何謂大數據分析：
廣義來說，大數據分析是由 題目 、 資料 、 解題方法 三部分組成
1.定義問題：2.收集資料：3.資料給電腦來找出規則：4.使用規則來預測
大數據分析功用：1 由電腦自行理解，並協助我們找出問題隱藏規則，再發展應用
(分類)判斷危險駕駛 (迴歸預測)找到選手、對手規律 => 糾正、攻克

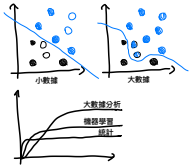
題目：由廠商或是個人定義對廠商有利的題目
→題目通常非常客製化，解決方法都差很多
資料：所有與題目有關的資料都要被盡量收集
→必須施涵蓋所有可能發生狀況 →魔鬼藏在細節中，資料是大數據分析的核心
解題方法：視題目定義與資料格式挑選合適方法
→只要能達到目的，什麼方法都可以，不一定要特殊方法，業界鼓勵使用套件
我們人類也需要幫忙一些=>如:找出並篩選不合理值資料
合併欄位 合併分散的資料 或是 資料擴增
或是 判斷是不是 高斯分佈 代表可以合理降維度 =>RBF
把資料視覺化 超重要 不然模型 和 結果 做出來很可能是錯的

大數據分析的痛苦：
不知要做什麼，無法定義題目；2.資料無法量化，格式不對，偏差；3.KPI 異動

成功案例
1 便利超商的熟食消費分析與櫃位配置 2 麥當勞報報 3 新光三越的櫃位配置
4 台積電的虛擬量測技術 5 鞋廠的成品瑕疵辨識

考 大數據分析失敗的原因
- case 1: XX 醫院的”超大”數據分析⇒ 資料集沒有全部狀況
- case 2: 第一屆高速公路數據競賽的慘劇⇒ 資料集沒有全部狀況
- case 3: 某某花園的許願籤詩分析案⇒ 沒有資料就沒有辦法做分析(資料是詩籤本籤)
- case 4: 塑膠射出機廠商想做良率改善⇒ 題目定義不準確
- case 5: 國家太空中心的太陽能板分析案⇒ 有資料，但資料精度不足
- case 6: 資料分析廠商用深度學習模型開發出的刀具磨耗預測系統沒人要買**
⇒ 沒有考慮到題目定義不合理

考 統計、機器學習、深度學習 在不同資料量下的效能：
1.資料量 200~500 筆用 統計 分析→由人類知識完全主導的方法
2.資料量 1000~10000 筆用 機器學習 →人類知識給予方向(架構)，再由機器學習來調整參數
3.超過 10000 筆時 大數據分析(deep learning) →由機器學習決定架構並調整參數是最好的



大數據分析的執行流程：
1.定義：沒有題目就無法分析
2.收集資料：確定資料是否符合預期
3.清洗資料：最關鍵，最耗時，最痛苦
4.取特徵值：全部拿來用
5.降維：直接用套件
6.分類、建模：要拿高薪要會改套件程式碼
7.報告製作：稱為 AI 落地，寫執行的 SOP，做防呆，以及品質檢測，並-做成餐廳能用的網頁

案例 1: 將全班分成男生與女生
定義目標案例實作環境。(落地，資料…etc.)
收集資料: 對全班同學做問卷。(200)
清洗資料: 去除或修正問卷內不合理的資料。
1. 不合理: 體重 150 公斤，身高 250 公分。
2. 缺值: 沒有填字。
3. 資料分布: 男女生比例(資料擴增)，是否是高斯關聯。
取特徵值: 由人從問卷中找出適合分辨的欄位。(50)
降維: 由電腦與既有紀錄找出適合分辨的欄位。
-->由歷史紀錄去做，不同資料及答案不同
分類: 利用既有資料建立分類模型
-->左半邊: 頭髮長度 右半邊:身高，體重 ⇒ 決策樹
建模: 不適用
分析及製作報告: 撰寫執行 SOP 文件與解釋。

2. 利用大數據分析降低學生餐廳廚餘量」(用分類技術來提升分析準確度)
分析案之每個分析流程要做什麼？
定義題目：再分成
1 廚餘：要分實剩的，沒吃完的 2 重量：再分含水、不含水 3 體積
收集資料：
天氣、節慶、行事曆、期中、期末、科系人數、男女生人數、各種族人數
(分類技術) 早，中，晚，節慶，放假各建一個模型
→取特徵: 早 8，早 9 的人數
清洗資料：把不平衡的去掉
台灣天氣是晴天雨天，美國要考量是否下雪，節慶是不平衡，餐廳的訂貨量單位，
取特徵值：不同國家學生點餐行為不同
降維：t-SNE、PCA

分類 建模：NN，對不同食物建立不同模型

課程活動 2. 目前台灣已有許多大數據案例被公布出來，請試著上網
找尋工業與商業各一個案例，並簡略說明他們的「題目」、「資料集」為何?

1.工業大數據案例：
台灣的博睿科技在智慧製造領域中運用了大數據來實現工廠自動化，進而邁向智慧化工廠。他們的解決方案基於 Google Cloud，結合了工業電腦（IPC）和工廠機台溝通獲得的數據，並提供能源監控系統來提升生產效能。該方案使用了分散式架構來處理工廠數據，從而幫助傳統工廠轉型為智慧工廠(Cloud Ace 雲一有限公司)。

商業大數據案例：
在商業領域，國立中山大學開發了一個商業大數據平台，分析商業交易與客戶行為數據。該平台利用如 R 語言的分析工具來進行資料視覺化、預測模型建構及社會網路分析，並透過機器學習技術挖掘商業數據中的潛在模式和趨勢
(國立中山大學 商業大數據平台)。

課程活動 3. 請每組提出想做的大數據分析題目 10 個
1. 預測產品的市場需求變化。
2. 分析社群媒體情感趨勢以預測品牌聲譽。
3. 使用交通數據進行智慧城市交通流量優化。
4. 預測股票市場波動並進行投資風險管理。
5. 分析電商顧客行為來提升個性化推薦系統。
6. 分析醫療數據來預測疾病的爆發與流行趨勢。
7. 利用氣象數據來改善農業生產力。
8. 分析零售業的銷售數據，預測最佳庫存管理策略。
9. 預測能源消耗趨勢以實現智能能源管理。
10.分析工業製造流程數據，提升設備維護效率。

AI 簡單來說就是用電腦 => 來學習輸入輸出間的規則
1 演繹、推理和解決問題：會根據過往歷史紀錄，進行推理思考
2 知識表示：從儲存的知識，並且推理演繹得到新的知識。
3 學習：從使用者和輸入資料，去判斷和輸出結果。
機器學習是對能通過經驗自動改進的電腦演算法
案例 1：讓電腦學會寫 8 是先初始化畫出一條線，然後調整出最佳解
案例 2：校正大富翁的遊戲參數 (土地價格)，讓破產率降低
最後證實打錯價格，確實會造成破產機率提升

類神經網路
概念一：他是模擬人類的神經元所設計而成的。
概念二：他具有學習的功能，其學習的流程就像你做完考卷後，
會利用訂正的過程來讓自己學會考題上的知識
概念三：他是個黑盒子，你不需要了解他的內部操作就可以使用他，直接用套件
也因為這樣 **可解釋化 AI** 和 了解 **AI 內部核心原理** 變得很重要
AI 裡的機器學習 又分成 淺層類神經網路 、 深度學習
淺層類神經網路：只有短短少少的幾層神經元
並且 受限於資料前處理結果 連論文都在討論資料前處理演算法
深度學習：硬體進步可以訓練更深的網路，更多的參數，更多的節點 (Alpha Go)
並且 模型套件的出現，又再度降低了門檻
深度學習的分支
1. 套件應用: 用套件解決各項問題,價值低,**成本高**
(一定得有 GPU 才能將訓練好的模型套在各種應用中)
2. 核心價值發揮: 從訓練好的深度學習模型挖掘各種資訊
,提高分析效能及降低使用成本
DL 專用的 GPU
1. 先分析 DL 會用的參數，函數有哪些
2. 替這些函數設計成 專用硬體
3. 加快速度

	離散	時序	時空
降維	LDA、PCA	FNN	
分群分類	KNN、 K-means	K-medoids	DBSCAN
建模	DNN	RNN、LSTM	CNN+LSTM

以實務面來看大數據分析跟 AI 關係
(低) Level 1：改變輸入，輸出型態來達成建模目標
(高) Level 2：把類神經拆解成多個小 block 看，再組合起來

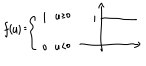
Level 1：改變輸入，輸出型態來達成建模目標【時序 -> 圖 -> CNN】
Level 2：把類神經拆解成多個小 block 看，再組合起來
Level 3：不能用套件的方式去設計符合環境限制，成本，size【IC ->硬體 -> model】

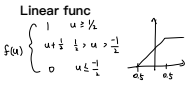
	資料前處理	使用模型
處理時間	超過 1/2 的計畫時間	頂多 1/2 的計畫時間
遇到問題	沒有正解，靠經驗	都套模，很難出錯

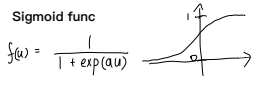
Perception:只有單一層神經元(最早的類神經) 原始是為分類而設計
Convolution 卷積層 就是在做升維度
Pooling 就是在做降維度

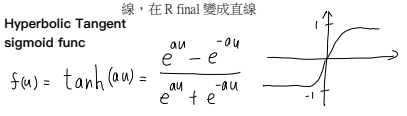
Regression 回歸 就是在一堆資料點 找一條最符合的 回歸線

類神經一定要有 bias 偏壓值 W0
才能加快運算，幫助平移 y=WX+W0,W0(座標軸位移)W(等比例壓縮或放大)
類神經物理觀點 1:轉置矩陣(空間映射)
前面 幾層轉置 (升維度) 找好處理的面，後面就可做分類(降維)動作
類神經物理觀點 2:一個神經元，就代表一條線
activation 激活函數:
線性:直線 非線性:曲線
深度:神經元越多 ==> 線越多 => overfitting
回歸:會去算資料分部有多少趨勢來決定神經元

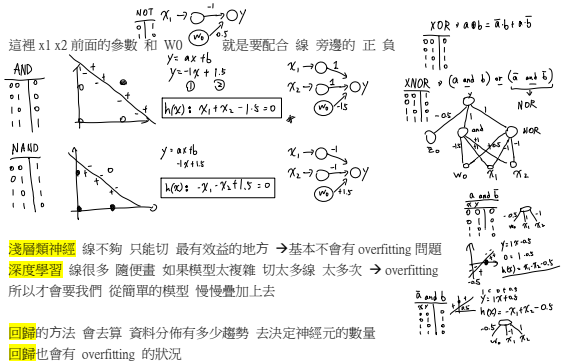








這裡 x1 x2 前面的參數 和 W0 就是要配合 線 旁邊的 正 負



淺層類神經 線不夠 只能切 最有效益的地方 →基本不會有 overfitting 問題
深度學習 線很多 隨便畫 如果模型太複雜 切太多線 太多次 → overfitting
所以才會要我們 從簡單的模型 慢慢疊加上去

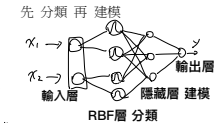
回歸的方法 會去算 資料分佈有多少趨勢 去決定神經元的數量
回歸也會有 overfitting 的狀況
如果做到後面 還一直做 就會連 雜訊的資料 也想要一起回歸線
所以會用early stop 提早結束 來解決最後的 overfitting
對回歸問題來說，最後一層用 線性相加 通常可獲得比用其他非線性 activation 更好的效果

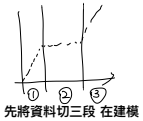
類神經架構 可調整的 五個部分:
1.連結的型態
傳統類神經:fully connected v.s. partially connected
深度學習模型:在 Deep Learning 中用 partially connective 是無效的
=>gpu 無法處理稀疏矩陣，所以實務上缺值處全補上 0
Why 淺層類神經 會有 partially connected?
DL 是不知道關聯，找關係，淺層通常是已知關係(公式)去做設計。
fully 容易過擬合，需正則化處理 Partially 較不容易過擬合
補充:不一定要連到下一層,也可以連其他層 (例如 ResNet 的連線)
類神經的設計很彈性，只要能說出物理意義都 ok
ResNet:
1.目前 CNN 為了克服梯度消失而設計(梯度消失會讓前面影響力不見,所以拉前面原始資料來補)
2.隱藏層神經元數目
MSE 對應隱藏層神經元數量折線圖:轉折點(最佳解) 在 train 時 要有轉折點
只有一條線向下的 最好繼續做下去 繼續去找到 local 或 global minimun
3.Activation Function *考*universal approximation
考 類神經的 universal approximation 定理
→所使用的架構與 activation function 能逼近任何函數
激活函數 通常都只用中間斜線 來回歸出 我們要的線
Why 不會用線性 activation ?
Ans:類神經的 universal approximation 定理(任何函數都可逼近) p.s.:如果是做迴歸 or 時序預測的話,輸出層用線性的 activation function 效果較好
只要擁有足夠的隱藏神經元數量，並使用非線性激活函數（如 ReLU 或 Sigmoid），則一個單隱藏層的前饋神經網路可以近似任何連續函數。
一層的神經網路就能近似，但需要足夠多的神經元。
無限多個線性函數（會是類神經一部份,但不會整個類神經都是線性函數）線性相加時,可逼近任何函數嗎?No,因為現實中大部分問題都是非線性的 Tangent sigmoid:具有 universal approximation 特性=>以 tangent sigmoid 為主題設計類神經,
定理:當有無限多個 tangent sigmoid 線性相加時,必然能逼近任何函數

Sigmoid Function：會把輸入變成 0 到 1 之間
Tangent Sigmoid ：會把輸入變成 -1 到 1 之間
為什麼 Tangent Sigmoid 會比 Sigmoid 表現還要好?
Ans:1.因為 有正 有負 才更能表達所有狀況 這樣的話 就不會限制到 w 的發展
只有 0 到 1 要相減時 只能是一個 一正 一負 的限制 w
Ans:2.類神經訓練目標 就是 weight 值的最佳解
=>所以我們習慣讓 weight 可操作範圍越大越好
=>能給的操作範圍 tangent sigmoid > sigmoid
Why 需要 Relu? CNN 前半段的重點就是判斷圖內是否有某物(Relu 的作用)
Relu 可以加強 經過某個 conv 的特徵提取後 的特徵加強
ReLU 更能避免梯度消失問題，因為 ReLU 在正區間的斜率是 1
不會像 sigmoid 那樣在數值較大或較小時趨向於零，導致梯度難以傳遞。
Why leaky Relu? Relu 的缺點(與倒傳遞有關=>透過微分結果覺得 weight 變動)
Ans 1：為了微分用 負數的地方才能有斜率 繼續慢慢走下去
不然都是平的 0 會動不了，
GPT Ans ： 避免「死亡 ReLU」問題：ReLU 在負區間輸出為零，而 Leaky ReLU 對負值給予一個小的斜率（例如：0.01），避免了長期為負的神經元完全失去作用，增加了激活的可能性。

NN 的四大題型 迴歸,時序分析 適合用=> tangent sigmoid 或 sigmoid
分類,分群 適合用=> Relu 或 leaky Relu

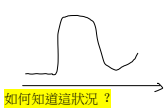
RBF 函數 (高斯函數) Radial Basis Function
沒有能讓 NN 達成 universed approximation 這定理 的特性
RBF 在類神經裡的物理意義：做 分類 的動作
因為它主要是在做 分類 ，並且無法逼近任何函數 →只會放在 類神經 最前面
例如： RBF-NN 、 Fuzzy neural network
先 分類 再 建模

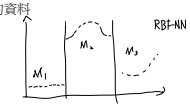




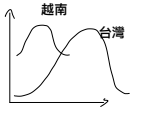
可以先將資料分成三類 再建模會比較簡單 等同於建三個線性模型
NN 是只找一條線去逼近資料做建模 等同於單純建一個非線性模型

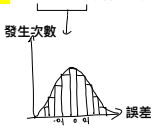
RBF-NN 適用狀況：資料不同區段差異大時
例如：工廠 物理 化學領域的資料



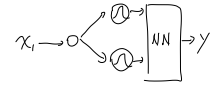


如何知道這狀況 ?
1.資料觀察
2.套最簡單模型後看效果
(發現資料跳躍處效果不好，就將原本的 NN 改成 RBF-NN 去做)
以資料分佈來看 越南 和 台灣 兩筆資料一起做 預測效果會很差
輸出：NN 的 Error histogram 呈現高斯分佈才算訓練成功

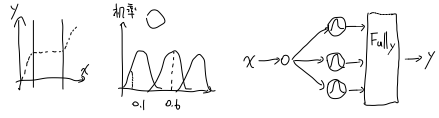




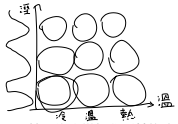
當資料出現多個高斯分佈組合時，用RBF-NN 來建模可提昇準確度
如越南 與 台灣的資料分佈
這種RBF-NN最少用兩個

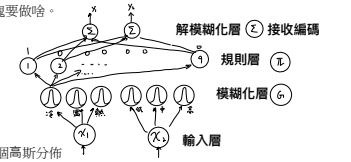


並且可以從輸入來找出要激活 全連接層相對應的部分

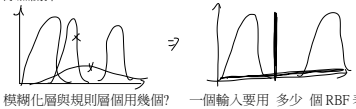


Fuzzy logic 的概念: 精確 : 25.2℃ (25.5℃, 25.6℃) 模糊 : 好熱 (25.5℃, 25.6℃ 是一樣的)
模糊設計的缺點：數學很難，叫 fuzzy 數學，後來用類神經網路求解，叫模糊類神經
會經過這三個動作：
1.模糊化：例如 35 度代表熱。將溫度跟濕度各轉換成三個高斯函數。
2.規則化：分割區塊。把溫度跟濕度所有可能發生的狀況分成九種規則。
3.解模糊化：告訴我每個區塊要做啥。





輸入進去模糊化層轉換成多個高斯分佈
到了規則層會輸出編碼，例如圖上的 1 0 0 0 0 0 0 0 0 而這些編碼會
啟動全連接層中相對應的部分。最後輸出控制器應該要做哪些動作，例如冷媒關掉、除濕關掉。



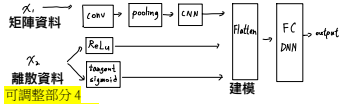
模糊化層與規則層個用幾個? 一個輸入要用 多少 個 RBF 表示？ 區分幾個區塊 ？
決定方法：
1. 由 Fuzzy 數學計算(超難)
2. 用多一點神經元，訓練中無用神經員會自己關掉
無用的話在過程中會越來越小ex.在過程中會變成很窄高的線或扁平的線，在 R final 變成直線

Discussion: 模糊化與規則層間的連接方式
老師目前教:部分連接實務上全連接?
已從 Fuzzy 得知要切 9 個區塊及其物理意義才能這樣做
不會知道 Fuzzy 切區塊方式，所以電腦自己調整，不是她區塊的規則會自己把 wrights 接近 0

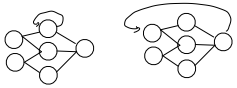
實務上 模糊邏輯 與 fuzzy-NN 關係的運用
1.商品 硬體：規則簡單。fuzzy logic（因為他的規則好用於硬體實現）
家電商品常用fuzzy 實現
2.但 fuzzy 數學算規則層超難
3.NN 出來後，出現先訓練 fuzzy-NN 再拆解訓練後 fuzzy-NN 的參數
把他對回去 fuzzy 理論(用硬體實現）

目前大數據時代做法
1.出廠時有一套 fuzzy logic 在控制器上（參數是出廠參數，實驗室做的）
2.你用一段時間後，有你 或 者全國使用者的參數。先訓練 fuzzy-NN 得最佳參數，透過網路 update 硬體的 fuzzy logic 參數

小波類神經 wavelet neural network
同一個類神經中，使用不同的 激活函數
現在的 DL ： add(… , Relu) 就是這一整層都在做 Relu
用一堆 func 湊出一條線 → 雖然能做出來 但花太多 func
所以用 複合多函數 來減少神經元數量



可調整部分 4
內遞迴、外遞迴



為何需要回饋？ANN 與 RNN 的比較，以手寫 8 為例
他不符合 類神經的規範
因爲類神經 是要推導出一個 一對一 或 多對一 的線性函數
在這地方 可以走的地方有兩個 變成 一對多
一個輸入 會有兩種輸出 就會錯 一個商品有兩個價錢
硬要用ANN 去做的話
這有可能 她會走兩個路的中間 但也有可能是隨機一條路

這個的解決方法 加上先前一個的資料
這就是外遞迴
但這個外遞迴 也有缺點 他的誤差會累積
模型一下就爆了 發散 輸出就直接變成 無限大 或 0 NaN

原 (0,0) –>(1,1)
(0,0) –>(1,–1)
遞迴加上以前資料
(–1,–1)–>(0,0)–>(1,1)
(–1,1) –>(0,0)–>(1,–1)
變成一對一函數
就可用NN訓練

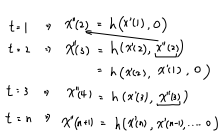
解決方法 把外遞迴 變成 內遞迴
這樣 輸出 的誤差累積 只會有一個人在影響
也是會 發散 但是比較慢

這樣子 內遞迴 物理意義 也變了
外：
拿輸出結果去 從頭去更新 去預測
包含很多無用資訊
內：
只對某個特徵節點 自己做更新 並再預測
這樣子的資料比較乾淨 (完全是 特徵節點 所需的資料)
這就是為什麼 內遞迴比較好

內遞迴 用公式來看
以前的所有資料 都會影響到下一個輸出
但 越遠的 影響越弱 有種 剃度消失 的感覺

內預測 不是所有情況都適用
因為 要預測星期二 早八 Ubike用量 而要用的資料 不是所有時間都影響
反而是和 某幾個時間點 前幾個時間段 更有相關
用到 沒幫助的資料 準確率也會下降

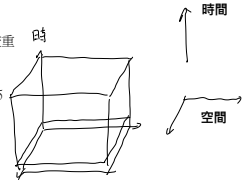
公定符號 資料進來之後
晚一個step再傳



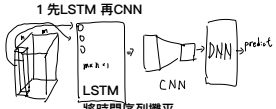
這種狀況 處理方法
1 取出真正有影響的資料 用一般的 NN 建模
這方法是一種 時序資料 的 降維 處理
2 用LSTM block
與 內遞迴 相同 但有加上 3 個 遺忘閘 Model 不同於 block
輸入遺忘閘 忘掉多少之前的輸入 Model 會有 去特徵值 神經節點 建模分類
記憶遺忘閘 忘掉多少 M 裡面暫存的記憶 Block 會有一層時序 一層時序
輸出遺忘閘 看輸出 要影響多少後面的層 的處理層的概念
但他的缺點就是
1 參數很多
2 訓練很久
3 而且這些也不確定要用多少 可能是隨機用
4 本質上還是一個 回歸類神經
還是有誤差累積 而發散的問題
3
所以 不想用 取特徵值 和 遞迴 就不會有發散問題
就是把它 暴力攤平

讓他自己去找 哪些資料更有用 DNN 就留下他的比重 沒用的 比重就給他 0

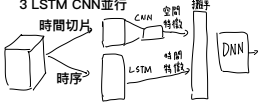
一般學生 跑模擬 時常犯的錯
注意跑離散資料 通常得先做 洗牌 的動作 shuffle
因為在資料庫 可能會先有一些排序了
這樣直接訓練 可能前面的資料會忘記
而 後面的資料 在模型上的影響 會比較重

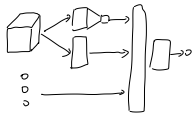


LSTM 物理意義就是 去取 時間特徵
這個問題就是 去預測 下個時間的 pm2.5
解題思路 邏輯
1 這同時會有 時間 混雜 空間 的資料
2 時間 用遞迴去解
空間 用cnn
結合這兩個去解
3 誰先做？
1 2
1 先LSTM 再CNN
2 先CNN 再LSTM

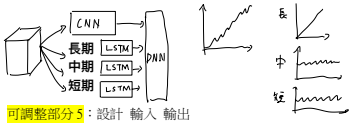


將時間序列攤平
3 4
3 LSTM CNN並行
4 就是3 再加上 外在因子



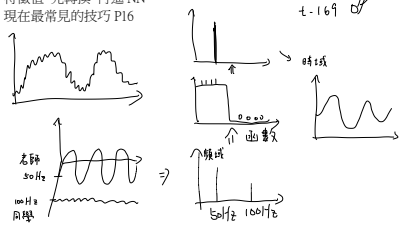


5 長期 中期 短期 資料



可調整部分 5：設計 輸入 輸出
傳統 RNN 可能會讓遠久的資料資訊不見的問題
(太久以前資料無法影響目前結果)
攤平之後，雖然網路無法記憶過去時間資訊，
但可確保過去有用資訊被納入

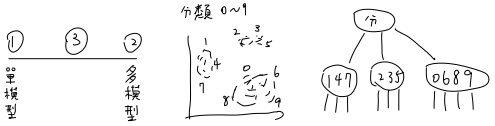
特徵值 先轉換 再進 NN
現在最常見的技巧 P16



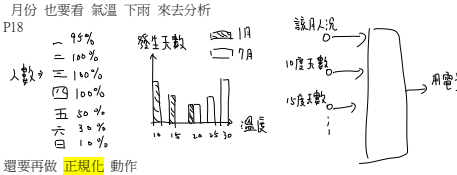
輸出型態改變
分 0 ~ 9 通常我們就是丟給機器 讓他幫我們分
然後取 最高機率 的那個當作答案
不準 常出錯 有 offset 造成的影響而有誤差 因為輸出的多維度互相拉扯造成

解決方法
一 最現代的做法 用softmax
模型最後一層導入 softmax(強迫所有輸出只有一個獨大，其他很小)來避免
二 拆 model MIMO 拆成 MISO
目前角度看 softmax 可取代此，但分類問題題

三 是二的延伸 用 樹狀結構
他是介於 單一模型 和 多模型 中間
先大分類器 在用 小區域分類器

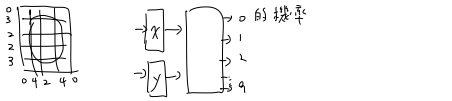


7 個實務問題
1 宿舍電量的預估
電量預測 用回歸模型 DNN 人數 月份數 都是固定的 只用這兩個直接測不準
所以要做轉換：
每個禮拜的 每個星期幾 分別有多少人
月份 也要看 氣溫 下雨 來去分析
P18

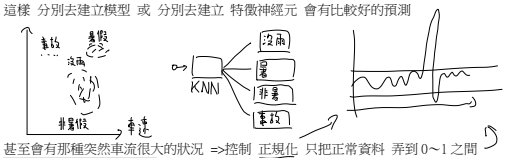


還要再做 正規化 動作
把輸入 輸出 調整到 0~1 的範圍內 不然 單位 和 範圍差太多
類神經總評估會失準 一個只有幾百人 一個有好幾萬瓦特
只用整數去拉大 會有很多到不了的地方 要用的是用區間 來去判別

2 0 到 9 的手寫辨識
1： 現在直接用 CNN 去處理
2： 提供另外一種解決思維(NN 重點再觀察資料，然後把對分析狀況最有效的特徵丟入 NN 中)
用 2：把它分別壓平 累積 到邊邊用這兩個去建模

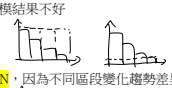


3 高速公路的車流量預估
但如果有做資料觀察的話 會發現他資料分很開 可以再分成
是不是暑假 有沒有下雨 有沒有事故 夜晚 白天 上班時間
這樣的車速 車流量的會有分別
這樣 分別去建立模型 或 分別去建立 特徵神經元 會有比較好的預測

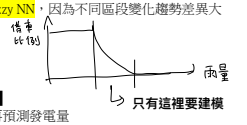


甚至會有那種突然車流很大的狀況 =>控制 正規化 只把正常資料 弄到 0~1 之間
5. Ubike 站點車流量的預估
在 大流量 的地方 是可預估的 在 小流量 的地方 是不可預估的 0~2 台
只要有多一個人去借 就會打破規律 這樣的資料比較像 隨機資料

6. Ubike 站點車流量受到雨量的影響
初學者 ⇒ 禮拜一下雨租借量和過去 52 個禮拜一平均的租借量，算差距
如此做問題:下雨天數不多，資料不平衡，建模結果不好
需要做資料平衡
1. 用 SMOTE 達成向上取樣
2. 用選基選幾個 0 雨量狀況做向下取樣索
如果一定得全部資料下去建模 ⇒ 用 Fuzzy NN，因為不同區段變化趨勢差異大

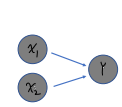


太陽能發電量預估
1. 從過去發電量時間序列預測 ✕
2. 從當下天氣狀況(感測器資料)評估 ☑
3. 天空養向+風向 ⇒ 先預測雲走向，再預測發電量



只有這裡要建模

HW1 1. 請畫出能實現 Nand gate 的 Perceptron
y = step_function(w1*x1 + w2*x2 + b)，設置 w1 = -2, w2 = -2, 和 b = 3



2. 請寫出分類、迴歸、時間序列分析的應用各兩種
分類 i. 電子郵件垃圾過濾：使用分類模型來區分垃圾郵件和合法郵件，以自動過濾電子郵件中的垃圾內容。
ii. 醫學診斷：利用分類模型來預測患者是否患有特定疾病，例如使用患者的健康數據來預測癌症、糖尿病等。
迴歸 i. 房價預測：使用迴歸模型來預測房屋價格，基於特徵如地區、面積、房屋類型、房齡等。
ii. 股票價格預測：利用迴歸模型預測股票價格，根據歷史數據、公司業績、經濟指標等特徵。
時間序列分析 i. 天氣預報：使用時間序列模型來預測未來的天氣情況，包括溫度、降雨量、風速等。
ii. 設備故障預測：使用時間序列模型來預測工業設備的故障，基於傳感器數據、歷史性能數據等。

3. 請用氣溫的預測為例說明 ANN 與 RNN 的不同
ANN 只用過去幾天來預測，可能無法充分捕捉到更長時間範圍內的趨勢或季節性。
RNN 可以將整個歷史序列作為輸入，可捕捉潛在趨勢和依賴關係
可處理不同長度的輸入
固定輸入、不用關注長期趨勢 =>用 ANN 反之 則用 RNN

4 假設我們目前正在做太陽能面板的發電量預測，且已知以下兩個專業知識：(1)這個面板的發電量跟前一個時間的發電量無關。 (2)發電量與日照量呈接近線性的關係，請問這個問題該用什麼類神經解決呢？ ANN 即可
5. 假設我們目前正在做太陽能面板的發電量預測，且已知以下兩個專業知識：(1)這個面板的發電量跟前一個時間的發電量高度相關。 (2)發電量與日照量的關係有三階段的關係，在日照量小於 20 時，幾乎不發電，在 20 到 80 區間呈線性關係，在 80 以上時，呈指數降低，請問這個問題該用什麼類神經解決呢？ RNN 或 LSTM 並希望進入神經元前先用 RBF 分類資料

NN 的四大題型以及分別用甚麼 acviation function? 迴歸,時序分析=>tangent sigmoid or sigmoid，分類,分群 =>Relu or leaky Relu 實務上 tangent sigmoid+leaky Relu+Softmax 以 f(M1,M2,M3)數值表現做分類
客製化類神經案例通常會如何做?由 RBF 函數實現模糊邏輯,並形成一個模糊類神經
單維 RBF 及多維的差異
單維 RBF(只在一個變量上運行):類神經的致命傷:X 的每個小變動都是有意義的,但常常變動只是來自於誤差,所以想考慮小變動.單維 RBF 無法忽略輸入的微小波動，容易受到噪音影響，
多維 RBF:多維情況下可用來分類(RBF 類神經適合處理需要先對輸入資料分群再處理的案例

大數據分析的各個流程&關鍵:
1.定義題目(找出目標問題的輸入與輸出，並判斷解題類型):沒有題目就不可能
分析、定義題目請從簡單到難、與第一線人員而非主管對談★只要輸入輸出訂好，問題就了一半 2.收集資料(收集所有有關的資料集,因為有給電腦判斷規則):確定資料是否可用，如視覺化，或了解資料收集方式、確定資料分佈跟題目所想是否一樣★大數據分析需要把所有資料丟給電腦由電腦判斷規則 3.清洗資料(觀察資料、找出與修正不合理,缺值的資料、合併資料欄位):大數據
分析最關鍵地方、技術不難，會 if、for、while 就會做但他是大數據
分析最耗時、最痛苦的地方★為何需要?把資料集修正符合所需要的,
有關的資料會分佈在不同資料集,需要合併到同一張 table 才能讓模型訓練
4.取特徵值(早期由人工找出對資料分析有用的欄位,近期更改資料欄位型態提升辨識準確率): 盡可能把所有想到的都取出來
5.降維(由「電腦」及「資料集」找出對資料分析有用欄位): 直接用套件 6.建模(建立起輸入輸出間的數學函數):直接用套件 or 改套件程式碼 7.報告製作(按照廠商的需求製作分析說明書 ex.方法 sop、落地、量化、質 化、專家意見)/知道廠商的終極目標並針對它製作成果報告
(分類:可放任何地,將未知資料分到多個定義好的群組中)

生產管理、設施規劃、作業研究、演算法
由專業知識與經驗設計方法，優:不需要事先收集資料. 缺:常發生沒有考慮到的地方
未來走向:由既有歷史資料與大數據分析技術重新建構專業知識
事後分析:小數據(統計,品質管理,機器學習), 大數據(大數據分析) 不需要大量知識就可以設計方法
優:只要數據夠大，就不會思考不周. 缺:需要事先收集資料與 Labeling
未來走向: 由既有專業知識加快數據分析流程與分析準度→算是廣義的 Transfer Learning: pretrain 模型加上額外訓練

大數據分析為何效果比小數據方法好的原因：
1.小數據容易因樣本不足而導致偏差或分類錯誤的結果。
2.由於數據量龐大，可以從大量數據中自動提取規則，使得預測和分類的準確度顯著提升。

台積電案例，分析案例的各個分析流程步驟：
收集資料： 在機台上安裝許多感測器收集資料
(總共 200 個製程，每個製程收集 200 種資料) 清洗資料： 清除資料中不合理的資料
取特徵值： 找出 40000 種資料中，可能具有代表性者
降維： 找出所有特徵值中，對預測損壞最有效者
建模:利用類神經網路、迴歸模型等建立模型
實際分析流程(通常分三輪實作)
? 第一輪 (必然是失敗的但可以當做分析基準) 利用最簡易套件了解資料分佈(資料不平衡 or 缺值)、題目的效能極限，設計第
二三輪大方向
? 第二輪 (失敗機率 50%) 依據經驗重新清洗資料、利用大方向中的各項套件進行分析，並產生初步成果、觀察結果是否還有需要改進的地方
? 第三輪 (失敗機率 80%) 依據上一輪認定該改進的地方進行資料或演算法調整、通常需要修改套件內的程式碼、成功的分析，其目標是效能比分析基準好 30%或符合廠商需求

案例 3：「台積電虛擬量測技術」在每一輪分析時會遇到什麼困難？
第一輪：資料不完整或噪音大，可能有關鍵製程數據遺失，或設備監測數據出現異常。初步分析的效果通常會較差，無法達到生產需求標準。
第二輪： 可能涉及去除噪音數據、補齊缺失值、或聚焦於特定製程步驟的數據。
第三輪：細化數據集，針對特定製程參數進行深入分析，並可能需要修改或優化模型代碼（如自訂損失函數、調整架構）。