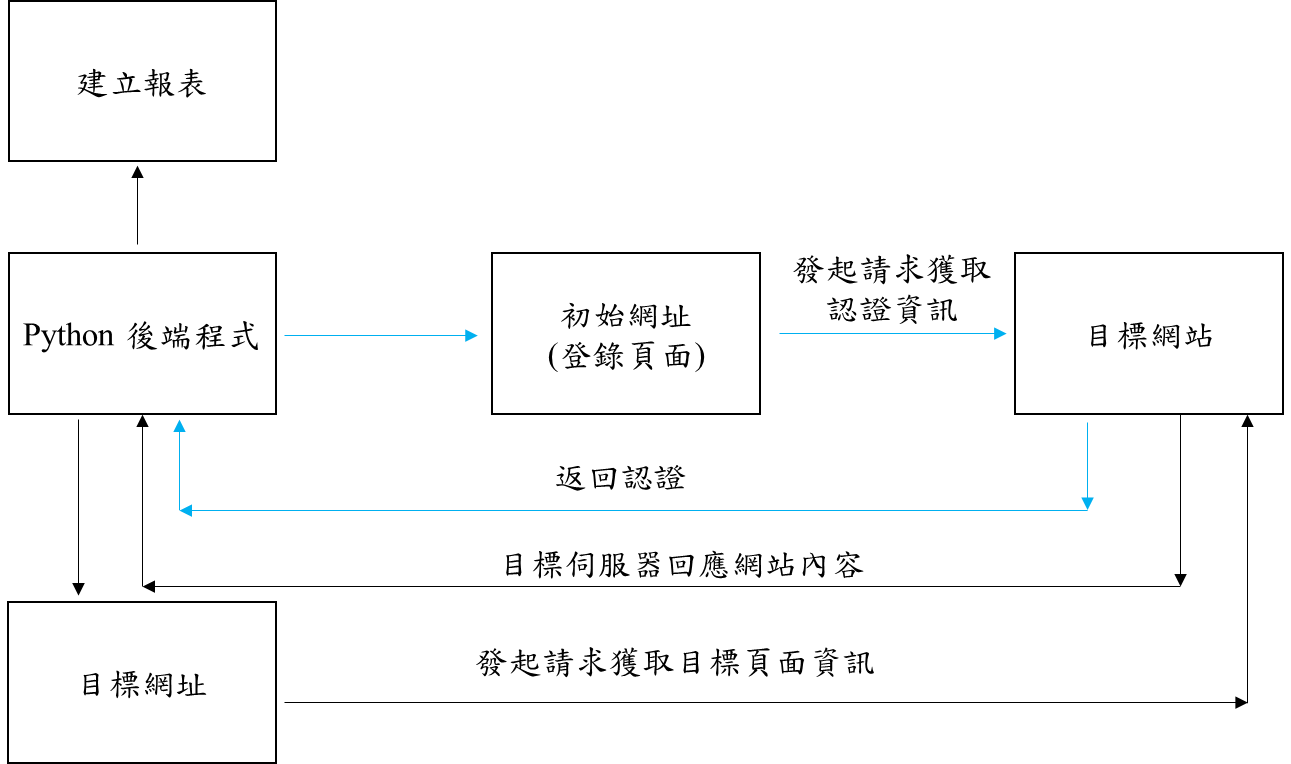
2-4、Python網路爬蟲流程圖



圖2-5、Python網路爬蟲程式示意圖(驗證登錄資訊)

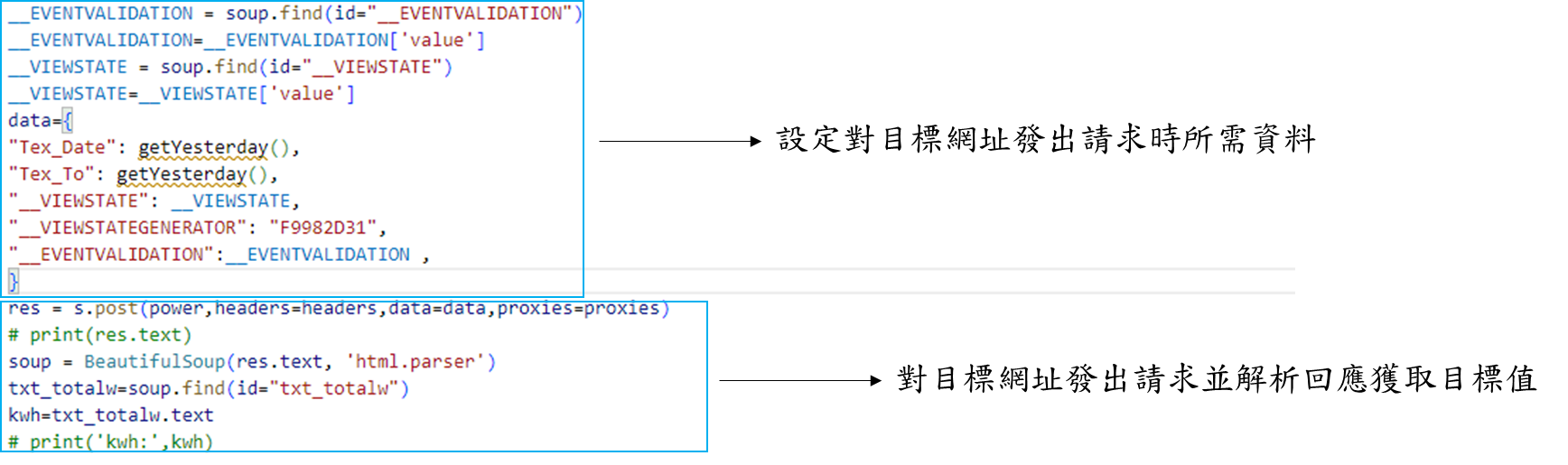


圖2-6、Python網路爬蟲程式示意圖(獲取目標值)

2-3 Python API串接介紹

應用程式介面（Application Programming Interface），縮寫為API，是一種計算介面，它定義多個軟體中介之間的互動，以及可以進行的呼叫（call）或請求（request）的種類，如何進行呼叫或發出請求，應使用的資料格式，應遵循的慣例等，API的工作是讓AB兩端資料拋接，一端提供資料輸入、另一端依據資料回傳結果，其目的在於「不必了解對方的技術與邏輯即可加速開發的共識」。

　簡單來說，API的功用就是減輕開發者的工作的一項技術，以AB兩端的資料拋接，一端提供資料輸入、另一端依據資料回傳結果，讓兩方技術人員「不必了解對方的技術與邏輯」，即可加速開發的共識。

在產品開發或者網頁開發中，資料API介面就像後端遞給前端的數據線。後端人員已經整理好想要展示的資料，只需要通過這根資料線傳輸給前端的開發人員，前端就可將其按需求進行後續的視覺化，而且這個介面還可供外界使用，如同本案以API串接盛齊綠電網站並獲取所需資訊。

與網頁爬蟲不同的是，API的設計更為簡單高效，此介面已經存儲好使用者所需要的資料，使用者不需要再花費額外的時間去解析網頁。

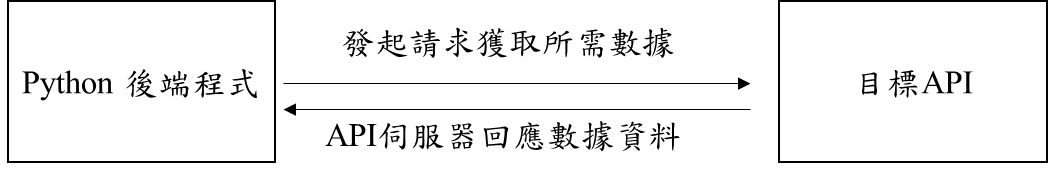


圖2-6、Python 與API交互關係圖

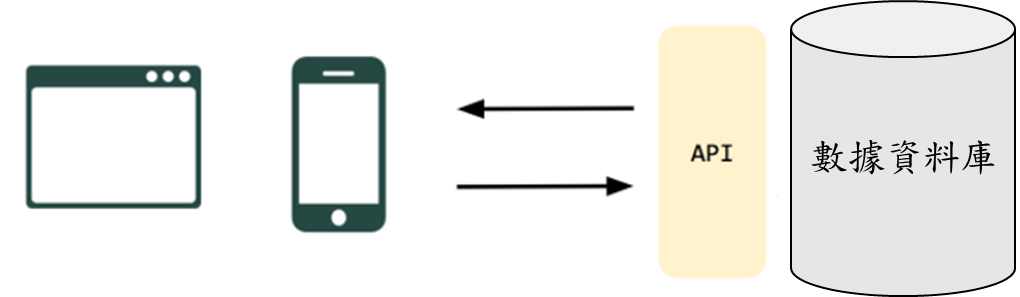
在移動設備當道的時代趨勢中，API 也日益重要。一個軟體服務可能會在多種設備上被人使用，例如手機、瀏覽器、穿戴式設備等，這些裝置必須要和同一個後端資料庫互動，才可能呈現出一致的內容。

圖2-7、移動/固定裝置與API交互關係圖

2-4 HyperText Transfer Protocol 傳輸協定介紹

在前面章節中均有提到網頁的請求(Request)及回應(Response)，其為 HyperText Transfer Protocol (HTTP) 傳輸協定的組合結構，HTTP是一種用來傳輸超媒體文件 (例如HTML文件) 的應用層協定，被設計來讓瀏覽器和伺服器進行溝通。

一般來說傳輸資料的兩端會分為 客戶端 ( Client ) 跟 伺服器端 ( Server )，Client端以網頁來說就是你的瀏覽器、電腦，主要會發送 「請求 request」到 Server 端而Server端收到request後開始處理資料，完成後會回傳「回應 response」到 Client端，其組成結構如圖2-8所示，而網頁所呈現的資訊，其實就是眾多的 request 跟 response交互而成，其步驟可拆解為下列幾點:

1. 瀏覽器發送 HTTP request 到 Server
2. Server 回傳 response 到瀏覽器
3. 瀏覽器進行解析 html、css、js、圖片檔案…等等，渲染成可讀性高的網頁內容



圖2-8、HTTP 的組成結構

從客戶端發出 HTTP Request 時，通常會定義以下的資訊，以瀏覽 Google 首頁為例，如圖2-9所示。

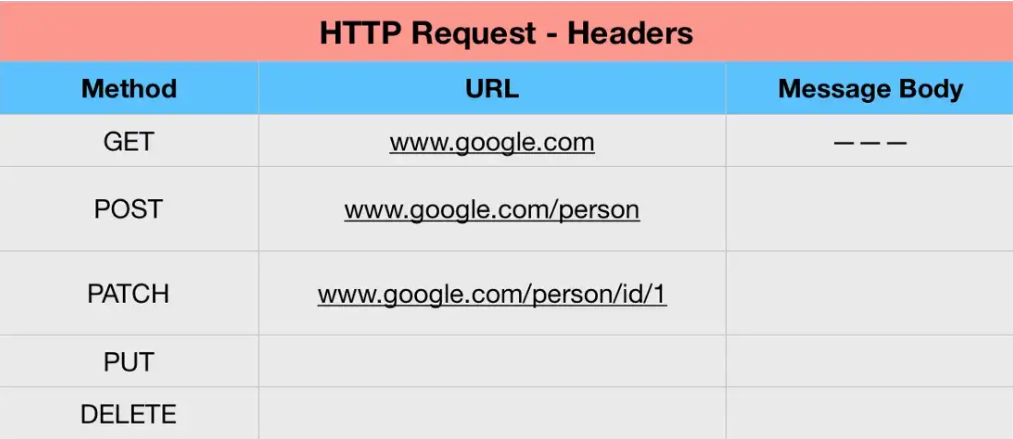


圖2-8、HTTP Request 所定義的資訊內容

透過日常的郵政系統來比喻，假如我們今天要寄出一封信件，若我們選擇 GET 掛號的寄件方式，而地址URL是 Google首頁（www.google.com），Message Body 則會是信件內容。上述的內容代表我們要對 www.google.com 做一個 GET的動作，才得以進入到Google首頁，而常見的 Method 方法如下：

* GET：讀取資料
* POST：新增資料（常搭配 form 標籤使用）
* PATCH：修改資料
* PUT：修改資料（若資料不存在，則新增資料）
* DELETE：刪除資料

通常一個 Method 會搭配一個 URL，也會對應伺服器端一組特定的資源，而 Message Body 的內容則取決於每次的動作。若單純是使用 GET 方法瀏覽網頁，則 Message Body 為空，但是，若以填寫表單為範例，那客戶端就會送出資料，這筆資料就會被傳送到伺服器的資料庫，此時，我們則會使用 POST 方法將資料送進 Message Body，提交給某個指定的 URL，進而建立資料或更新資料。

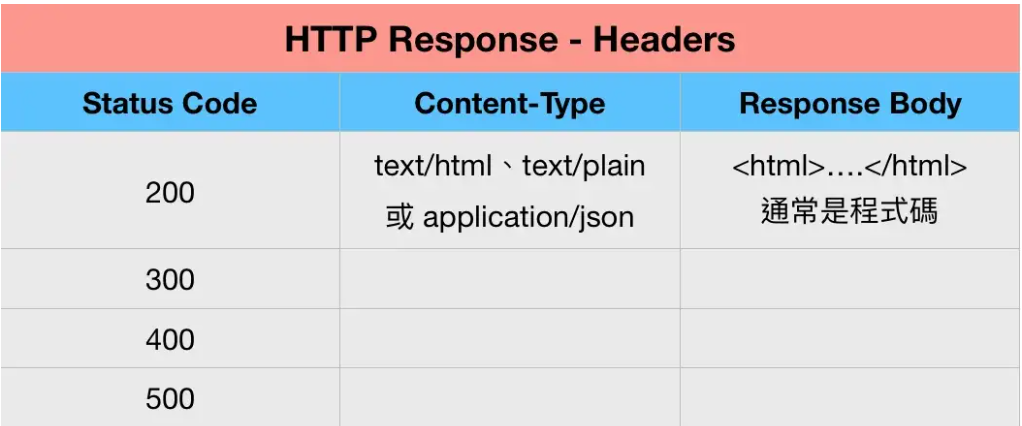
從伺服器端回傳 HTTP Response 時，通常會定義以下的資訊，以瀏覽 Google 首頁為例，如圖2-9所示。

圖2-9、HTTP Response 所定義的資訊內容

狀態碼 (status code)是一種狀態碼的呈現，也是客戶端與伺服器端溝通的狀態情況，常撿的狀態代碼如下。

* 1XX 訊息類 (收到請求，請求者繼續執行操作)
* 2XX 成功類 (操作被成功接受並處理)，例如：200 成功回應
* 3XX 重定向類 (需進一步操作才能完成)，例如：301 成功轉向
* 4XX 客戶端錯誤類 (請求語法錯誤或無法完成請求)，例如：404 找不到資源
* 5XX 伺服器錯誤類 (後端的問題)，例如：500 伺服器錯誤

content-type 這個項目，它則會定義回應格式，例如 text/html、text/plain 或 application/json 等等，客戶端才會知道該如何打開訊息，而 Response Body 通常會是一個一個的程式碼檔案，最終藉由瀏覽器渲染畫面並呈現於使用者。

2-5 Python Openpyxl套件介紹

在開發Python網頁爬蟲或API串接時，除了可以將取得的資料儲存在資料庫外，也能匯出成Excel檔案製作成報表，進而使用其中的圖形化功能，來製作統計圖表，有效率的進行資料分析。

Openpyxl是一個開源專案，其是一個讀寫Excel 文檔的Python庫，其不僅能夠同時讀取和修改Excel文檔，而且可以對Excel檔內儲存格進行詳細設置，包括儲存格樣式等內容，甚至還支援圖表插入、列印設置等內容，使用openpyxl可以讀寫xltm, xltx, xlsm, xlsx等類型的檔，且可以處理資料量較大的Excel檔，跨平臺處理大量資料是其它模組沒法相比的，因此，openpyxl成為處理Excel複雜問題的首選庫函數。

而本案則利用此套件將每日搜集的資料彙總於每月Excel報表中，供使用者查看，如圖2-10、圖2-11所示。

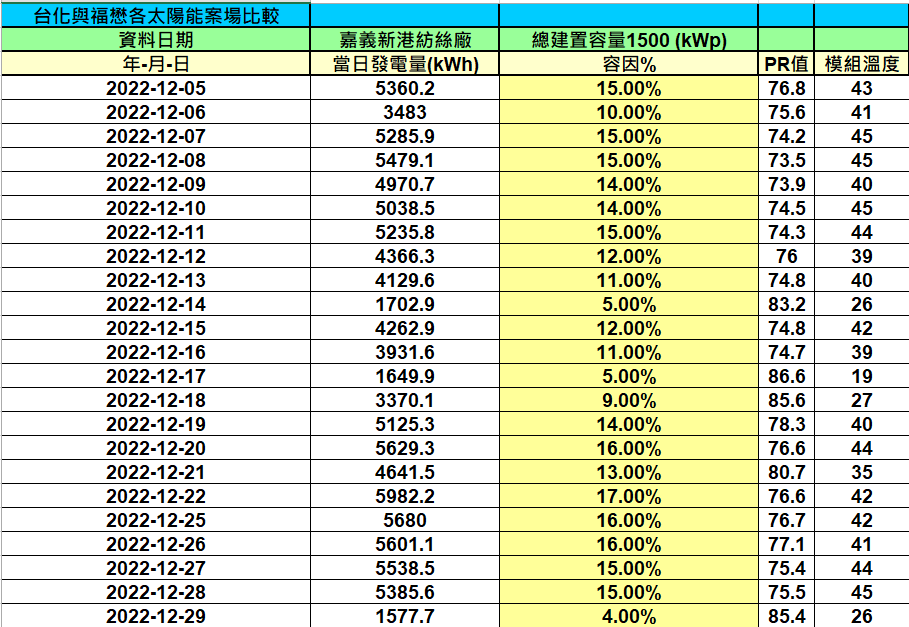
圖2-10、Excel 每月匯總報表示意圖

圖2-12、定時資料抓取式畫面

AI機器學習介紹

3-1 XGboost Model介紹

C914萃取油分離塔節能優化AI Model利用經典的 XGboost 演算法進行回歸數值預測，XGboost 全名為 eXtreme Gradient Boosting，是目前競賽中最常見到的算法，同時也是多數得獎者所使用的模型。此機器學習模型是由華盛頓大學博士生陳天奇所提出來的，它是以 Boosting 為基礎下去實作，並結合 Bagging的優點作為實現，Bagging及Boosting將於3-2小節詳述。

XGboost 保有 Gradient Boosting 的做法，每一棵樹是互相關聯的，目標是希望後面生成的樹能夠修正前面一棵樹判斷錯誤的地方。此外 XGboost 是採用特徵隨機採樣的技巧，和隨機森林一樣在生成每一棵樹的時候隨機抽取特徵，因此在每棵樹的生成中並不會每一次都拿全部的特徵參與決策。

此外為了讓模型不過於複雜，XGboost 在目標函數添加了標準項正則化。因為模型在訓練時為了擬合訓練資料，會產生很多高次項的函數，但反而容易被雜訊干擾導致過度擬合。因此 L1/L2正則化之目的是讓損失函數更佳平滑，且抗雜訊干擾能力更大。最後 XGboost 還用到了一階導數和二階導數來生成下一棵樹。其中 Gradient 就是所謂的一階導數，而 Hessian 即為二階導數。

3-2 Bagging 與 Boosting 比較

在機器學習中須了解Bagging 與 Boosting 兩者間的差異， Bagging 透過隨機抽樣的方式生成每一棵樹，最重要的是每棵樹彼此獨立並無關聯。上一小節提到的隨機森林就是 Bagging 的實例。另外 Boosting 則是透過序列的方式生成樹，後面所生成的樹會與前一棵樹相關。本章所提及的 XGBoost 就是 Boosting 方法的其中一種實例。正是每棵樹的生成都改善了上一棵樹學習不好的地方，因此 Boosting 的模型通常會比 Bagging 還來的精準，而決策樹通常為一棵複雜的樹，而在 Boosting 是產生非常多棵的樹，Boosting 希望新的樹可以針對舊的樹預測不太好的部分做改善。最終再把所有樹的值合在一起才能當最後的預測輸出。

1. Bagging 透過抽樣的方式生成樹，每棵樹彼此獨立
2. Boosting 透過序列的方式生成樹，後面生成的樹會

與前一棵樹相關

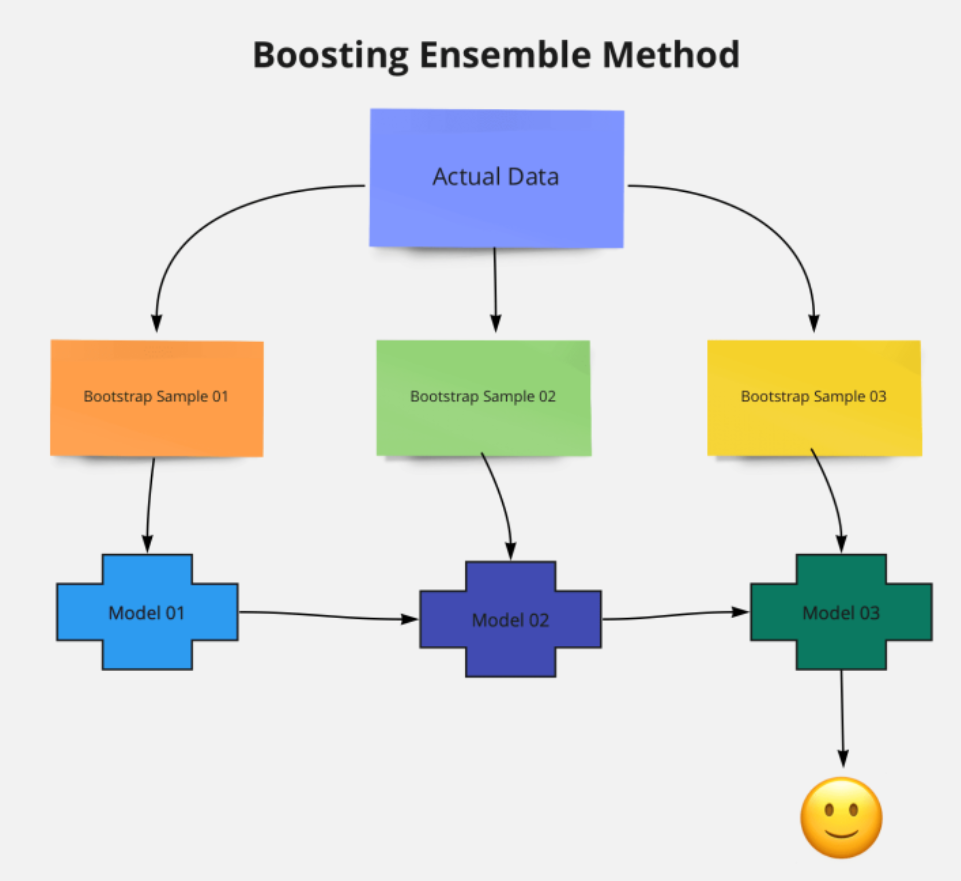
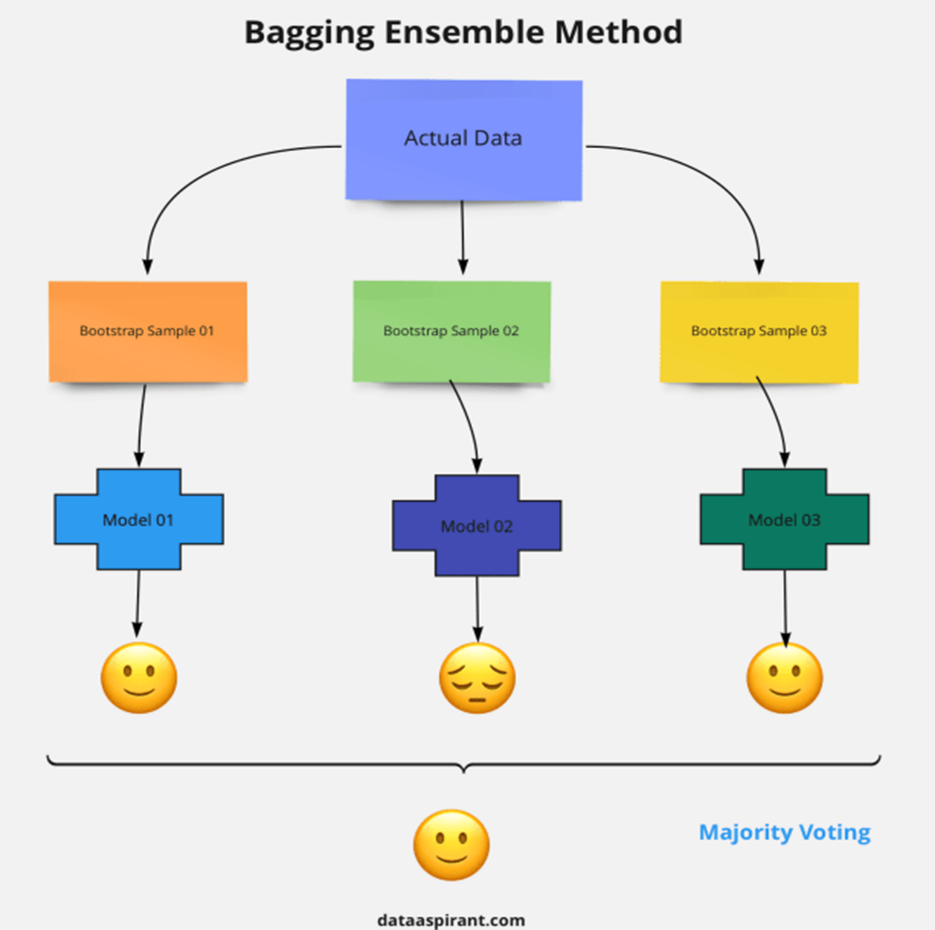


圖3-1、Bagging 與 Boosting 比較示意圖

3-3 XGBoost 優缺點比較

XGBoost 除了可以做分類也能進行迴歸連續性數值的預測，而且效果通常都不差。並透過 Boosting 技巧將許多弱決策樹集成在一起形成一個強的預測模型，而其優缺點比較如下。

1.優點

* + - * 利用了二階梯度來對節點進行劃分
      * 利用局部近似算法對分裂節點進行優化
      * 在損失函數中加入了 L1/L2 項，控制模型的複雜度
      * 提供 GPU 平行化運算

2.缺點

* + - * 預排序過程的空間複雜度過高，不僅需要存儲特徵值，還需要存儲特徵對應樣本的梯度統計值的索引，等會消耗較端記憶體

3-4 C914 XGBoost AI Model 介紹

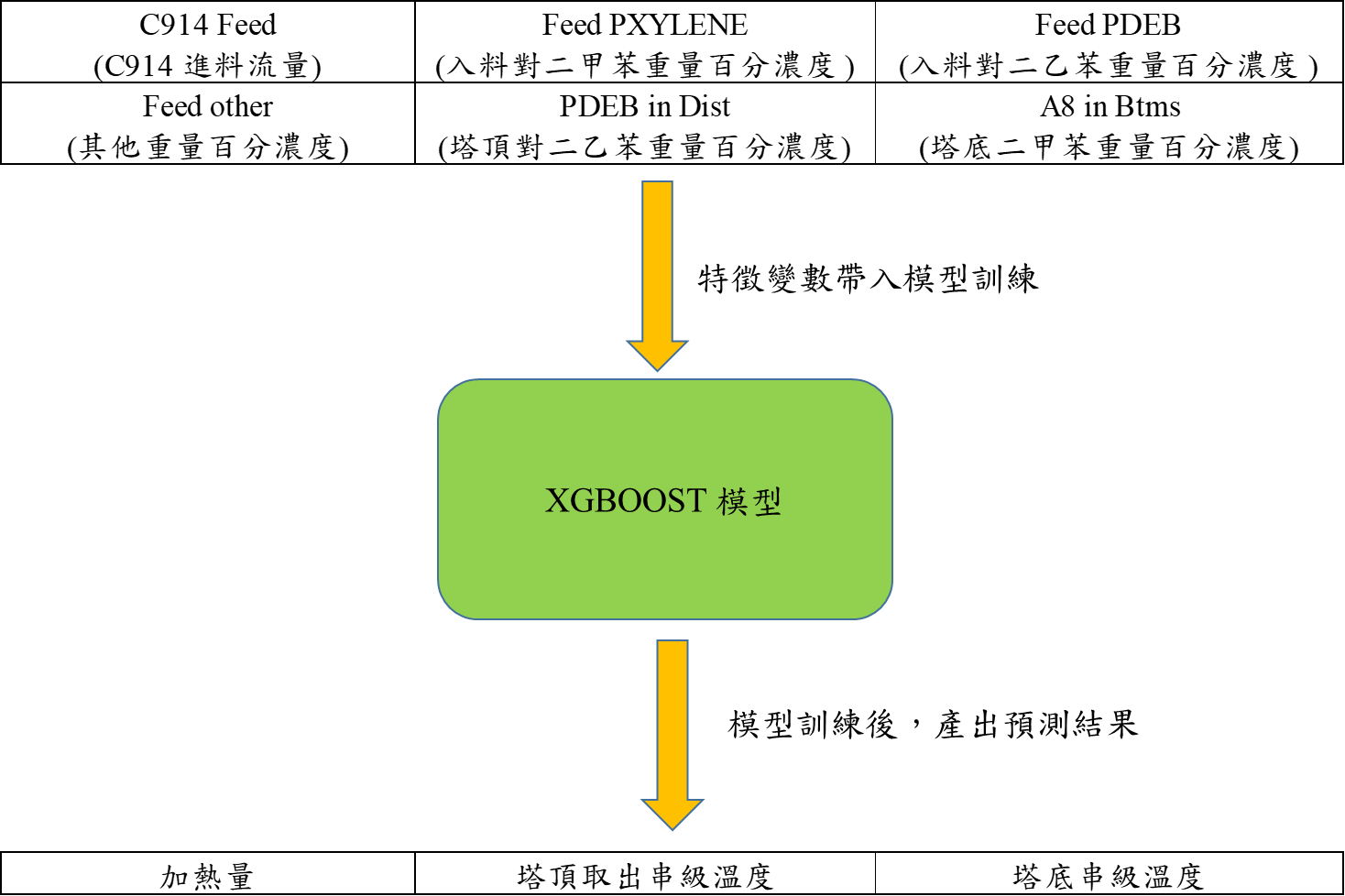
C914 AI mode以經過篩選後之六個特徵變數帶入XGBoost模型中進行深度學習運算，並預測出加熱量、塔頂取出串級溫度、塔底串級溫度等三個目標值，其流程圖如下圖 3-2所示，其中塔頂對二乙苯及塔底對二甲苯濃度為C914萃取塔之管制標的，製程人員可依模型預測得到特定標濃度下之製程操作值。

圖3-2、C914 AI mode 流程示意圖

接著進一步將化一部所提供之訓練資料共3550筆，劃分訓練集及測試集比例為8:2，使用化一部建模時使用之六個特徵變數分別帶入XGBoost 及現場常用之Linear-regression兩種model訓練，並以 R-Square、MSE、MAPE等三種指標，判斷其訓練結果，從圖3-3可看出在訓練相同資料的情況下XGBoost之訓練結果三種指標均優於Linear-regression，且 R-Square 可達到0.99，足以見到XGBoost之預測模型預測結果較Linear-regression好。

圖3-3、XGBoost與Linear-Regression結果比較

於3-2小節有提到XGBoost訓練過程中會產生多棵的決策樹，新的樹可以針對舊的樹預測不足的部分做改善，每棵決策樹將影響下一棵決策樹之複雜度，藉以改善上一輪訓練之缺陷，持續優化模型，也是藉由此方式讓XGBoost仍舊在訓練樣本有限、訓練時間短、等場景下具有獨特的優勢。相比深度神經網路，XGBoost能夠更好地處理表格資料，並具有更強的可解釋性，另外具有易於調整參數、輸入資料不變性等優勢。

AI深度學習介紹

4-1 AI神經網路介紹

ARO3\_3C8201去庚烷塔AI Model利用近年崛起的人工神經網路(Artificial Neural Network)進行回歸數值預測，人工神經網路是一個資訊處理系統，它是由許多人工神經元 (artificial neuron) 所組成，模擬人腦神經的架構及原理來解決一些決策判定上的問題，也可以理解為深度學習(Deep Learning)下的一個分支。

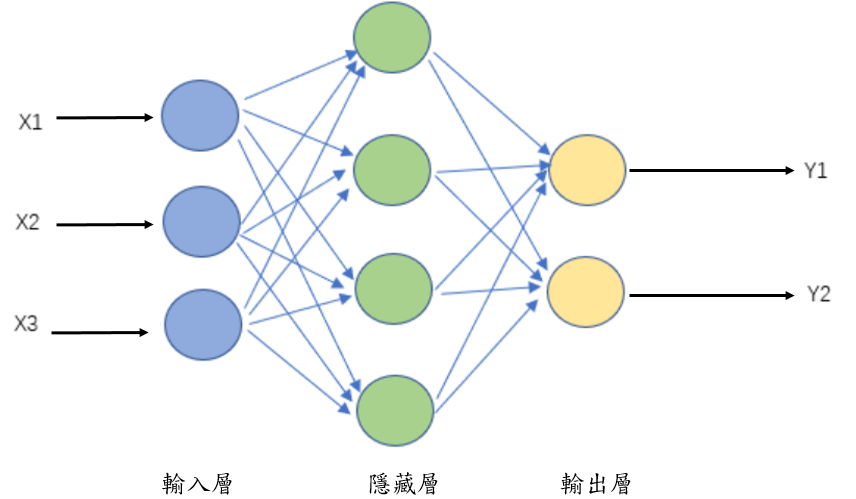
一個AI mode至少包含三種不同的層，輸入層(input layer)、隱藏層(hidden layer)及輸出層(output layer)。模型可以擁有多個隱藏層。每層可以包含一個或多個neuron，每個neuron個別得到來自上一層的輸入後計算並產生輸出，最後會由輸出層產生最終的輸出，如圖3-4 所示。

圖4-1 神經網路模型基本架構圖

1. 輸入層

* + - * 在神經網路的第一層被稱作輸入層，這層取得在外部數據，傳至這層的輸入就是常說的特徵值。
      * 這層neurons的數量和傳入特徵值的數量相同。

2. 隱藏層

* + - * 在輸入層與輸出層之間的層面被稱作隱藏層，一個神經網路至少要有一層隱藏層，這裡就是學習開始的地方。隱藏層的neuron為了學習需求而做運算，模型需自行決定需要的層數。當隱藏層數越多，計算的複雜度與計算時間也會隨之增加。

2. 輸出層

* + - * 在神經網路的最後一層就是輸出層。 輸出層會得到在最後一個隱藏層的輸出。輸出層的neurons數量由使用者想要神經網路輸出幾組值決定。
      * 回歸問題，當網路需要預測一串連續的數值，像是溫度預測，這樣的輸出層只會有一個neuron。
      * 二元/多元問題，當網路需要預測多個中其中一個類別(one of many classes)，輸出層的neurons會有和所有可能類別(class)的相同數量。假如網路被訓練用來預測四種之中某一種可能的動物，貓、狗、獅或牛，那麼輸出層就會有四個neurons，一個neuron代表一個class。

4-2 AI神經網路訓練

上圖4-1中每一條箭頭，均代表神經元間相對應之「權重」神經網路的運作可以簡單理解成，神經元輸入值乘以權重並進入隱藏層運算，最後輸出預測值，在這個過程中權重是一個很重要的關鍵，他決定了每個特徵的重要性，所以一個好的神經網路，必須要有完善的「計算權重」的環節。

權重的計算無法一步到位，要反覆訓練、檢驗誤差並且修正，才能得到一個好的網路，因此，計算權重的流程如下：

1.隨機初始化權重，使得權重無限接近 0 但不等於 0

2.將第一組觀察數據輸入"輸入層"，每個變量特徵輸入對

應神經元

3.正向傳播：神經網路從左至右進行計算，每個神經元得

到來自上一層神經元的輸入與權重的運算，並計算預

測值和實際值的差異

4.反向傳播：神經網路從右至左進行計算，依據誤差函數

相對於權重的梯度，對每個權重進行更新，過程中學習

速率與梯度將決定更新的速度。

5.當所有數據都輸入神經網路後，稱之為一期 (epoch) 的

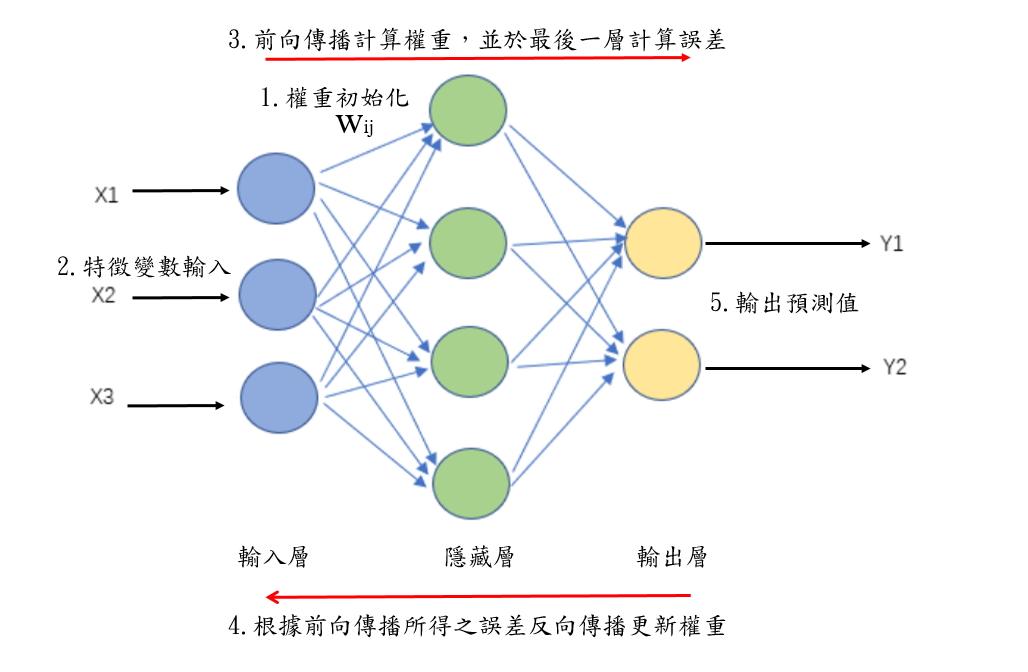
****訓練執行若干個 epoch（次數需自己設定），並對每一 組新的觀察數據，重複步驟1-4 (batch learning)持續訓練模型，直到最後一個 epoch，模型即訓練完成。

圖4-2 神經網路模型訓練流程

4-2 ARO3\_3C8201 AI神經網路模型介紹

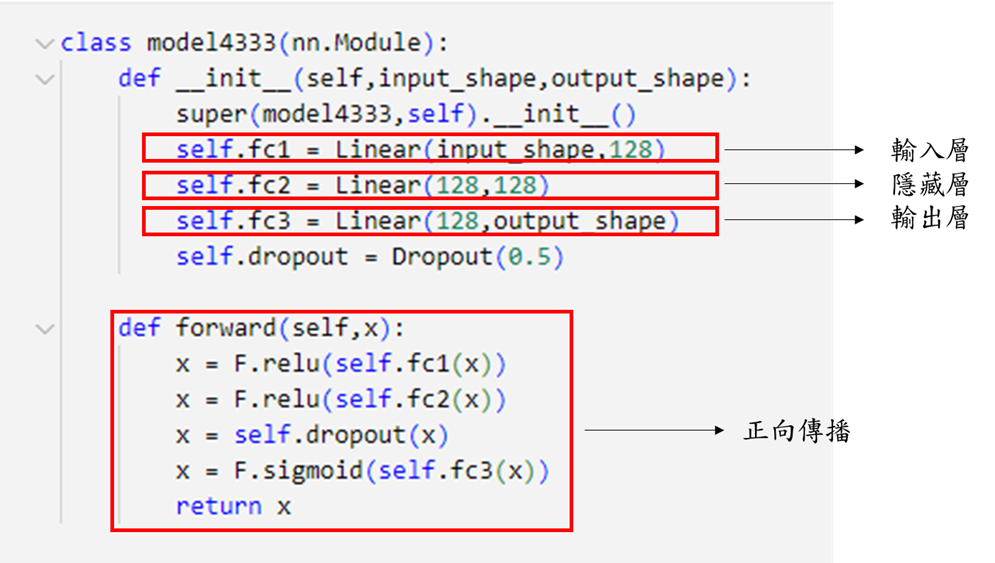
ARO3\_3C8201 AI model框架如圖4-2所示，包含一層輸入層、一層隱藏層(全連接層)及一層輸出層，藉由輸入溫度、流量、進料組成等22個特徵變數進入神經模型中進行深度學習運算，並預測出熱油氣用量、塔頂串級溫度、塔底溫度等三個目標值，其中塔頂二甲苯及塔底甲苯濃度為3C8201去庚烷塔之管制標的，製程人員可依模型預測得到特定標的濃度下之製程操作值，圖4-3為3C8201去庚烷塔神經網路模型架構。

圖4-3 3C8201去庚烷塔神經網路模型架構

工作心得

職於九月中從嘉義新港廠區輪班訓練完成，回到台北工務部自控處智能專案組報到，來到這邊從四班三輪的輪班生活與現場轟鳴吵雜的機台聲到台塑大樓安靜有紀律的生活，差異非常大，雖然需要重新熟悉日常與工作內容，但主管與同事都很親切，有問題或是工作上不懂的地方都很願意說明與幫忙。

於報到第一週，我承接了「ARO1廠C914節能優化人機介面建置」，待第一案完成後又安排「ARO3廠去庚烷塔操作AI預測畫面建置」工程案，雖然之前有寫過相關後端程式的經驗，但還是有許多地方需要理解與處理，像是目前公司主要所用的PI系統、資料庫連接，或是如何架設Apache等，這些都是先前未接觸過的，為了解決這些困難，有請教前輩與同事們後慢慢解決，也有搜尋相關官方文件或網路資料等，將這些困難點在專案過程中一一釐清，並順利完成這些案件。

在專案過程中我了解到，最重要的一部分就是有效的雙向溝通，無論身處何位都必須學習如何與他人溝通，傾聽客戶實際需求，在設計案件時，從接到案件、理解消化案件、到與業主溝通、確認所需資料及業主訴求，都必須與業主進行大量的溝通才能真正地開始設計案件，藉由雙方有效的交流，既能節省彼此時間，也能加速專案的推進。

在這段時間學習到不少公司制度與規範，更從主管與同仁之間學習到不一樣的專業與處理事情的角度，在工務部需與不同領域的專人互相學習、溝通與合作，發現自己還有許多不足的地方需要更努力，也感謝公司給予資源與環境讓我學習與增進自己的能力。