

人工智慧簡介

胡念祖

虎尾科技大學 資訊管理系

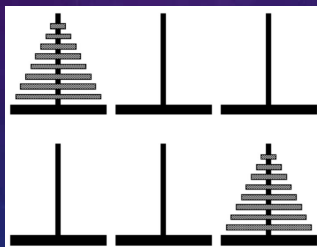
2017/11/1

大綱

- 人工智慧發展沿革
- 常見演算法介紹與演練
 - 線性迴歸
 - 決策樹
 - 類神經
 - 群集分析
- 深度學習
- 實務案例介紹

人工智慧發展 (HTTPS://WWW.INSIDE.COM.TW/2017/07/10/AI-HISTORY)

- 1945年，第一部通用電腦被開發出來
- 1956年，**人工智慧**在Dartmouth的研討會中被提出
- 河內塔問題



- 需5849億年，但宇宙只有137億年
- 1970年，受限於硬體與系統的限制，造成第一次人工智慧泡沫化

人工智慧發展 (HTTPS://WWW.INSIDE.COM.TW/2017/07/10/AI-HISTORY)

- 1965年，Intel創辦人提出摩爾定律
- 1980年，提出「**機器學習**」，是實現「人工智慧」的方式之一
 - 1980年，諾貝爾醫學獎得主研究大腦內部的神經迴路
 - 1986年，Hinton和Rumelhar提出反向傳播，無法超過3層
 - 1995年，Vapnik提出支援向量機(SVM)
- 2010年，由於網路的發達，提出了「雲端運算」
- 2012年，**大量社群媒體**與**電子產品技術**提升，發展出大數據與物聯網(IoT)
- 2013年7月，Google收購一家只有三人的公司(**DNNresearch**)
 - 2006年，Hinton成功訓練多層神經網路，深度學習
 - 2012年，Hinton的二位學生用(深度學習 + GPU) 參加ImageNet競賽 (16.42% vs. 26.22%),
 - 2015 Microsoft 3.5%, Human 5%
- 2016年—2017年，**AlphaGo**

ALPHAGO

- 2014年，啟動ALPHAGO計畫，運用分散式運算（1202顆CPU與176片GPU）
- 2015年10月，擊敗樊麾，第一個無需讓子擊敗職業棋士
- 2016年3月，透過自我對奕來強化，擊敗李世乭。(ALPHAGO LEE)
- 2016年12月，化身MASTER挑戰中日韓高手，60戰全勝
- 2017年5月，運用GOOGLE TPU擊敗柯潔。(ALPHAGO MASTER)
 - 運算資源為2016年的1/10，單機版
- 2017/10月，ALPHAGO ZERO（3天，21天，100：0）
 - 無需人類指導，使用新的強化學習方式

GPU ??

- CPU (Central Processing Unit)
 - 程式碼>編譯>執行
 - 適合高速、邏輯成份高的工作，平行度低，context switch (OS)
- GPU (Graphics Processing Unit)
 - 2D/3D 光影、塗色、影像處理
 - 適合同時大量簡單，計算密度高的工作,SIMD
 - 2006-2007，CUDA (Compute Unified Device Architecture)
- GPGPU (General-Purpose computing on GPU)
 - 氣象模擬、分子模型、物理或化學研究

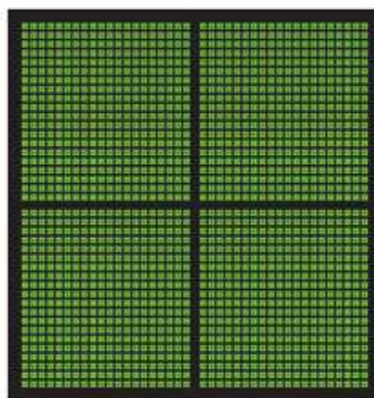
求質數

```
int i, j, k;  
for (i = 2; i <= 1000; i++)  
{  
    k = 1;  
    for (j = 2; j < i; j++)  
    {  
        if (i % j == 0)  
        {  
            k = 0;  
            break;  
        }  
    }  
    if (k == 1)  
    {  
        Console.Write(" {0:D3}", i);  
        Console.WriteLine();  
    }  
}  
Console.ReadLine();
```

GPU (GRAPHIC PROCESS UNIT)

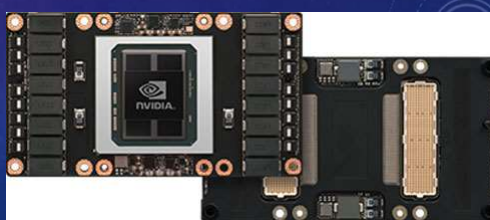


CPU
MULTIPLE CORES



GPU
THOUSANDS OF CORES

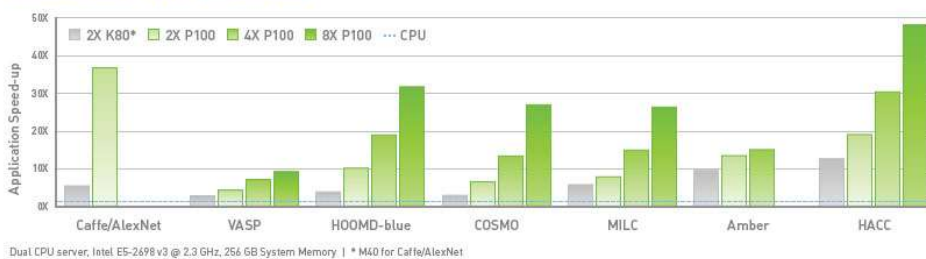
NVIDIA K80, P100, ...



CPU，GPU 效能比較

- 940mx (384cores 953.9 GFLOPS)
- K80 (4992 cores, 2910 GFLOPS)
- P100 (3574cores, 4670 GFLOPS)

NVIDIA Tesla P100 Performance



TPU (TENSOR PROCESSING UNIT)

- 15-30x (Intel Haswell CPU, Nvidia Tesla K80 GPU)



Model	Die										Benchmarked Servers				
	mm ²	nm	MHz	TDP	Measured		TOPS/s		GB/s	On-Chip Memory	Dies	DRAM Size	TDP	Measured	
Haswell E5-2699 v3	662	22	2300	145W	41W	145W	2.6	1.3	51	51 MiB	2	256 GiB	504W	159W	455W
NVIDIA K80 (2 dies/card)	561	28	560	150W	25W	98W	--	2.8	160	8 MiB	8	256 GiB (host) + 12 GiB x 8	1838W	357W	991W
TPU	NA*	28	700	75W	28W	40W	92	--	34	28 MiB	4	256 GiB (host) + 8 GiB x 4	861W	290W	384W

Table 2. Benchmarked servers use Haswell CPUs, K80 GPUs, and TPUs. Haswell has 18 cores, and the K80 has 13 SMX processors. Figure 10 has measured power. The low-power TPU allows for better rack-level density than the high-power GPU. The 8 GiB DRAM per TPU is Weight Memory. GPU Boost mode is not used (Sec. 8). SECDEC and no Boost mode reduce K80 bandwidth from 240 to 160. No Boost mode and single die vs. dual die performance reduces K80 peak TOPS from 8.7 to 2.8. (*The TPU die is \leq half the Haswell die size.)

人工智慧 VS. 機器學習 VS. 深度學習 (NVIDIA)

ARTIFICIAL INTELLIGENCE

Early artificial intelligence stirs excitement.



MACHINE LEARNING

Machine learning begins to flourish.



DEEP LEARNING

Deep learning breakthroughs drive AI boom.



1950's

1960's

1970's

1980's

1990's

2000's

2010's

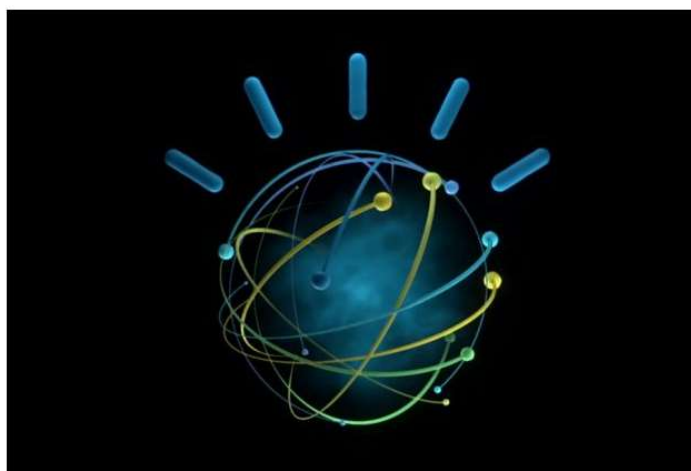
Google 旗下人工智慧公司 **DeepMind** 昨 (30 日) 宣布 與英國生物醫學重鎮

「倫敦大學學院醫院」(University College London Hospital, UCLH) 合作, 探討如何將人工智慧用在治療頭部、頸部癌症 (如口腔癌、咽喉癌、鼻咽癌等) 病人, 希望開發出自動識別癌細胞的放射治療儀器。



沒想到救了她一命的，居然是 IBM 的「華生」

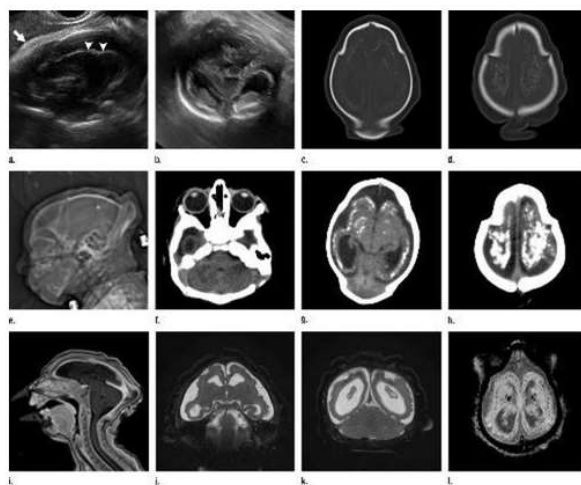
作者 龔范兒 | 發布日期 2016 年 08 月 09 日 7:30 | 分類 AI 人工智慧, 醫療科技 [Follow](#) [G+](#) [讚 1,250](#) [分享](#)



誰是IBM「華生」？

- IBM 人工智慧 Watson 利用 10 分鐘時間診斷出一名 60 歲女病人患上罕見的急性骨髓性白血病，還找到最適合她的療法，目前患者已經可以出院
- 將病人的**基因變化**與 **2,000 萬篇癌症研究論文數據庫**進行比較，提供準確的診斷並且提出先進且適合的治療方案
- 運用技術：
 - NLP (Natural Language Processing)
 - Data Mining
 - Image Processing

目前，放射科醫師都採取電腦斷層掃描（computed tomography, CT）或核磁共振造影術（Magnetic Resonance Imaging, MRI）去找出病人癌細胞、再畫出區域施以放射治療。由於頭部、頸部細胞組織格外複雜，過去完成 1 次掃描治療可能就要 4 小時，Google 相信使用人工智慧可將時間壓縮在 1 小時內。



常見演算法介紹與演練

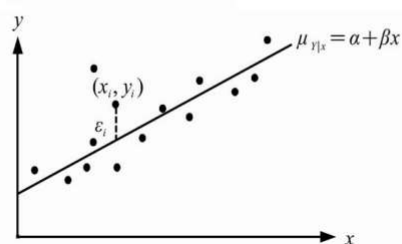
- 實現人工智慧的重要方法－機器學習
 - Linear Regression
 - Decision trees
 - Neural networks
 - Cluster
 - Instances
 - Graphical models (Bayes/Markov nets)
 - Model ensembles
 - Etc.

INSTALL

- Anaconda2 & 3
- Seaborn
- Pydot
- Graphviz

LINEAR REGRESSION(線性迴歸)

$\mu_{Y|x} = \alpha + \beta x$
 而實際依變數的觀察值 y_i 與 $\mu_{Y|x=x_i}$ 存在一個隨機之誤差值 ε_i



http://web.ydu.edu.tw/~alan9956/docu1/0951_stat/stat_13.pdf

模型解釋能力

變異來源	平方和	自由度	均方	f 值
迴歸模型	20.512	1	20.512	84.066
隨機誤差	0.488	2	0.244	
總和	21	3		

而其拒絕域為 $\{f_0 \geq f_{0.05}(1,2)\} = \{f_0 \geq 18.51\}$

然而抽樣所得之檢定值 $f = 84.066 > 18.51$ ，因此拒絕 H_0 ，即此迴歸模型具有解釋能力

$$R^2 = \frac{SSR}{SST} = \frac{20.512}{21} = 0.977$$

LINEAR REGRESSION(線性迴歸)

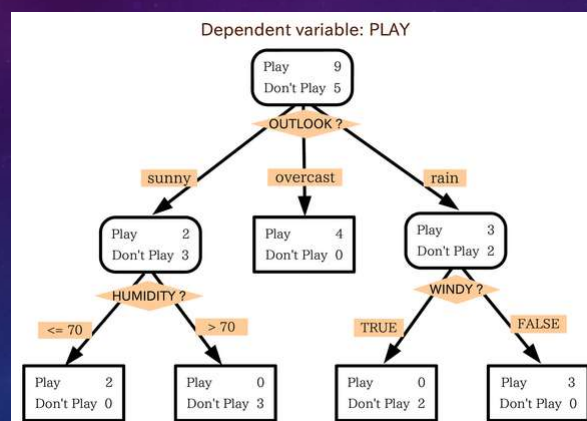
- 優點
 - 具有完整的統計理論基礎
 - 有許多支援的分析工具
 - 容易操作
 - 在研究上具有公信力
- 缺點
 - 需滿足基本假設
 - 變數的掌握
 - 真實資料裡不易存在線性關係

機器學習－決策樹（問題描述）

Play golf dataset

Independent variables				Dep. var
OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
overcast	83	78	FALSE	Play
rain	70	96	FALSE	Play
rain	68	80	FALSE	Play
rain	65	70	TRUE	Don't Play
overcast	64	65	TRUE	Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
rain	75	80	FALSE	Play
sunny	75	70	TRUE	Play
overcast	72	90	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play

機器學習－決策樹（模型建構）



<https://zh.wikipedia.org/wiki/%E5%86%B3%E7%AD%96%E6%A0%91>

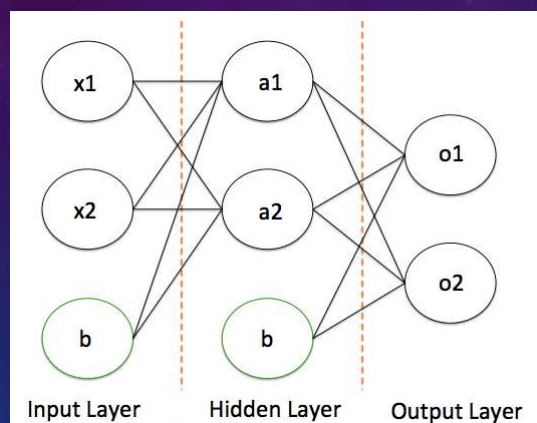
ENTROPY (熵)

$$I_E(i) = - \sum_{j=1}^m f(i, j) \log_2 f(i, j)$$

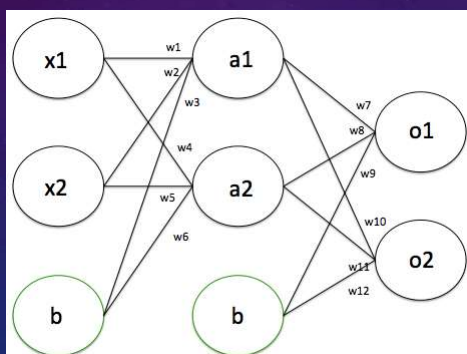
DECISION TREE(決策樹)

- 優點
 - 可顯示決策規則
 - 有許多支援的分析工具
 - 容易操作
 - 速度快
- 缺點
 - 類別型變數
 - 變數的掌握
 - 在複雜的資料裡不易導入

NEURAL NETWORKS



NEURAL NETWORKS



倒傳遞演算法

1. 初始化神經網路所有權重
2. 將資料由input layer往output layer向前傳遞運算(forward)計算出所有神經元的output
3. 計算誤差由output layer往input layer向後傳遞運算(backward)算出每個神經元對誤差的影響
4. 用誤差影響去更新權重
5. 重複2,3,4直到誤差收斂到夠小

- <http://terrence.logdown.com/posts/1132631-neural-networks-with-backpropagation-one-notes>

NEURAL NETWORK(類神經網路)

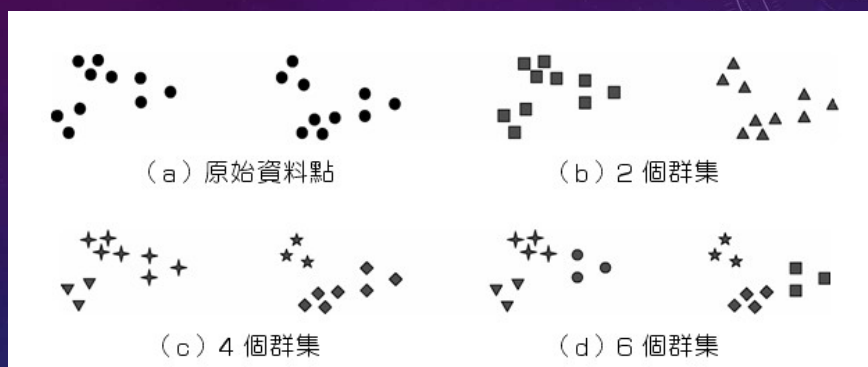
- 優點
 - 可處理稍微複雜資料
 - 有許多支援的分析工具
 - 容易操作
 - 速度快
- 缺點
 - 無法顯示規則
 - 參數的操作
 - 隱藏層的數量有限

群集分析

- 群集分析將資料分成許多有意義或有用的群體（群集）
- 若以有意義的群體為目標，則群集應捕捉資料的基本結構。但在一些情況下，群集分析對其他應用而言只是一個起點而已，例如資料彙總
- 不論是針對理解性（understanding）或使用性（utility）而言，群集分析在心理學與其他社會科學、生物學、統計學、樣式辨識（pattern recognition）、資訊檢索、機器學習與資料探勘等領域上，都扮演很重要的角色

<http://web.nchu.edu.tw/pweb/users/ykchan/lesson/6570.ppt>

群集CLUSTER



K-MEANS

- 以雛形為基礎之群集技術可建立具有一個階層之資料分割，這樣的技術有一些，但最常見的兩個技術是K-means與K-medoid。
- K-means使用雛型的中心，通常是一群資料點的平均，且典型地被應用於連續 n 維度空間的物件上。K-medoid使用雛型之中心點，以一個最具代表性的點來表示一群資料點，可應用的範圍很廣，因為它需要對成一對的資料點做鄰近值測量，使得中心點不會對應至實際的資料點，而中心點根據其定義必須是實際的資料點

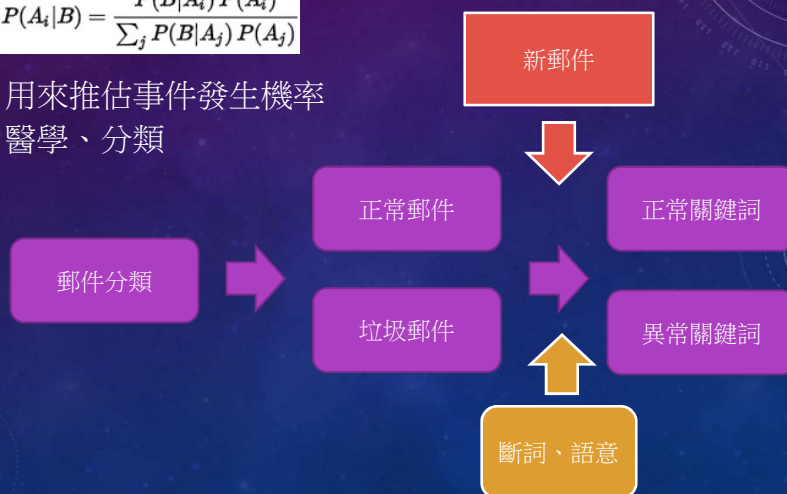
CLUSTER(群集分析)

- 優點
 - 可快速掌握新的資料概況
 - 有許多支援的分析工具
 - 容易操作
 - 速度快
- 缺點
 - 群組數不定
 - 受到離群值、組數的影響
 - 對於密度不均的資料效果較差

貝氏機率

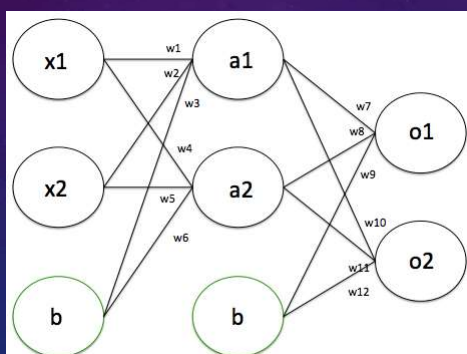
$$P(A_i|B) = \frac{P(B|A_i) P(A_i)}{\sum_j P(B|A_j) P(A_j)}$$

- 用來推估事件發生機率
- 醫學、分類



深度學習 DEEP LEARNING

NEURAL NETWORKS



激活函数

激活函数

- Rectified Linear Unit(ReLU) - 用于隐层神经元输出
- Sigmoid - 用于隐层神经元输出
- Softmax - 用于多分类神经网络输出
- Linear - 用于回归神经网络输出 (或二分类问题)

ReLU函数计算如下：

$$\phi(x) = \max(0, x)$$

Sigmoid函数计算如下：

$$\phi(x) = \frac{1}{1+e^{-x}}$$

Softmax函数计算如下：

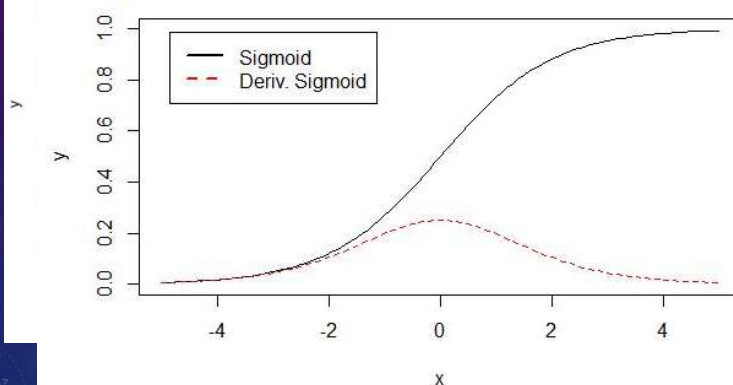
$$\phi_i(z) = \frac{e^{z_i}}{\sum_{j \in \text{group}} e^{z_j}}$$

http://blog.csdn.net/Leo_Xu06/article/details/53708647

SIGMOID FUNCTION

$$\phi'(x) = \phi(x)(1 - \phi(x))$$

对应的图形如下：



SOFTMAX

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{for } j = 1, \dots, K.$$

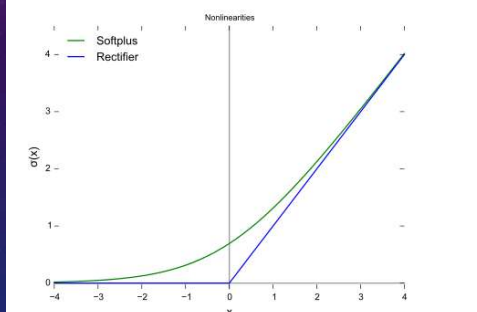
- 適合用於輸出多個類別，各類別機率合計等於1

```
import math
z = [1.0, 2.0, 3.0, 4.0, 1.0, 2.0, 3.0]
z_exp = [math.exp(i) for i in z]
print(z_exp) # Result: [2.72, 7.39, 20.09, 54.6, 2.72, 7.39, 20.09]
sum_z_exp = sum(z_exp)
print(sum_z_exp) # Result: 114.98
softmax = [round(i / sum_z_exp, 3) for i in z_exp]
print(softmax) # Result: [0.024, 0.064, 0.175, 0.475, 0.024, 0.064, 0.175]
```

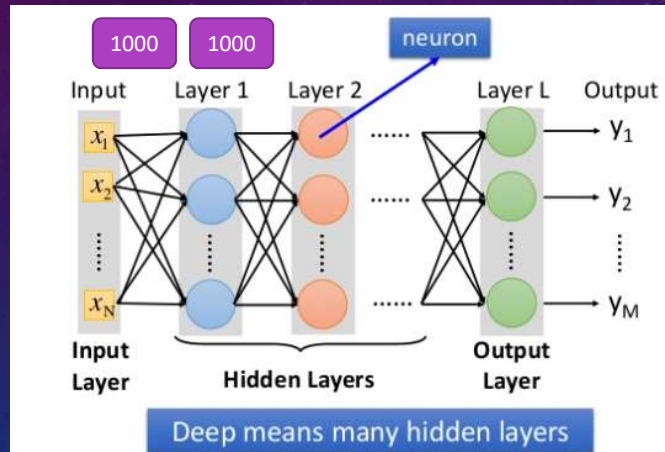
RELU(RECTIFIED LINEAR UNIT)

$$\phi'(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$$

Relu函数的形状如下（蓝色）：



FULLY CONNECT FEEDFORWARD NETWORK



- <https://panx.asia/archives/53209>

Ultra Deep Network

http://cs231n.stanford.edu/slides/winter1516_lecture8.pdf

16.4%

8 layers

AlexNet (2012)

7.3%

19 layers

VGG (2014)

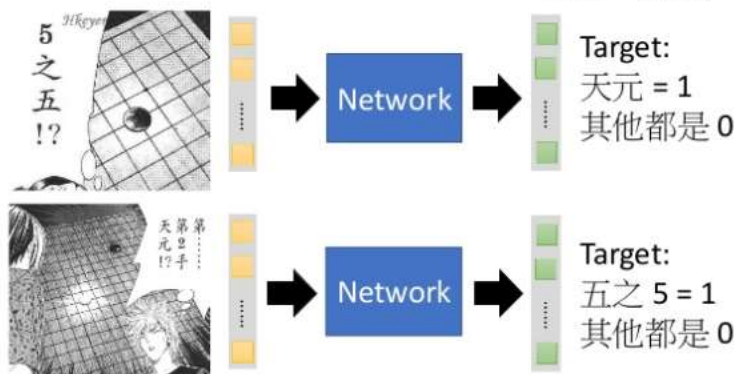
6.7%

22 layers

GoogleNet (2014)

Playing Go

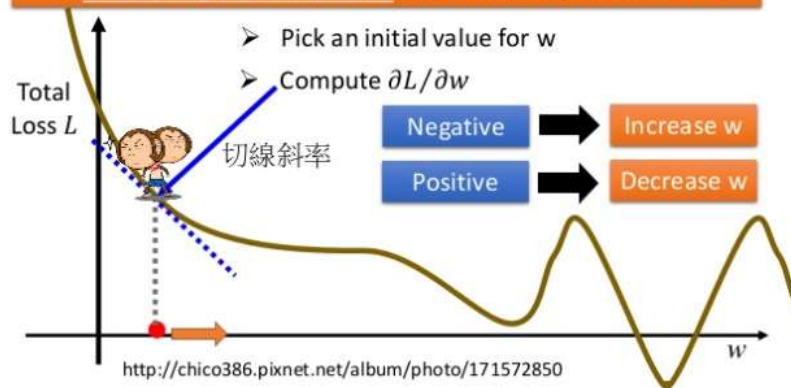
Training:

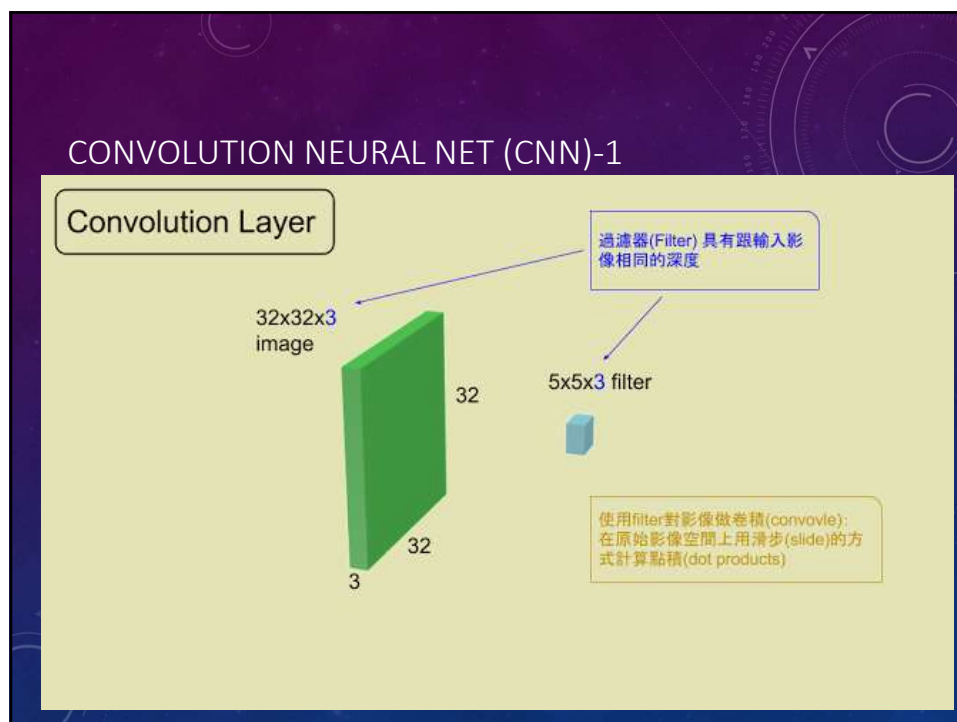
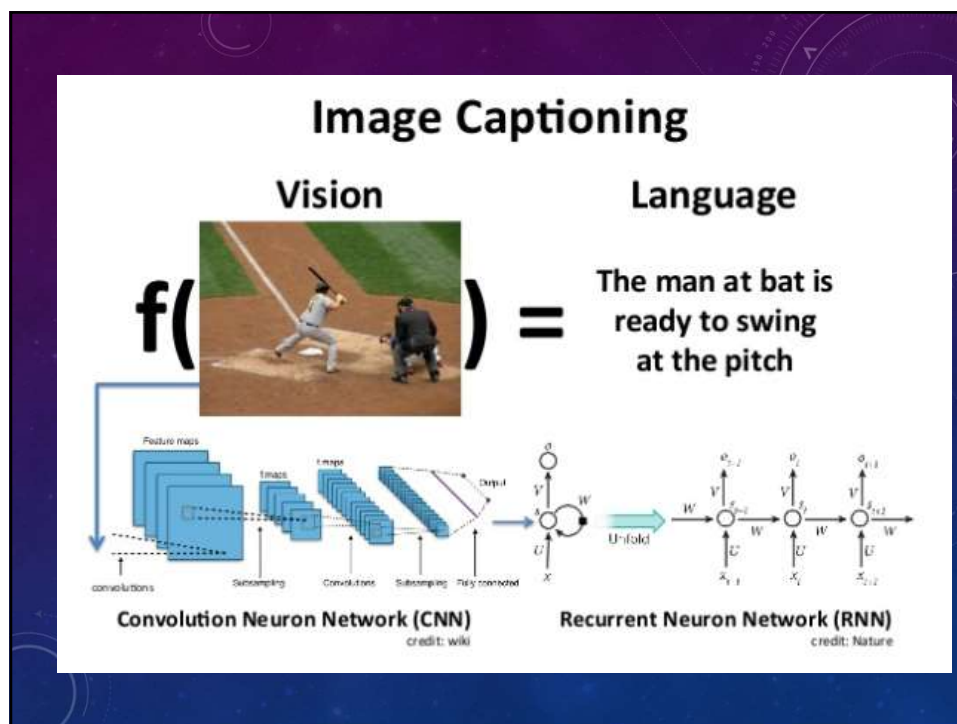


Gradient Descent

Network parameters $\theta = \{w_1, w_2, \dots, b_1, b_2, \dots\}$

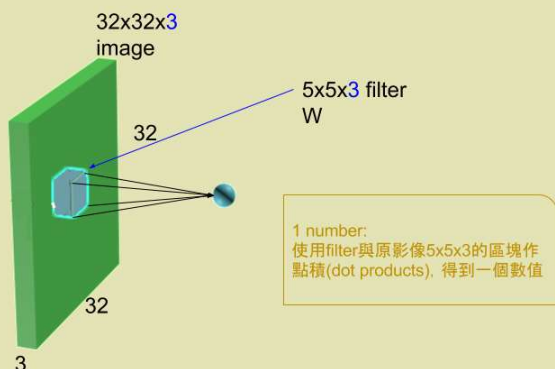
Find network parameters θ^* that minimize total loss L





CONVOLUTION NEURAL NET (CNN)-2

Convolution Layer



CONVOLUTION NEURAL NET (CNN)-3

1	1	0	0	1	1	0
x1	x0	x1	x0	x1	x0	1
0	1	1	0	1	1	0
x0	x1	x0	x1	x0	1	1
1	1	0	1	1	1	0
x0	x1	x0	x1	x0	1	0
0	0	0	1	1	1	0
x1	x1	x1	x0	x1	1	0
0	1	1	0	1	1	0
x1	x1	x0	x1	x1	1	0
0	0	0	1	1	0	1
1	1	1	1	1	0	1

$1x1+1x0+0x1+0x0+1x1+$
 $0x0+1x1+1x0+0x1+1x0+$
 $1x0+1x1+0x0+1x1+1x1+$
 $0x1+0x1+0x1+1x0+1x1+$
 $0x1+1x1+1x0+0x1+1x1=9$

9		

卷積後的值

使用 5x5 filter 作卷積

CONVOLUTION NEURAL NET (CNN)-4

Stride=1 往右移一步

1	1	0	0	1	1	0
0	1	1	0	1	1	0
1	1	0	1	1	1	1
0	0	0	1	1	1	0
0	1	1	0	1	1	0
0	0	0	1	1	0	1
1	1	1	1	1	0	1

$$1 \times 1 + 0 \times 0 + 0 \times 1 + 1 \times 0 + 1 \times 1 + 1 \times 0 + 1 \times 1 + 1 \times 0 + 0 \times 1 + 1 \times 0 + 1 \times 1 + 1 \times 1 + 0 \times 1 + 0 \times 1 + 1 \times 1 + 1 \times 0 + 1 \times 1 + 1 \times 1 + 1 \times 1 + 0 \times 0 + 1 \times 1 + 0 \times 0 + 1 \times 1 + 1 \times 1 = 12$$

9	12	

卷积後的值

使用 5x5 filter 作卷积

CONVOLUTION NEURAL NET (CNN)-5

Stride=1

1	1	0	0	1	1	0
0	1	1	0	1	1	0
1	1	0	1	1	1	1
0	0	0	1	1	1	0
0	1	1	0	1	1	0
0	0	0	1	1	0	1
1	1	1	1	1	0	1

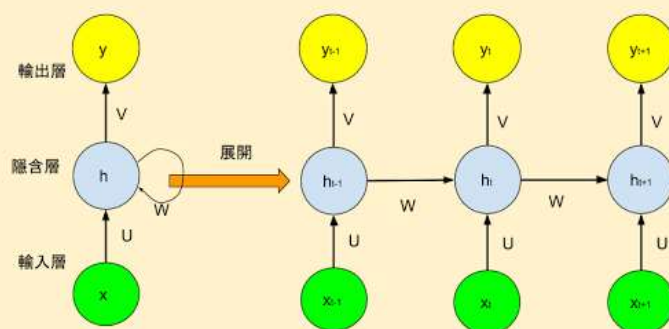
$$0 \times 1 + 1 \times 0 + 1 \times 1 + 1 \times 0 + 1 \times 1 + 0 \times 0 + 1 \times 1 + 1 \times 0 + 1 \times 1 + 0 \times 0 + 1 \times 0 + 0 \times 1 + 1 \times 1 + 0 \times 1 + 0 \times 1 + 1 \times 1 + 1 \times 0 + 0 \times 1 + 1 \times 1 + 1 \times 1 + 1 \times 0 + 0 \times 1 + 1 \times 1 = 11$$

9	12	9
9	9	10
10	11	11

卷积後的
feature map

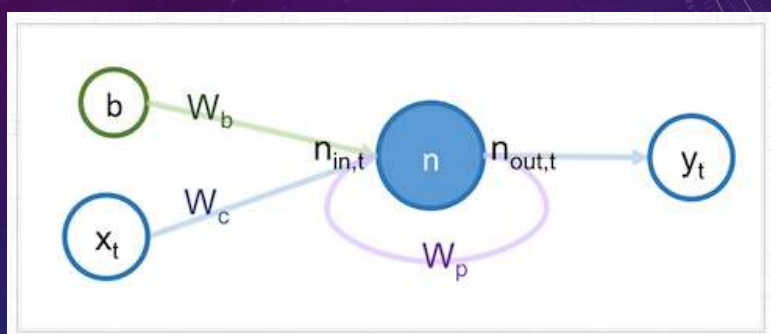
使用 5x5 filter 作卷积

RECURRENT NEURAL NET (RNN)

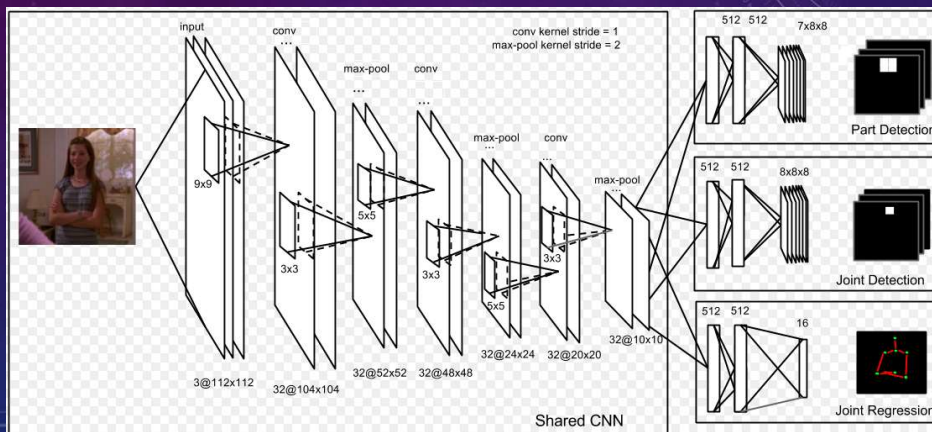


循環神經網路(Recurrent Neural Networks)

SIMPLE_RNN



CONVOLUTIONAL NEURAL NETWORK

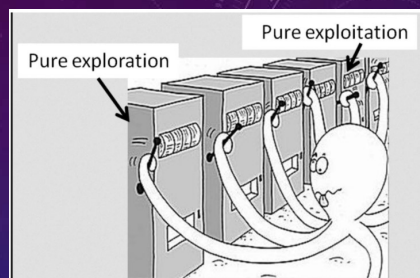


機器學習

- 步驟
 - 資料清理
 - 特徵萃取、選擇
 - 模型選取
- 類別
 - 監督式學習 (先貼標籤)
 - 非監督式學習 (免貼標籤)

強化學習

- 多拉桿吃角子老虎機(multi-armed Bandit)
- 探索 (exploration) 各個機台報酬的可能性
- 開發 (exploitation) 重複玩期望最高的機器
- 假設賭徒的資產足夠讓他玩 1000 次，而賭徒心中設定 $\epsilon = 10\%$ ，那麼在進行前 100 次遊戲時，賭徒將會隨機拉動一個拉桿，並記錄得下的報酬。之後的 900 次遊戲，賭徒將會根據過去的報酬的紀錄，每次遊戲皆選擇平均報酬最高的機台進行，全力衝刺賺取最大報酬。



<https://blog.yoctol.com/%E8%B3%AD%E5%BE%92%E7%9A%84%E4%BA%BA%E5%B7%A5%E6%99%BA%E6%85%A72-%E8%87%A8%E5%BA%8A%E8%A9%A6%E9%A9%97%E4%B8%AD%E7%9A%84%E5%90%83%E8%A7%92%E5%AD%90%E8%80%81%E8%99%8E%E6%A9%9F-2f661cd27c6c>

強化學習應用

- 新聞推薦
 - 每種類型新聞是一台Bandit
 - 按日期排序，顯示最新的文章
 - 記錄使用者點擊過程

2016失敗的人工智慧案例

- Northpointe 公司開發了一個設計用於預測犯罪的人工智能系統
- 加入了人工智能的視頻遊戲《精英》
- 特斯拉自動駕駛儀模式下的死亡事件
- 春季的時候微軟在 Twitter 上發布了「小冰」
- Pokémon Go 使得遊戲玩家集中到現實世界
- 中國的面部識別學習預測罪犯(86%)
- 由 Knightscope 平台所創造的一個模擬環境中，一個 16 月大的男童受傷。

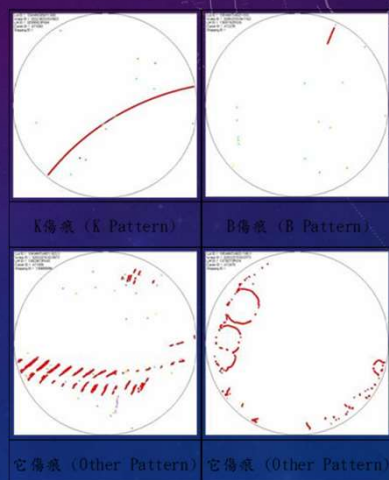
原文網址：<https://kknews.cc/zh-tw/>



- 實務案例介紹

SVM應用—晶圓傷痕種類自動辨識

- K 傷痕 (K Pattern)
 - 連續性或線性，成一長圓弧狀圖形。可能發生位置為隨機性。
- B 傷痕 (B Pattern)
 - 缺陷點為小於10 mm的短直線圖形可能發生位置為隨機性。。
- 它 傷痕 (Other Pattern)
 - 除上述兩種特定圖形外，其他各種造成缺陷的特殊圖形



59/
58

處理流程

- 目標物 (光源)
- 照相
- 儲存
- 前處理
 - 強化、去雜訊、加強對比、加強邊緣、擷取特徵、壓縮、轉換
- 分析演算法
- 導入製程

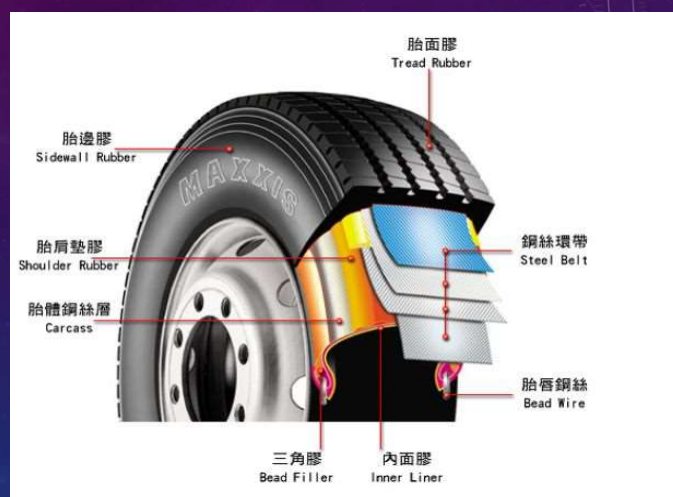
改善數據

序號	c	g	正確率
1	100	0.000000095	64%
2	10	0.000000095	64%
3	2	0.000000095	64%
4	1.5	0.000000095	83%
5	1.2	0.000000095	89%
6	1.1	0.000000095	89%
7	1	0.000000095	89%
8	0.8	0.000000095	69%
9	0.5	0.000000095	69%
9	0.2	0.000000095	69%
9	0.01	0.000000095	69%

序號	c	g	正確率
1	100	0.000000124	66%
2	10	0.000000124	66%
3	2	0.000000124	66%
4	1.5	0.000000124	72%
5	1.2	0.000000124	86%
6	1	0.000000124	88%
7	0.8	0.000000124	88%
8	0.5	0.000000124	68%
9	0.2	0.000000124	68%
10	0.01	0.000000124	68%

61/
58

類神經應用－輪胎組成與構造

62/
58

預測數據實測驗證

編號	PHR			模擬物理性能數據				
	BR	NR	SBR	M99.9	M300	拉伸力強度	伸長率	抗撕拉力
F11	0	80	20	34.0	146.6	257.8	483.4	99.9
F12	0	40	60	33.1	159.5	199.5	355.6	87.0
F13	20	80	0	33.8	143.5	254.4	482.4	99.8
F14	60	40	0	37.9	133.9	157.1	300.2	90.8

編號	PHR			驗證（現狀）物理性能數據				
	BR	NR	SBR	M99.9	M300	拉伸力強度	伸長率	抗撕拉力
F11	0	80	20	36.3	155.6	260.1	473.5	99.5
F12	0	40	60	34.2	158.2	209.3	389.2	88.3
F13	20	80	0	33.8	141.3	248.3	480.1	99.6
F14	60	40	0	35.9	131.5	180.7	375.4	91.3

back

63/
58

物聯網應用－田徑選手訓練輔助系統

RASPBERRY PI

A small and affordable computer that you can use to learn programming



訊號放大器



壓力感測器



訊號轉換器

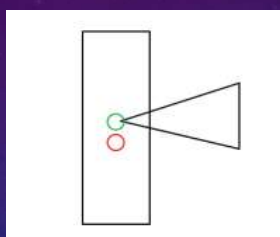


起跑架與壓力感測設備

物聯網應用－田徑選手訓練輔助系統

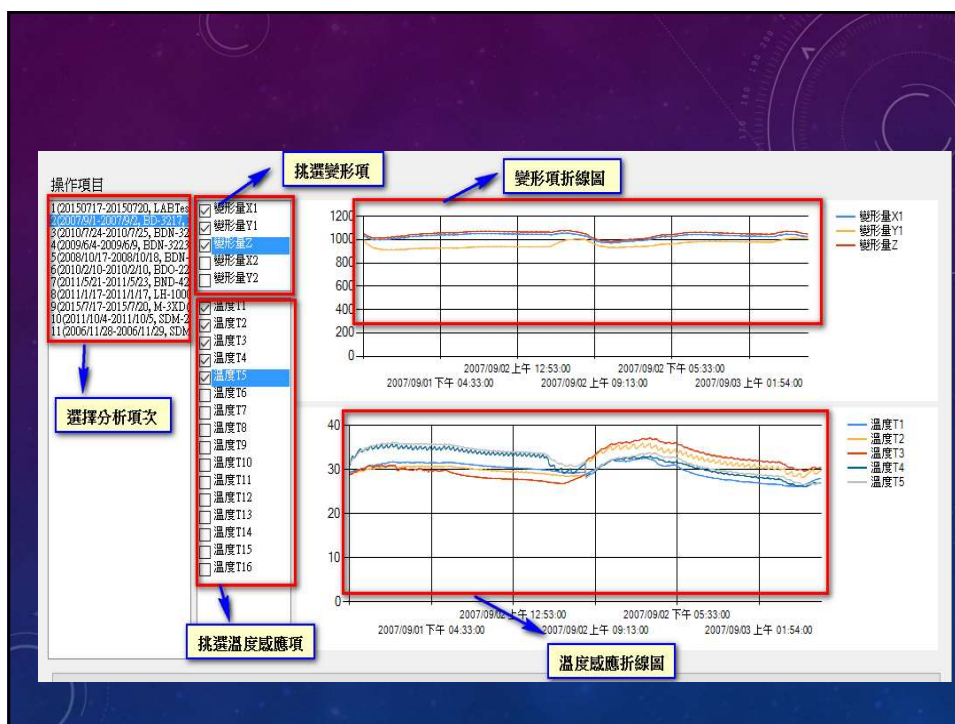


精密加工路徑預測



量測位移資料、溫度感測器偵測值

熱變形分析預測系統
開發工具：C#
演算法：多元線性迴歸、類神經網路、支援向量機
資料庫：SQL Server



製程資料說明

- 參數資料
 - 資料筆數：300萬，150個參數

	filename	MinDate	MaxDate	RowCount		filename	MinDate	MaxDate	RowCount
1	1.csv	2014-01-11	2014-01-14	25,921	11	11.csv	2014-06-04	2014-06-05	8,641
2	2.csv	2014-01-23	2014-01-26	25,921	12	12.csv	2014-06-05	2014-06-13	69,120
3	3.csv	2014-01-26	2014-01-31	51,839	13	13.csv	2014-06-28	2014-06-30	17,281
4	4.csv	2014-02-01	2014-02-07	51,841	14	14.csv	2014-06-30	2014-07-02	17,280
5	5.csv	2014-02-12	2014-02-14	17,281	15	15.csv	2014-07-02	2014-07-06	34,560
6	6.csv	2014-03-18	2014-03-19	8,641	16	16.csv	2014-07-08	2014-07-09	8,641
7	7.csv	2014-03-19	2014-03-25	51,840	17	17.csv	2014-07-18	2014-07-22	34,561
8	8.csv	2014-04-05	2014-04-07	17,281	18	18.csv	2014-09-28	2014-09-30	17,281
9	9.csv	2014-04-07	2014-04-14	60,480	19	19.csv	2014-10-07	2014-10-11	34,561
10	10.csv	2014-05-17	2014-05-22	43,201	20	20.csv	2014-10-11	2014-10-14	25,920

化工產線製程分析



69

資料分佈

資料行 data line	最小值	最大值	平均	標準差
2014/1/1 上午...	2016/1/22 上午...	2014/1/1 上午...	2016/1/22 上午...	2014/1/1 上午...
id	2014011100000...	2016120200000...	2014963617838...	8114221058.3632
	-0.109941304	118.5099945	13.6155741369	6.28580066428895
	0	0	0	0
	0	1.004999518	0.52305182542	0.389666580590035
	0	10.04999924	7.27553574375	3.13636156556183
	0	20.03759384	2.23041131503	3.94845666469525
	0	15.04794407	0.1312428517	1.18480140151127
	0	100.5	72.796761772	32.1702881173174
	0	10.0300069	3.02215467886	2.62680787239413
	0	240.1999969	123.25491549945	22.2067199411635
	1.58276E-07	0.486999869	0.45934452715	0.019420322250496
	0	0.08000005	0.07596529672	0.011754502176941
	0.001	0.501999974	0.43711588745	0.0561675510913868
	-0.238812014	15.02647114	4.75011824580	1.32142034669219
	-0.238812014	15.02647114	4.75011824580	1.32142034669219
	-2.5	10.30999947	0.55601318452	0.159077384197949
	0	20.63000107	5.48012491790	1.0471040484124
	0	194	1.11837924079	13.4846206622956
	-64	5156	2050.14680374	1018.54383484849
	0	63.6173172	0.01560791373	0.39365519048764
	0	19665.1582	10083.1861323	3279.82860318956
	0	3315.541504	120.116177705	164.764419711225
	0	30948	9846.43557562	1908.4365791891
	0	251952	159488.266498	40443.0221185655
	0	23488.01953	14175.3066542	2549.0660511759
	0	22855.32813	11067.0314935	3117.21006055414
	0	41661.98047	26748.8193754	5421.25232843569
	40	221987.6875	161378.21162667	40814.577115667
	0	78712.96875	51993.009658822	11117.9112638693
	0	71515.90625	42265.7288022	11201.7019789781
	0	125901.5625	55766.432414	21897.2604762313
	47.32277879	15620.87207	3142.3625275	2512.19675502028
	0	20794.45898	14175.3552173	2549.4181581286
	-1.279866219	26.05960274	-0.00175565638	0.582895616441099
	-25	-0.01	-18.385691289	11.0238404527137
	0	419.83813	7.2287215171	36.1735540747483
	-204.0123138	30947.57813	9846.39606965	1908.70993795972
	0	1444	968.820093114	253.86369575175
	0	1444	894.678777836	339.43489284912
	0	2.601680279	0.03557865906	0.11473109014043
	0	332.8470459	3.22561447190	22.3800784267028
	0	1421.811768	1178.1828305887	173.203512018188
	0	1421.811768	1178.1828305887	173.203512018188
	0	1493	646.607617805	498.85642602275
	0	1493	92.0145787518	290.326544359116
	0.45.4532145	17760	11710.6423436	1068.83970016319

70

資料分析結果 - 變數重要性

no	Feature	Tag	Description	Gain
1	0			0.000159066
2	5			2.91E-09
3	5			2.66E-09
4	0			2.14E-09
5	4			1.76E-09
6	9			1.70E-09
7	7			1.55E-09
8	0			1.45E-09
9	5			1.45E-09
10	5			1.44E-09
11	7			1.44E-09
12	2			1.39E-09
13	4			1.38E-09
14	5			1.12E-09
15	4			1.06E-09
16	9			9.17E-10
17	9			9.04E-10
18	1			8.42E-10
19	5			7.61E-10
20	4			7.24E-10
21	9			6.67E-10

71

資料分析結果 - 變數挑選之誤差分析

No	變數選擇數目	誤差 (預測-實際)
1	10	0.10
2	11	0.10
3	12	0.10
4	13	0.10
5	14	0.08
6	15	0.08
7	16	0.08
8	17	0.08
9	18	0.08
10	19	0.08
11	20	0.08

72

現場即時預測分析說明



結論

- 確認分析主題
 - 變數掌握度
 - 效益夠顯著
 - 具有再現性
- 蒐集測試資料，證明方法可用性
- 探討相關流程，建立一套完整的資料彙整程序
- 測試相關演算法，找出參數調整方式
- 建置分析與回饋機制
- 定期驗證準確度，適時調整模型



感謝您的聆聽