

人工智慧初探

程式實作與實例分析



目錄

1. 簡介.....	1
1.1 資料科學.....	1
1.1.1 資料描述.....	1
1.1.2 資料探勘.....	2
1.2 人工智慧.....	3
1.2.1 監督式學習.....	4
1.2.2 非監督式學習.....	9
1.3 智慧雲計算.....	11
1.4 邊緣計算.....	12
1.4.1 工業物聯網.....	12
1.4.2 Jetson Nano.....	14
2. 程式語言與開發工具.....	17
2.1 Fortan	17
2.2 Python	19
2.3 R.....	21
2.4 Weka.....	23
2.5 MatLab.....	25
3. 資料描述與探勘.....	27
3.1 統計.....	27
3.2 機率.....	31
3.3 估計.....	37
3.4 迴歸.....	43
3.5 檢定.....	46
3.6 聚類.....	51
4. AI 框架與程式實作.....	58
4.1 資料來源.....	58
4.1.1 開放式資料數據集.....	58
4.1.2 資料蒐集-網路爬蟲.....	69
4.2 AI 框架.....	73
4.3 Python 實作	74
4.3.1 數字辨識.....	74
4.3.2 人臉辨識.....	76
4.3.3 自然語言處理.....	81
4.3.4 Jetson Nano.....	84
4.4 R 實作	86
4.4.1 運用 R 進行股票交易	86

4.4.2 運用 R 進行權證交易	87
4.4.3 運用 R 進行權證交易	88
4.5 Weka.....	90
4.6 雲計算.....	92
5 智慧分析應用實例.....	98
5.1 智慧製造－產線運轉狀態分析.....	98
5.1.1 境況.....	98
5.1.2 分析.....	100
5.1.3 研發.....	109
5.1.4 探討.....	115
5.2 自然語言－政經新聞輿論趨勢分析.....	121
5.2.1 境況.....	121
5.2.2 分析.....	124
5.2.3 研發.....	143
5.2.4 探討.....	152
5.3 智慧交易－金融市場市場趨勢分析.....	160
5.3.1 境況.....	160
5.3.2 分析.....	163
5.3.3 研發.....	182
5.3.4 探討.....	195
5.4 天災預警-強震預警分析	198
5.4.1 境況.....	198
5.4.2 分析.....	200
5.4.3 研發.....	207
5.4.4 探討.....	216
5.5 空間潛勢-地質災害潛勢分析.....	223
5.5.1 境況.....	223
5.5.2 分析.....	224
5.5.3 研發.....	244
5.5.4 探討.....	251
5.6 智慧導引-人員移動行為分析.....	260
5.6.1 境況.....	260
5.6.2 分析.....	262
5.6.3 研發.....	268
5.6.4 探討.....	269
附錄 A.....	280
附錄 B.....	301
About Us	321

1. 簡介

1.1 資料科學

1.1.1 資料描述

世界萬物，經由人類記載或各種感測器監測，轉由對觀測事件的描述資訊，或感測儀器的監察數據保存於世間，對於僅記載而未彙整分析的原始資料，藉由資料特徵擷取、多維度數據可視化(如散佈圖表示數據分佈情況、雷達圖表示數據分佈比例、直方圖表示資料各級數據的分佈比例.....等)可供使用者直覺理解資料內涵的數據意義，進而針對資料特性與實務需求，制訂數據使用策略。

敘述統計(Descriptive statistics)將觀測資料經數據分析，以平均數、中位數、眾數、幾何平均數、調和平均數、全距、平均差、標準差、相對差、四分差.....等多種統計分析數值來描述觀測資料的數據結構，並以數據可視化方式直觀表達資料的數據特徵與分佈情況。

記載與觀測的原始資料，可藉由統計分析以敘述統計方式，就各因子的分類或分群分析、數值上限、下限、平均值、標準偏差、各分組內數字個數.....等，透過這些統計數據描述的數據特徵，可依照分佈比例進行線性內插補值或異常點去除，再依照數據使用規劃來進行下一步的分析與資料探索。

1.1.2 資料探勘

資料探勘中常應用於分析資料的技術包括:分類(Classification)，分群(Clustering)，關聯式規則(Association rule mining)，類神經網路等。分類透過監督式學習的訓練方式，將相異數據資料立訓練樣本，並依照不同特徵屬性如數值，類別等，選擇 SVM(Support Vector Machine), Naïve Bayes, Decision tree 等不同分類器技術，依所需要分析的資料類型，找出共通的特性，完成資料類型分類。雖然分類方法已有各種廣泛應用，然而在分類前必須訓練好適當的預先定義類別，因此訓練資料的數量會影響預測的精確度。

由於資料分佈中可能有極大差異，且資料中若含有未精細分類的資料，可能影響分類結果。相對而言，分群則透過非監督式學習方法，不需事先經過訓練即能依據資料彼此相似程度，自動聚合成不同群，使得相似資料分在同一群，而不同群的相似度最低。因此分群也能夠處理未知的案例。

常見分群技術包括 K-Means clustering, Hierarchical agglomerative clustering (HAC)等。當資料量逐漸增加，分群演算法的計算量通常也會較大；而且在資料並非一次全部收集完成的情形下，每次重新分群的負擔太大，將影響其實用性。

Association rule mining 發掘資料庫中經常出現的項目，以找出其關聯性。例如 入侵偵測系統可以從入侵者的過往紀錄，偵測出未來可能的網路攻擊。受到人類大腦神經元連接方式的啟發，類神經網路利用不同節點彼此連接及自我調適的能力，根據過去所觀察到的不同例子來自主學習，類神經網路得以處理各種輸入並產生對應的輸出。不論是監督式學習，或是非監督式學習，大多是經過特徵擷取 (Feature extraction)，特徵選取 (Feature selection)，模型學習等階段後，利用各種不同方法進行資料相似度 (Similarity estimation)，以完成資料分析。

1.2 人工智慧



諸多學者觀察生物生活習性或演化過程激發，建構演算法來模擬各類生物於自然環境中的訊號處理、覓食抉擇、突變進化等現象，使得機器能依演算法設計進行認知、學習、推論等工作，就目標問題找出較優或最適解。此類運用人工智慧系統來解決問題的方式可謂之人工智慧 (Artificial Intelligence, AI)，其意指使用科學和工程方法建置智慧型機器，早於 1969 年代 McCarthy and Hayes 即以哲學角度探討人工智慧的知識學習方法及邏輯演算型式。

人工智慧 (Artificial Intelligence, AI) 自 1950 年發展至今，用以發展智慧系統、智慧型機器來延伸人類智慧，擴展智慧行為。在機器學

習、專家系統、智慧製造(或稱工業 4.0)、電腦視覺、資料採擷 (Knowledge Discover in Database, KDD)、自然語言處理、智慧車(自駕車)、機器人等相關領域取得良好應用。其專研以人工方法與技術令電腦能如人類般思考，適逢現代電腦高速計算能力與大量記憶體優勢，對於原先需要人類智慧才可推論的困難問題，讓電腦以機器學習方式自我調整生成解決方案。

人工智慧發展歷程中，歷經多個發展時期，1930-1950 年發現生物神經元傳遞訊息分為通過、不通過等兩種狀態，並於 1951 年發展出早期神經網路系統；1950 年發展出圖靈測試，假定當 AI 以非接觸式對方和人類對話，若該人無法分辨發言者為 AI，則可稱該 AI 具有智慧；1951 年發展西洋棋 AI，遊戲 AI 為評估當前時代人工智慧水準的方式；1956 年正式提出人工智慧定義；1943-1956 年人工智慧學門的誕生；1997 年 IBM 發展深藍(Deep Blue)優勝當時人類西洋棋王；2017 年 Google 發展 AlphaGo 優勝當時圍棋世界第一棋士，掀起 AI 熱潮。

1.2.1 監督式學習

類神經網路

類神經網路源自於觀察生物神經細胞交換訊息流程並模擬的計算模式，藉由神經元(Neuron)及各單元間的連結(Links)和連結權重

(Weight)來模擬人類神經學習新知之方式，為人工智慧之一環。類神經網路由大量彼此聯系的神經元及連結所構成，由神經元的數目、連結方式及各連結權重共同建構此分析模型，藉由眾多的訓練資料，使類神經網路學習輸入資料與目標資料間的關聯，當網路模型接收輸入資料後經神經架構層運算而得一結果值，經與目標值的誤差計算後，反向傳遞回類神經網路修正權重，透過多次學習減少網路模型分析結果值與實際目標值的差距，此類神經網路模型即為倒傳遞類神經網路，擁有高複雜非線性系統的分析能力，可運用於建立複雜分析境況之輸入資料與預測資料間的關聯，屬監督式學習法。

類神經網路為一個非線性動態系統，由大量互相聯繫的並聯神經元所構成，各個神經元皆包含一部分的動態資訊和簡單的計算能力，透過輸入值與目標值之誤差計算各神經元彼此間之訊號關連性，藉由大量的訓練動態各個神經元間之加權值，並以各處理單元的數值權重決定該神經元所傳遞的訊號強度，各神經元之訊號傳遞如式 1.2.1、式 1.2.2 所示：

$$U_i = \sum_{j=1}^m W_{ij} V_j + I_i \quad \text{式 1.2.1}$$

$$V_i = f(U_i) = \frac{1}{1 + \exp(-U_i)} \quad \text{式 1.2.2}$$

其中(U)來自於神經元本身匯集上一層各個神經元發出之訊號，
乘上被分配到的權重，並加總組合經活化計算而得，此活化作用是透過 S 型函數(sigmoid)將訊號侷限於 0~1 之間，或可使用 Relu...等。

倒傳遞類神經網路為多層前饋類神經網路 (Multi-layer Feed-forward Neural Networks, MLFF)之一種，由輸入層、輸出層及一或多層隱藏層組成，將數個神經元安排在各層中，各層的神經元彼此互相連接，然而同一層的神經元互不相連。屬於監督式學習網路。類神經網路是由人工神經元所組成，可依序分為輸入層、隱藏層及輸出層等三個部分，輸入層用以接收網路的輸入變數，隱藏層負責處理類神經網路中各單元間的交互影響，輸出層將網路運算結果輸出。

監督式學習可以不斷修正網路中的傳遞權重，以符合使用者的期待，在學習的過程中給予類神經網路訓練範例，每一個訓練範例中同時包含輸入項和目標輸出值，目標輸出值會不斷的督促網路修正傳遞權重的值(如式 1.2.3)，藉由訓練過程中逐步調整網路連結的強弱，來降低網路輸出值與目標輸出值之間的差距，直到差距到達網路使用者給定的誤差量才告停止。

$$\Delta w_{ij}(t) = -\eta \frac{\partial E}{\partial w_{ij}} \quad \text{式 1.2.3}$$

其中

W_{ij} 為神經元編號 ij 的權重

η 為學習速度

多種神經網路架構(深度學習、強化學習、長短周期、生成對抗)

人工智慧自 1950 年發展至今，用以發展智能系統、智慧機器來延伸人類智慧，擴展智慧行為。在機器學習、專家系統、智慧製造(或稱工業 4.0)、電腦視覺、資料採擷(Knowledge Discover in Database, KDD)、自然語言處理、智慧車(自駕車)自動駕駛系統、機器人等相關領域取得良好應用。其專研以人工方法與技術令電腦能如人類般思考，適逢現代電腦高速計算能力與大量記憶體優勢，對於原先需要人類智慧才可推論的困難問題，讓電腦以機器學習方式自適應生成解決方案。

早於 1960 年代，McCarthy and Hayes 即以哲學角度探討人工智慧的知識學習方法及邏輯演算型式。而其中的機器學習(Machine Learning, ML)在數十年的變革中有重大的進展，基於類神經網路演進出身深度學習(Deep Learning)，其可解決多種類型問題被視為通用學習方法，包含監督式、半邊都式、部分監督式以及無監督式等。藉由神經網路的結構、網路層次和各神經元的權重，共同描述特定求解問題的學習結果，自 2012 以來發展出多種神經網路架構，用於各類領域的高度複雜性問題。

深度神經網路(Deep Neural Network, DNN)建構數十至數百層神經網路架構，以窄而深的多層數隱藏層網路架構強化對求解問題的高維空間解析，力求更佳學習效果且在獨立的樣本資料中有良好表現；捲積神經網路(Convolutional Neural Network, CNN)針對電腦圖學之影像識別比較，建構通用像素特徵轉化方式，萃取影像特徵進行學習解析與識別並運用於自然語言等其他領域；遞迴神經網路(Recurrent Neural Network, RNN)透過神經網路的神經元遞迴運算，可強化學習輸入資料間的前後依賴關係，運用於時序性資料和自然語言。

長短期記憶神經網路(Long Short Term Memory, LSTM)針對時序性資料，抽取短時間的資料特徵且能記憶長周期的資料特性，適用於自然語言處理中的文本生成、連續對話語意解析、金融領域的市場資訊解讀等；門閥遞迴神經元(Gated Recurrent Units, GRU)可改善神經網路的學習、記憶、遺忘及更新；自動編碼器(Auto-Encoder, AE)以神經網路方式將輸入資訊轉為高維特徵；生成對抗神經網路(Generative Adversarial Network, GAN)將原先類神經網路評估學習效果的成本函數，轉化為另以獨立的神經網路模型，來評估該次類神經網路運算的學習效果，透過兩個神經網路彼此的訓練、對抗而強化整體問題的深度解析，多用於電腦圖學的影像生成如影像風格轉化、人臉替換等；

深度強化學習(Deep Reinforcement Learning, DRL)透過深層神經網路與學習成果回饋改善網路。

1.2.2 非監督式學習

生物演算法

近年來因 AI 技術的蓬勃發展，有許多 AI 領域的各類分析方法如深度學習(Deep Learning)、類神經網路(Artificial Neural Network, ANN)、基因演算法(Genetic Algorithms, GA)、粒子群演算法(Particle Swarm Optimization, PSO)以及人工蜂群演算法(Artificial Bee Colony, ABC)等仿生計算，近年已大幅度運用在多種領域，而工業生產也借重此類分析方法探索制程中的優化因子。

粒子群演算法由 Eberhart and Kennedy 在 1995 發展之啟發式演算法，經 Shi and Eberhart 在 1998 年針對慣性因子來優化，藉由鳥類群體覓食，分散搜尋的生物行為模式創立 PSO，其具有計算式簡易，搜索快速的優點，搜索廣度隨分組鳥數的提升而增大，也影響計算分析時間，屬非監督式學習法。PSO 藉由多組粒子同時搜索較佳解，將目標問題的數值邊界映射至求解空間，如同多組鳥于求解空間中飛行移動，鳥群于求解空間的移動軌跡，可視為該組鳥的求解過程，藉由各組鳥于求解空間的移動路徑，即可瞭解該求解問題之歷次反覆運算行為。求解空間中每一粒子的向量即代表一可行解，參照各粒子當前所

在位置，透過分享各組鳥的歷次解，使用局部最佳解(pbest)和全域最佳解(gbest)來調整每次粒子新的躍進向量，收斂後各粒子將趨近於同一點，找到全域較佳解。

優化探索

自 2010 年起，現代電腦的高速計算(包含 CPU 計算、GPU 計算)、大量記憶體、高速儲存媒介的發展，網際網路的世界聯通性，結合長久對環境、製造、醫療、金融等體系的數據資料以及近年社交網路領域的社群數據，為人工智慧發展的基石與應用領域，即用人工的方法和技術，研製智慧型機器或智慧系統來模仿、延伸和擴展人的智慧，實現智慧行為。

諸多實務問題擁有眾多變因且具有不確定性，針對問題探求解決策略時，需對變因變化情形和境況不確定性進行模擬演變，可將目標問題視為由眾多影響因子共同構建成的互動模型。個別影響因子的辨識尺度、解析要點、評估策略可能差異甚大，兩兩因子間可能互相獨立或互相關聯，包含線性解算和非線性演化，基於實務問題力求有限時間、資源內尋求較佳的解決策略，而非群舉所有可能性與花費高昂資原建立理論最佳解決策略，故經由優化演算法(如 ABS, Fuzzy, GA, PSO,... 等)能針對當前關注課題與成本函數(Cost Function)找出全域

較佳解，而如何避免區域較佳解進而找到全域的更好解決策略，為智能算法建構的重要考量要點。

1.3 智慧雲計算

雲計算將應用、資料和資訊技術透過網路提供相關服務給使用者使用。雲計算也是一種基礎架構管理的方法論，大量的計算資源組成資訊技術與運算資源池，用於動態創建高度虛擬化的資源提供用戶使用。雲計算可以追溯到 1990 年左右開始的網格計算，以及後來的效用計算和軟體即服務(Software as a Service, SaaS)，逐漸演變而成。雲計算可總結為三個主要的層次：基礎設施服務(Infrastructure Services)，平臺服務(Platform Services)和應用服務(Application Services)。

平臺服務將應用運行所需的資訊技術與運算資源和基礎設施以網路服務的方式提供給使用者，包括了仲介軟體服務，資訊服務，連通性服務，整合服務和消息服務等多種服務形式。為實現平臺服務，業界提出了“平臺即服務(Platform as a Services, PaaS)”的交付模式。PaaS 模式，基於互聯網提供對應用完整生命週期(包括設計、開發、測試和部署等階段)的支援，減少了用戶在購置和管理應用生命週期內所必須的軟硬體以及部署應用和資訊技術基礎設施的成本，同時也降低了以上工作的複雜度。為了確保高效地交付具備較強靈活性的平臺服務，在 PaaS 模式中，平臺服務通常基於自動化的技術通過虛擬

化的形式交付，在運行時，自動化，自優化等技術也將被廣泛應用，以確保即時動態地滿足應用生命週期內的各種需求。

研發者將生產線的各種觀測資料、廠區整體各區域細部能源及用水耗用情形、工廠多種環境參數，綜整至雲端系統，藉由同步異地備援機制保存各類資料並進行大資料分析以及特定人工智慧演算法推論，管理者與相關使用人員能透過互聯網以手機、平板為顯示載體，異地控管廠區整體狀況，達到即時管理之效，整體流程如圖 1.3.1。

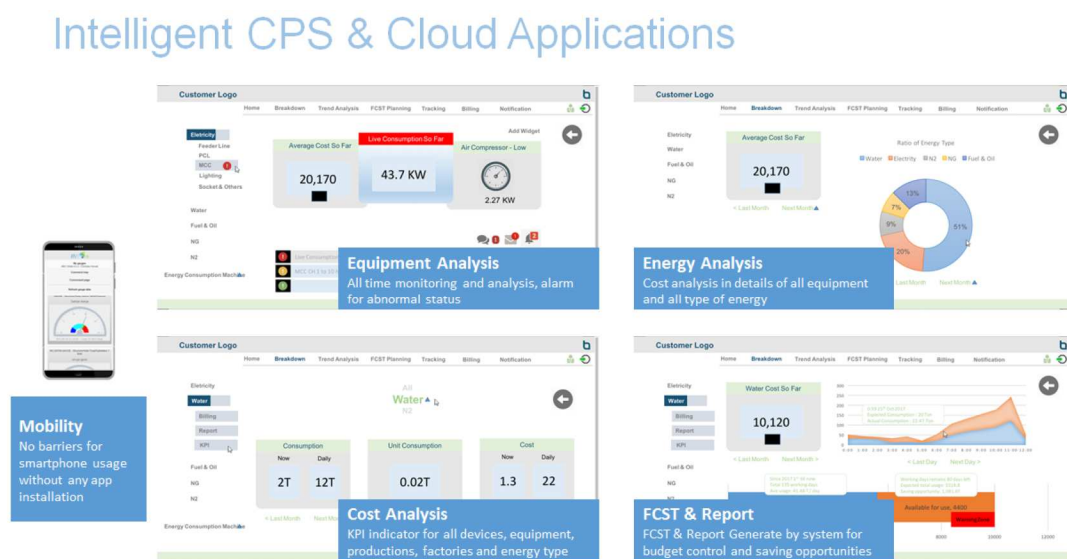


圖 1.3.1 智慧雲計算示意圖

1.4 邊緣計算

1.4.1 工業物聯網

自 1996 年聯合國經濟合作與開發組織(Organisation for Economic Co-operation and Development, OECD)宣告知識經濟時代以來，知識由

原先的承先啟後演變為可作為經濟活動的交易載體，人類探索自然、工藝精進及交流互動等執行過程中的資料，演進至現今可由雲端資料彙整，經大資料分析提取關鍵資訊，用以追蹤環境變化、潛在趨勢變遷、工業生產之品質控管等，對於環境變遷領域、工業 4.0 深有幫助。

若能藉由通訊模組與多種環境感測器、工業產業之機具相連，由人與人的即時聯繫擴展至物聯物，進而延伸至人、物、環境、空間及時間整體資訊聯繫彙集之萬物互聯的萬聯網(Internet of Everything)資訊交流狀態，人們將更能便捷享受教育、醫療等多種資源來改善生活。

經由介面控制器可與現有之多種工業規格接頭銜接(如 RS485, RS232, DI/DO 等)，無縫整合至已有之運行體系，多種工業通訊規範已實行眾多領域之廠區，因廠內充斥多種金屬物件、帶電設備具有電磁場干擾多種常見免費頻段之無線通訊模式(如 Wi-Fi)，為確保廠區內各設備之連線正常，大多為有線通訊佈設廠區。

以 RS232 有線通訊為例，若通訊線路超出約 15 m，則訊號將逐漸衰減，超過 30 m 則訊號傳遞不良，以有線方式架設各機具間的通訊環境時，需考慮佈線建置設計、施工成本(人工佈線以長度計價，佈線越長花費越高)以及管理維護情形，當單一有線線路異常，需逐步探查整體線路的好壞，隨者線路佈設總類增多、總長度增長和建置

地點的複雜化，皆會提高管理與後續維護的難度。如以無線通訊取代有線通訊模式，進行機械設備的通訊服務(Machine to Machine, M2M)，可改善實務上因線路通訊長度、施工不易的問題，研發者藉由長距離無線傳輸模組與介面控制器無縫連結已有之機具介面，經高容量開道架構大型無線物聯網，應用於物聯網之中國製造 2025 與工業 4.0 等領域，工業資訊傳遞架構如圖 1.4.1。

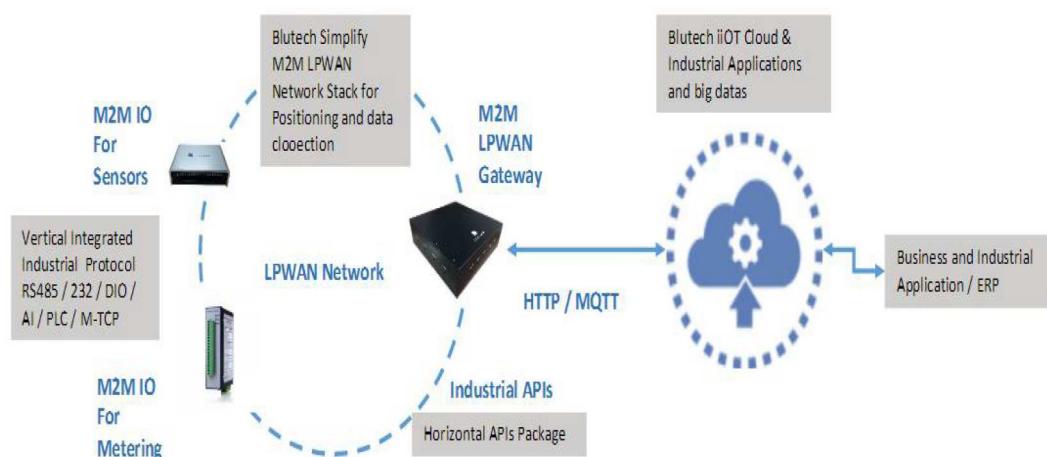


圖 1.4.1 工業物聯網之資訊系統架構示意圖

1.4.2 Jetson Nano

人與人藉由手機彼此聯繫，拉近人們之間的溝通距離；而物與物的相互聯繫，受惠於無線通訊技術的推進，異地即時聯網通訊技將觀測環境的事物經由各類感測器，透過物聯網觀測機制結合互聯網的即時雲端人工智慧分析機制，可協助使用者對特定事物進行進早預判、事前預警、工作進程智慧協作以及事後複檢等效用。

以智慧製造為例，透過物聯網觀測機制結合互聯網的即時產製零庫存策略，藉由人工智慧針對生產製程的優化技術(如提高良率、拉高稼動率)，採用對環境友善的生產製程和設備機具，並即時廣泛監測廠區周遭環境狀況(如土壤、空氣、廢棄物、水)，減少製造生產對廠區環境品質的衝擊。

將資訊經各感測器彙整至區域運算節點，再總合傳遞至雲端進行智慧分析與任務指派，可將遠地控管各觀測區域的實時狀況並調配。然而對部分智慧分析與判定功能，可下放至區域運算節點，減少資訊彙總至雲端以及等待雲端指派的計算耗時與通訊失聯風險。將人工智慧、分析程式、資料儲存等運用於各區域運算節點，由集中式雲運算改為分佈散式運算架構，以更小、更靈活的計算單元應用整體服務可稱為邊緣運算(Edge computing)。

因應人工智慧的興起，對區域運算節點的硬體配置須考量智慧分析的平行加速需求(如顯示卡運用於平行運算用於加速巨型矩陣乘積)。NVIDIA 至 1993 年起致力於圖形處理器的設計與銷售，於 2007 開放 CUDA 讓使用者可自訂義數據運算邏輯植入 NVIDIA 的圖形晶片卡來進行數值運算加速，對於 2016 年的第三次人工智慧熱潮興起有非常顯著的助益。

有鑑於邊緣計算的智慧分析需求，NVIDIA 發展輕量級(70 x 45 mm)、低耗能的人工智慧應用平臺- Jetson Nano(如圖 1.4.2)，可用於家用機器人(智能環境偵測與移動判定)、網路攝影機(智慧影像識別)、智慧開道器等，應用於自駕車和機器人領域(如圖 1.4.3 的智慧小車)、智慧城市等各類終端設備的智慧分析需求。



圖 1.4.2 Jetson Nano 多視角攝影圖



圖 1.4.3 智能小車應用實例

2. 程式語言與開發工具

2.1 Fortran

自十九世紀電腦問世以來，人類的生活方式與歷史演進獲得極大之進展，人們藉由電腦的幫助簡化了許多繁瑣複雜的運算，而演進至今二十一世紀，電腦已走入人類生活中的各個層面，普及化成為生活必需品之一。在各種應用領域，如國防、通訊與學術領域等，皆可看到使用電腦來輔助相應技術發展之實例。電腦為需輸入指令並依照指示執行之工具，人們需先依其使用需求以電腦瞭解之語言來與電腦溝通，使其能依照使用者之想法與要求進行相應之資料處理、數據分析及情境模擬等。在電腦早期發展時代，需以機械語言來與電腦溝通，然機械語言和人類用語有極大之差異且不直覺，增加使用者學習方面與撰寫程式設計之困難度，因此往後陸續發展出組合語言與高階語言來協助使用者進程式設計。

高階語言有 C/C++、Java、Cobol、Fortran、Python、Rube.....等，每種語言皆有其各自之特色，如 C/C++ 廣為業界所用、Java 具有物件導向與跨平臺功能、Cobol 為商業上所常用之語言，而 Fortran 於數值分析領域中，其數據處理能力較上述語言為佳。Fortran 自 1957 年由 IBM 推出第一個 Fortran 編譯器以來，歷經了 Fortran 66, Fortran 77,

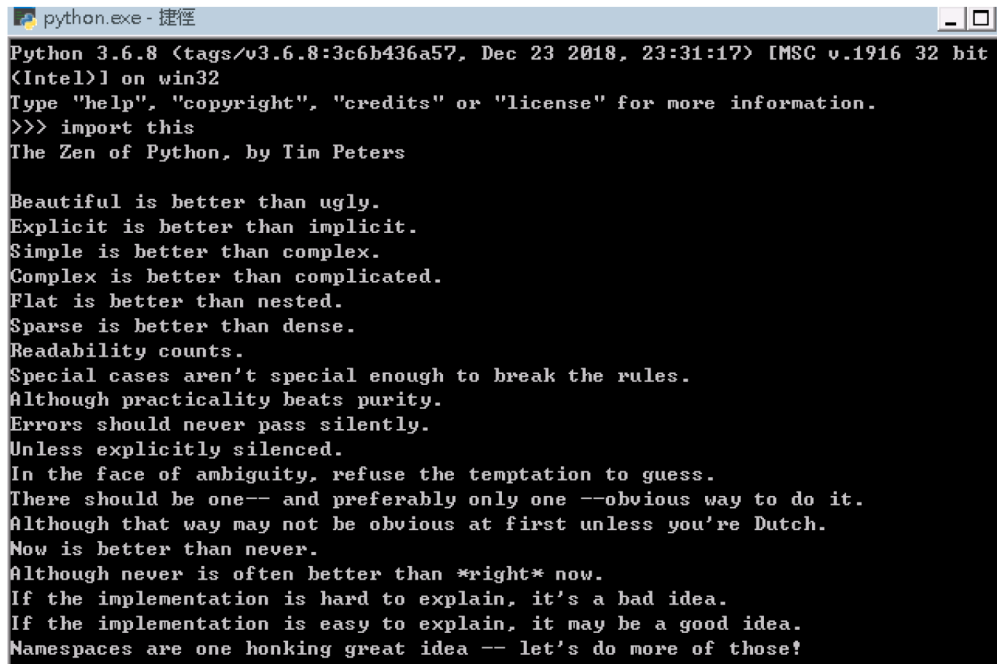
Fortran 90/95 和 Fortran 2003 這些版本的演進，至今已有 60 年的歷史，為市面上歷史最久的高階語言，廣泛運用於土木工程領域中，如美國普渡大學開發之邊坡穩定分析程式 STABL。Fortran 於理工領域之數值處理而言，具備複數運算、陣列索引功能，而其矩陣運算也能供使用者以較直觀的數學方式來寫入，陣列處理能力極優，且 Fortran 為了能提高數據運算速度，使用傳址呼叫來傳遞變數，當主程式呼叫子程式時，其主程式變數與函式中變數之記憶體位置相同，省略資料傳遞時間，以增進運算效率。Fortran 除了數值運算能力較佳外，因其歷史極為悠久，現存有許多免費、公開且高效率之數值運算函式可供參考，程式開發人員能藉由大量的舊有優質程式碼，運用前人研究成果大幅減少開發數值模型之困擾與時間。

Fortran 95 其對舊有版本支援度高，能參考前人所使用之數據函式來增進與改善系統功能，而相對舊版而言，Fortran 95 提供 Free format 的自由寫作格式，由於早期 Fortran 為打孔機輸入方式，故格式為 Fixed Format，而 Fortran 95 之 Free format 能提供程式開發人員較方便、自由之寫作方式。Fortran 95 加強了陣列處理能力、矩陣運算能力與動態陣列等功能，如在矩陣運算方面，舊有寫法需使用迴圈方式來撰寫兩矩陣彼此元件的乘積，而於 Fortran 95 則可藉由內建函式以直觀之數學計算式寫法來撰寫。

2.2 Python

Python 於 1991 由 Guido van Rossum 開發並發布第一版，著重程式碼的可讀性、簡潔性(相較 C, C++以大誇號{}或關鍵字，其透過空格縮排來分段各程式碼區塊)。為直譯式的高階程式語言，擁有動態型別系統、廢棄參數空間自動回收與記憶體空間釋放功能，能自動管理記憶體空間分派與回用，支援多種程式設計法(Programming paradigm)如物件導向、命令式、函數式和程式式.....等。

Guido van Rossum 延續 ABC 語言(由 Leo Geurts、Lambert Meertens, Steven Pemberton 共同創建)致力於協助非專業程式設計師可容易進程式開發的宗旨，創建開放原始程式碼(Open source)的開源 Python，期許能為一個簡單直觀的程式語言且有強大競爭力；程式碼公開且任何人都可為 Python 貢獻；程式語言具有易讀性如自然語言-英語；對短期開發可有效縮短開發時程。以「優雅」、「明確」、「簡單」等原則來建構 Python 的設計哲學，設計準則如圖 2.2.1。



```
python.exe - 捷徑
Python 3.6.8 <tags/v3.6.8:3c6b436a57, Dec 23 2018, 23:31:17> [MSC v.1916 32 bit
<Intel>] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

圖 2.2.1 Python 格言

Python 可用於 Web 全端開發，經由 PWSGI (Python Web Server Gateway Interface) 協調網路伺服器(http)與 Python 和 Web 間的溝通，而藉由 Python-Web 設計框架(如 Django, Flask(常用於輕量級 Web 系統快速開發), Pyramid, TurboGears.....等)可讓程式設計師依循框架元素構建 Web 系統，如透過 Flask 架構智慧金融(Fintech)分析系統。

在科學計算方面，Python 與 R 常用於數據分析、演算法建構等，如 NumPy, SciPy, Tensorflow, Matplotlib 等 Python 科學計算套件，對於數值統計、資料數據描述、矩陣運算拆解、聚類演算法與迴歸分析、人工智慧之監督式或非監督式學習法等甚有助益，待資料集經由演算法解析後，可由 Matplotlib 輸出多種數據分佈與比較分析圖，用於使

用者探索數據價值、真理挖掘與智慧模型建構之用。為了提升運算效率、運用 C, C++, Fortran 等編譯式語言先行建構特定演算法的運算模組元件，再由 Python 建構整體邏輯運算步驟並呼叫運算模組元件，可在運算效率與廣泛通用性取得雙贏，廣泛用於科學計算中。

在作業系統中，Python 對多種網路協定有良好的支援，可用於伺服器軟體開發、或撰寫網路爬蟲自動蒐整各類網路資訊，與各類程式語言的良好接合，令 Python 可如膠水般聯繫異質程式語言用於共同作用目的上，被戲稱為膠水語言。因應 Python 的廣泛應用、易讀好上手等特性，諸多高科技資訊公司(如 YouTube, Google, Yahoo!)與研究單位(如 NASA)大量運用 Python 於快速原型產品開發、科學計算演算法構建與分析、網路系統運用等。

2.3 R

R 語言於 1993 年由 George Ross Ihaka and Robert Clifford Gentleman 開發而成，後由 R 開發核心團隊進行研發與維護，基於 Scheme 語言的實現，開放原始碼軟體(Open source software)。R 大多以命令式方式進行操作互動，而 RStudio 為當前較多人使用的跨平臺整合開發環境，可於 Unix, Windows and MacOS 等多種作業系統上運行。

R 專注於統計分析(Statistical Analysis)、資料探勘(Data mining)、大數據分析(Big Data Analytics)與資料繪圖呈現，近年來因應人工智慧的興起，也支援深度學習的分析框架。R 包含多種統計學理和數據分析能力，並可由多種自訂義套件(Packages)擴充分析能力，如使用 rpart 建構決策樹；使用 e1071 建構支撐向量機(Support vector machine, SVM)分析模型；使用 Keras, TensorFlow for R 建構深度學習模型以訓練並測試；運用 Spark 集群來分析數據.....等，相較收費統計分析軟體(如 SPSS, SAS)，R 為統計分析、資料探勘分析專業人員的常用分析工具。

在Web 方面，經由網路協定支持套件(如 rsconnect)，可以簡潔命令指令進行經自有伺服器或雲端伺服器(如 shinyapps.io)進行網際網路通訊；而在資料庫連結方面，同樣的數據程式碼可經由 dplyr 支援多種異質資料庫，可用於本地或異地資料庫。這些數據分析成果，則可經由 ggplot2 套件生成多種客製化成果(如圖 2.3.1)，R 已有 7000 餘種的擴充套件，對財經分析、經濟計量、社會科學和人文科學相關研究人員甚有助益。

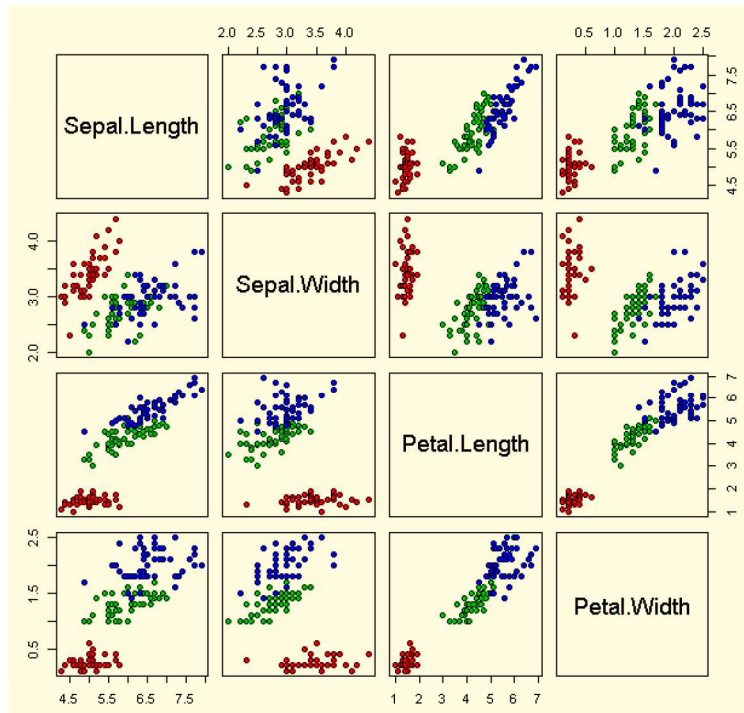


圖 2.3.1 R 數據分析分佈圖範例

2.4 Weka

Weka (Waikato Environment for Knowledge Analysis) 由 University of Waikato 自 1994 發展以來，當前為 Weka 3。Weka 3 使用 Java 為開發語言，為基於 GUN 公開授權的開放式軟體，專注於資料探勘(如資料前處理(Data preparation)、分類(Classification)、聚類(Clustering)、迴歸(Regression)、關聯式規則探勘(Association rules mining).....等)及機器學習演算法，且支援深度學習(Deep learning)。廣受到非程式開發專業人員，卻需進行資料分析與人工智慧探索的財經分析、經濟計量、社會科學和人文科學相關研究人員的長期使用。

Weka 由 Program, Applications, Tool, Visualization, Windows, Help 等 6 個部分所構成，在 Program 中記載 Weka 運行記錄於 LogWindow；在 Applications 中表列 Weka 的主要應用程式，首先由 Explorer 建構目標資料的探索環境、再由 Experimenter 管控各類分析演算法與相應的統計檢定、接者由 KnowledgeFlow 進行增量學習 (Incremental Learning)、最後針對無命令模式的作業系統可經 SimpleCLI 執行 Weka 命令。

在 Tool 中包含多種有助於分析的工具，可查詢資料庫、建立集成式選擇介面；在 Visuallization，可將資料分析成果依據檢視目的，分別輸出為二維散點圖 (Plot)、比較分析與收斂成效的 ROC curve (Receiver operating characteristic curve)；最後可經由 Windows 與 Help 視窗分別檢視當前所有工作進程視窗和查詢 Weka 使用上的相關問題。

分群分析常用於將未建立關聯的觀測資料進行分組與歸類，以 K-Meanss, Expectation 常見用於分群演算法，而決策樹 (Decision tree)、快速決策樹 (Very fast decision tree)、隨機森林 (Random decision forests) 等常見用於分類演算法。對於與時間相關的時序資料分析，Weka 可經迴歸分析、類神經網路分析、之稱向量機進行線性多變數迴歸或非線性數據預測，包含人工智慧及機器學習相關的演算法。

2.5 MatLab

MatLab(MATrix LABoratory)於 1984 年由 Cleve Barry Moler 為提高教學互動成效由 Fortran 開發出第一版，後由 John N. Little 等人創辦的 The MathWorks 以 C 重新編寫，廣泛用於各領域專業人士與研究學者，可進行數值計算如統計、最佳化、線性代數、數值積分、傅立葉分析.....等；可進行互動式工具設計用於數值疊代、問題探索、求解方程設計；可用於多種專業領域如物理、化學、機械、結構等數值模擬問題分析與求解；可呈現多種資料視覺化以二維數據趨勢圖或三圍數據分佈圖展示分析成果；可與多種外部程式語言(如 C, C++, Fortran)於 Linux, Windows 混和開發，歷經數十年的發展，全球已有數百萬的專業工程師和科學人員用於多種專業領域。

MatLab 以開放式的 MatLab 創建多種工具箱(Toolbox)用於多種領域，在數值分析與最佳化中(Math and Optimization)，包含 Curve fitting toolbox, Optimization toolbox, Global optimization toolbox, Symbolic math toolbox, Mapping toolbox, Partial differential equation toolbox；在人工智慧、資料科學與統計分析中(AI, Data Science, and Statistics)，包含 Statistics and Machine Learning toolbox, Deep learning toolbox, Reinforcement learning toolbox, Text analytics toolbox, Predictive maintenance toolbox；在資料庫方面可以 Database toolbox

進行資料連結設定，而在平行運算可藉由 Parallel computing toolbox 進行運算設定。

在機器學習方面，由 Statistics and Machine Learning toolbox 可以統計分析探查數據結構與分佈特性，藉由數據分佈機率與隨機擬真數值，補全或移除數據集中的異常值。而對於多維數據集，可以特徵分析鑑識數據集中的特徵；以逐步迴歸(Stepwise regression)將各變數分別導入模型經 F Test 去除非顯著者和避免多元變數的共線性；以主成分分析(Principal components analysis, PCA)找出數據的各特徵向量與權重，可有效降低數據維度並保有數據特徵。

待分析完觀測數據集，可以監督式學習法如 SVM, Decision Tree 針對學習資料建模而學習資料特徵，可以分群演算法如 K-Meanss, K-Nearest Neighbor, K-Medoids 進行數據分類，或由分層聚類、HMM(Hidden Markov Model)探索數據中的隱含參數。

在深度學習方面，由 Deep learning toolbox 延續 Neural network toolbox 基礎，以資料流方式提供多種神經網路演算法(如 CNN, RNN, LSTM)對中小型數據集以 SqueezeNet、Inception-v3, ResNet-101, GoogLeNet and VGG-19 網路模型進行學習運算或遷移學習(Transfer Learning)，而對於大型數據集可經多核心平行運算或雲端叢集運算來加速。

3. 資料描述與探勘

3.1 統計

萬物變遷，世事時變，為描述世上萬物的外表樣態、行為互動，內裡材質，可由口述代代相傳、文字記載或影音旁觀紀錄描述，而描述事物的性質能用質性的方式，經相對比較、好壞優劣表達物件與外界互動的資訊；也能用數值量化方式，相較質性描述更精確表達物件本體意涵，如物體的長寬高、使用率，並探討事物與外界互動的情況。

對於事物本體狀態及其與外界人事物的互動行為，經由觀察、蒐集、分析、探討，以數值資料(Data)對特定事物進行描述、解釋，歸納多個數值資料並對這些數據進行分析評估，且能對同類事物進行演譯，由已知的觀察樣本，經數據探討後，用於不確定性的類似事物，近而論述母體性質且擴展至數據分佈特性，此類由觀察樣本資料，進行推論分析用於具有不確定性的性質樣態推論，供使用者用於決策分析的方式隸屬於統計學(Statistics)與不確定性風險分析(Uncertainty risk analysis)。

統計使用數字來描述事物樣態，周朝已記載大量關於政府收入、支出的數據，埃及、巴比倫則使用計數的概念用於人口普查，在 16 世紀，統計多用於描述國家地區的整體當前狀態，結合數據圖表供使

About Us

We are a team consisted of data scientists and developers, working for the academics and corporations. The approaches focus on artificial intelligence, heterogeneous data, big data analytics, and data visualization. We assist researchers with various services, such as data collection, data mining, data transformation, Big data analysis, statistical analysis, data visualization, and technology implementations. The diverse applications include smart manufacturing, engineering construction, environmental sustainability, E-commerce advertising optimization, text emoticons, and semantic text analytics.



ZP

NTUT

Ph.D.
Assistant Professor

- College of Engineering
- AI, Big Data analytics, Data visualization
- Engineering safety, Water resources management, Disaster prevention
- Smart manufacturing(Industry 4.0), Automated energy monitoring
- E-commerce, Natural language processing, AI-Fintech

SG

NTU

Ph.D.
Visiting scholar.

- Department of Civil Engineering
- AI(Deep Generative Model, Variational Autoencoder, Generative Adversarial Network)
- Structural Engineering, Civil Engineering
- Structural damage diagnosis, Earthquake Early Warning

Peter

NTU

Ph.D. candidate
Experts in Industrial IoT

- Graduate Institute of Electrical Engineering
- AI, Big Data analytics, Data visualization
- Advanced intelligent image analysis
- Smart manufacturing(Industry 4.0), Automated energy monitoring

Mark

NTUT

Ph.D. candidate
Experts in IT

- Department of Computer Science and Information Engineering
- AI, Natural language processing, E-commerce, AI-Fintech
- System test and verification, System stability and reliability management, Performance analysis
- Cross-platform software system, Scalable cross-platform infrastructure, Huge heterogeneity virtual system,

Bruce

NTUT

Experts in IT.
20 Years' Experience

- College of Engineering
- Civil Engineering, Geotechnical Engineering
- Develop detailed project plan, Project status updates to monitor and track progress
- Ensure overall solution or service fulfills user requirements and expectation
- Perform risk management to minimize the project risks.
- Resolve any issues and help look for resources to solve problems throughout the project life cycle.

Echo

MCU

Experts in IT .
15 Years' Experience

- Department of Criminal Justice and Department of Computer Science and Information Engineering
- Software development planing , System test and verification, System stability and reliability management
- System architecture planing and design, Establish cloud computing platform
- Review & Track project deliverables produced by stakeholders, business analysts, technical service owners & QAs

Zen

NTU

Ph.D. candidate
Founder CEO of TDUGR

- Institute of GeoSciences
- Fluorescence and Micro Raman Spectroscopy of Minerals
- Sedimentary environments and sequence stratigraphy

Frank

NTUT

Ph.D. candidate
Founder CEO of Hourmasters

- College of Engineering
- Civil Engineering, Geotechnical Engineering
- E-commerce, SEO
- Start-ups, Strategic Planning, Business Development, Entrepreneurship
- Product Marketing, Project Management