



---

# 深度學習在投資領域的應用

丘祐瑋 – David Chiu

EMAIL: [david@largitdata.com](mailto:david@largitdata.com)

網站: [www.largitdata.com](http://www.largitdata.com)

電話: +886929094381

# 關於我

---



- 大數軟體有限公司創辦人
- 前趨勢科技工程師
- 大數學堂

<http://www.largitdata.com/>

- 粉絲頁

<https://www.facebook.com/largitdata>

- R for Data Science Cookbook

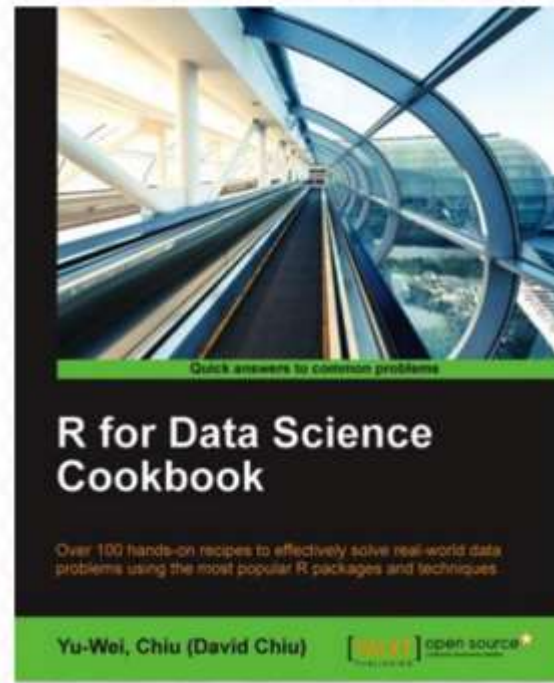
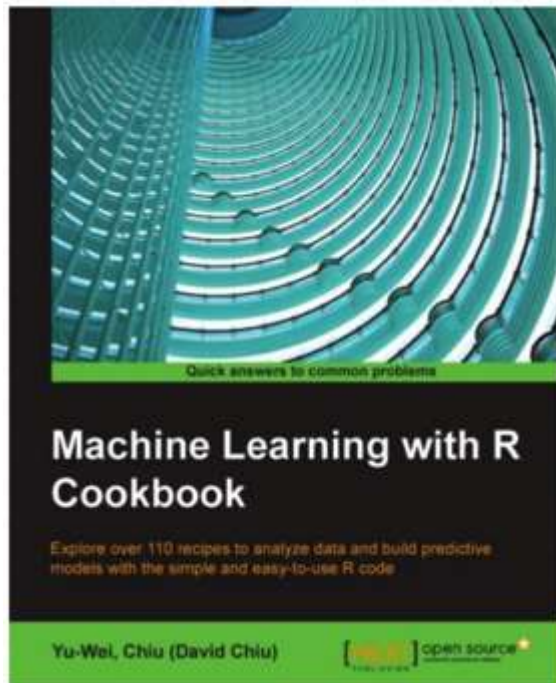
<https://www.packtpub.com/big-data-and-business-intelligence/r-data-science-cookbook>

- Machine Learning With R Cookbook

<https://www.packtpub.com/big-data-and-business-intelligence/machine-learning-r-cookbook>

# Machine Learning With R Cookbook (機器學習與R語言實戰) & R for Data Science Cookbook (數據科學:R語言實現)

---



---

Author: Yu-Wei (David) Chiu



# 人工智慧的發展歷史

---



AlphaGo

使用深度學習技術打敗頂尖棋手



## AlphaGO Zero

只需要三天就能熟悉第一代AlphaGo  
所有的圍棋知識



A white Waymo self-driving car is shown from a rear three-quarter view. The car has a black sensor dome on its roof and a green and blue 'W' logo with the word 'WAYMO' below it on its side. The background is a blurred outdoor scene with trees and a road, suggesting a test drive environment.


# Google 成立 Waymo

## 打造全球第一個無人車上路測試



什麼是人工智能(A.I)?



A man in a dark suit stands with his back to the camera, looking at a large, complex machine. The machine is filled with many circular components, possibly rotors or dials, arranged in a grid. The scene is dimly lit, with a warm, golden light emanating from the machine, creating a dramatic and historical atmosphere. The machine appears to be a cryptographic device, likely the Enigma machine or a similar device used during World War II.

模仿遊戲 (The Imitation Game)，講述了**計算機科學之父 艾倫·圖靈**的傳奇人生，故事主要聚焦於圖靈協助盟軍破譯德國密碼系統**英格瑪**，從而扭轉二戰戰局的經歷

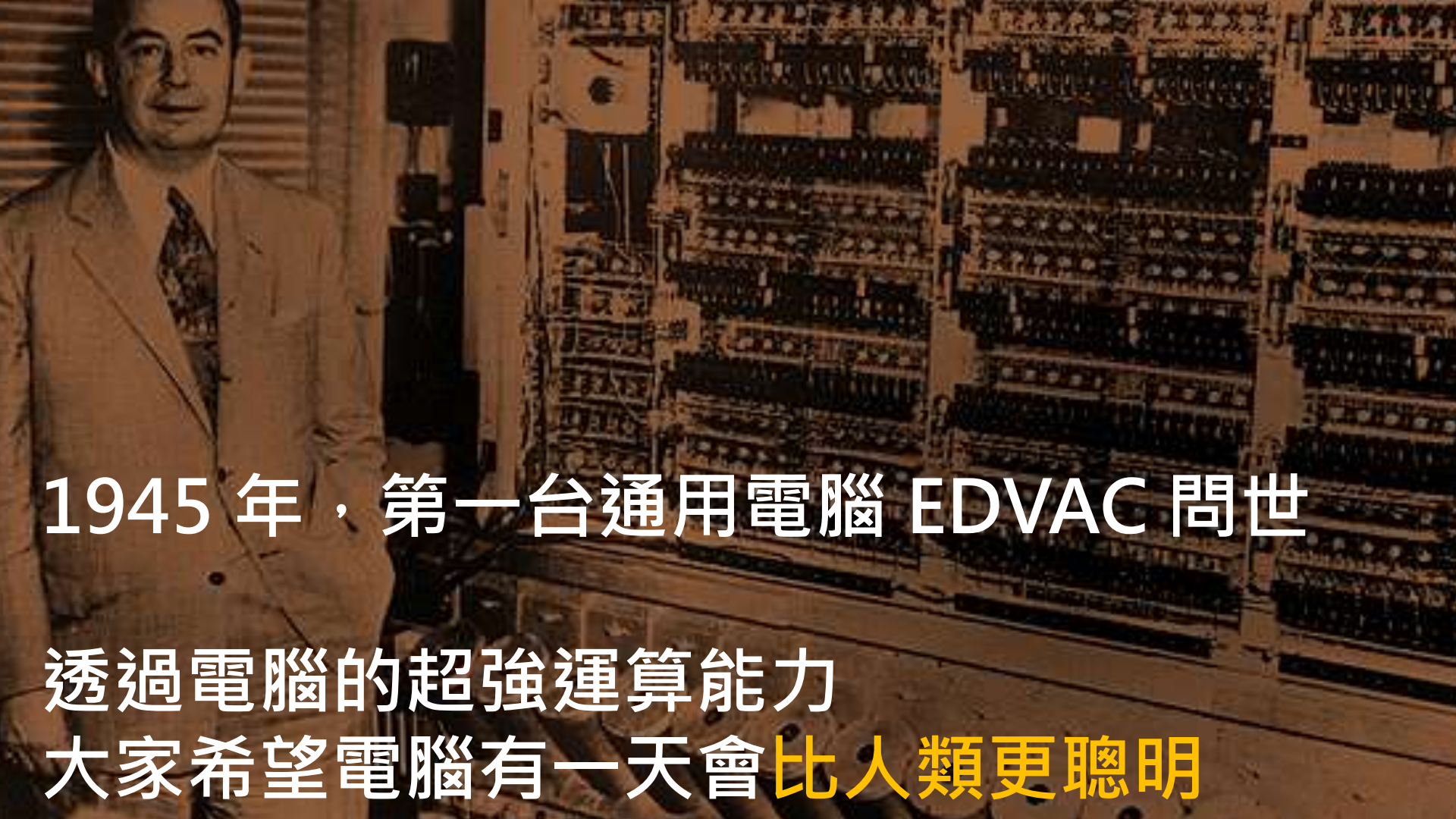
# INSPIRATIONAL STORY OF ALAN TURING

FATHER OF MODERN COMPUTING

讓我問你一個問題，你必須  
作答。根據你的回答，我將  
能判定：你是人，還是機器







1945 年，第一台通用電腦 EDVAC 問世

透過電腦的超強運算能力

大家希望電腦有一天會比人類更聰明



1956 年舉辦于達特茅斯的一場研討會

紐厄爾 (Newell)、西蒙 (Simon) 展示了  
“全世界第一個人工智慧程式”  
邏輯理論家 (Logic Theorist)

A large group of humanoid robots, resembling the models from the movie "I, Robot", are standing in rows. They have a metallic, silver-colored body with a more human-like face. The background is dark and industrial.

阿西莫夫《我，機器人》中設定**機器人三定律**做為機器人的行為準則

第一定律：機器人不得傷害人類個體，或者目睹人類個體將遭受危險而袖手不管

第二定律：機器人必須服從人給予它的命令，當該命令與第一定律衝突時例外

第三定律：機器人在不違反第一、第二定律的情況下要盡可能保護自己的生存



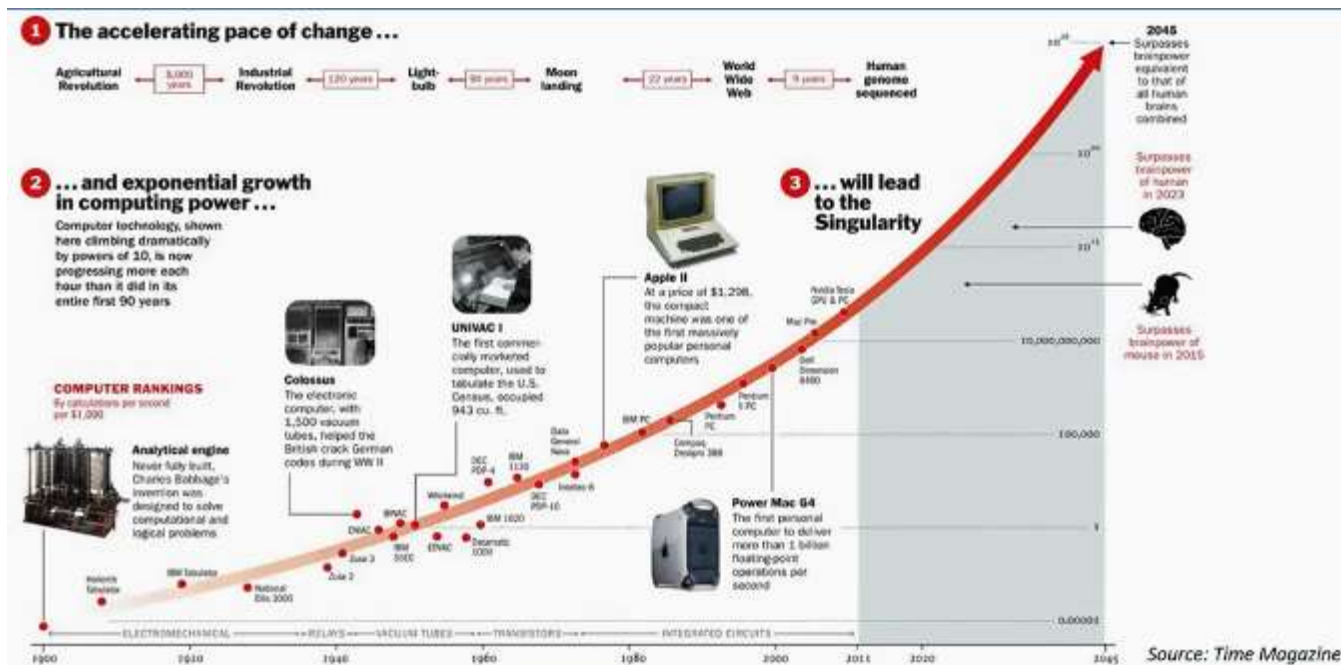
# 硬體成本逐漸降低



# 莫爾定律

當價格不變時，**電晶體**上可容納的元器件的數目，約每隔18-24個月便會增加一倍，性能也將提升一倍

換言之，每一美元所能買到的電腦性能，將每隔18-24個月翻一倍以上



A photograph of John Searle, an elderly man with white hair, wearing a striped shirt. He is gesturing with both hands raised, palms facing forward, as if explaining something. The background is dark and out of focus.

1980年，約翰瑟爾(John Searle)提出  
強人工智慧(Strong A.I.)和弱人工智慧(Weak A.I.) 的分類

強A.I.：指人工思考智慧，大膽假設電腦能具有與人相同程度的思考能力

弱A.I.：指人工類比智慧，主張機器只能模擬人類具有思維的行為表現，而不是真的懂得思考



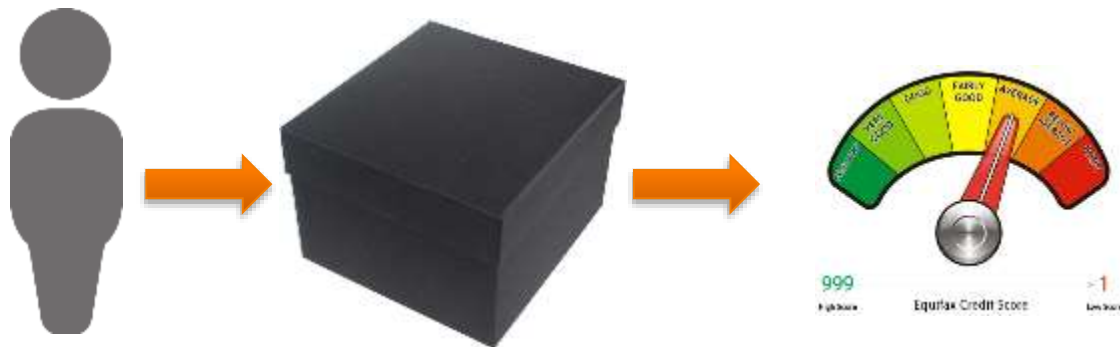
## 強人工智慧

模仿人類推理過程的思考模式，需要百分之百確定的事實配合，實務上應用困難

# 弱人工智慧 - 預測模型

是否能用**統計機率學**來處理人工智慧的問題呢？

e.g. 根據個人行為的特定機制 (是否點擊廣告)，預測模型把個人特性(身高，性別，職業)當輸入資料，提供預測分數當產出資料，分數越高，產生該行為的機會越高。







# 機器學習

---

# 統計

---

日常生活中常需要根據不完整的資訊做決定

統計可以把不確定的程度量化，用精確的方式來表達，掌握不確定的程度

## 統計學的目的

- 分析資料，將資料做出摘要
- 做出更好的決定
- 辨識出能提升做每件事的效果
- 評估決策或事項的效用

# 敘述性統計 v.s. 推論性統計

---

## 敘述性統計


- 有系統的歸納資料，瞭解資料的輪廓
- 對數據樣本做敘述性陳述，例如：平均數、標準差、計次頻率、百分比
- 對數據資料的圖像化處理，將資料摘要變為圖表

## 推論性統計

- 資料模型的建構
- 從樣本推論整體資料的概況
- 相關、回歸、單因數變異數、因素分析

# 讓機器做預測分析

從歷史資料建構預測性模型，以預測未知值



Name	Balance	Age	Employed	Bad debt
Mike	20000	42	N	Y
Mary	25000	33	Y	N
Claudio	115000	40	N	N
Robert	29000	23	Y	Y
Dora	72000	31	N	N

# 機器學習

---

機器學習的目的是：歸納 ( Induction )

- 從詳細事實到一般通論

*A computer program is said to learn from experience  $E$  with respect to some task  $T$  and some performance measure  $P$ , if its performance on  $T$ , as measured by  $P$ , improves with experience  $E$  -*  
- Tom Mitchell (1998)

找出有效的預測模型

- 一開始都從一個簡單的模型開始
- 藉由不斷喂入訓練資料，修改模型
- 不斷提升預測績效



# 機器學習 v.s. 統計

---

## 統計學家

- 專注在於模型甚於資料
- 可以用小量資料便可進行預測
- 建立預測模型時已考慮隨機誤差

## 機器學習學家

- 專注在於資料甚於模型
- 用大量資料校正模型

# 用機率與規則產生預測模型

---

## 線性模型

e.g. 針對一個化妝品廣告，對女性的吸引力可以給予權重90%，男性權重只有10%，以權重搭配個人點擊機率 (15%) 可以算出對該使用者推薦的分數(或機率)

女性13.5%，男性1.5%

## 規則模型

e.g.

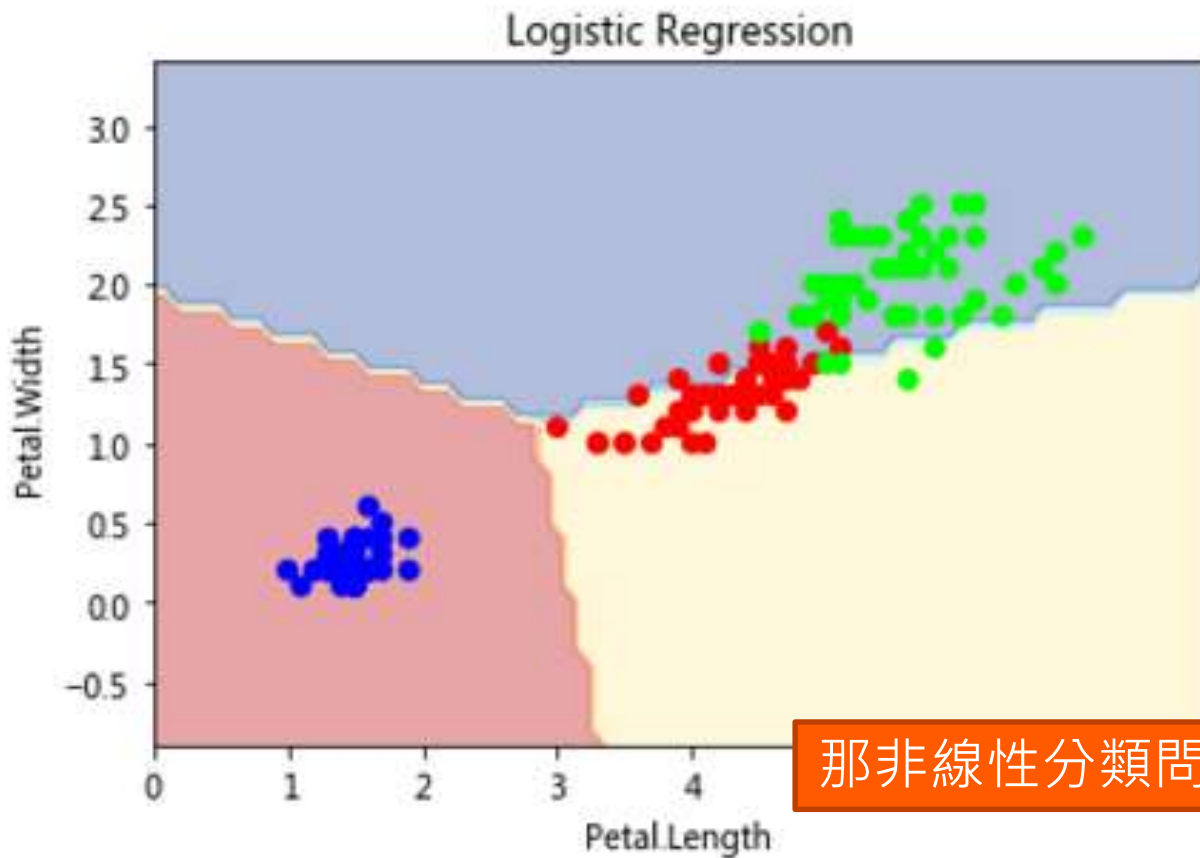
假使使用者為女性

而且月收入高達3萬以上

而且還沒看過這廣告

點擊機率為11%

# 解決線性分類問題



那非線性分類問題呢?

# 影像辨識

---



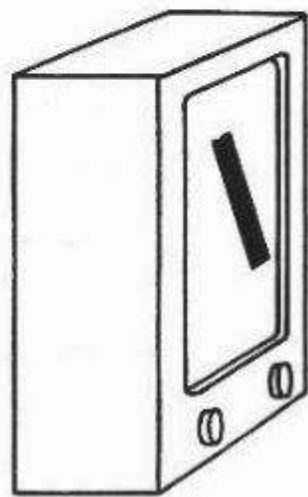


# 類神經網路

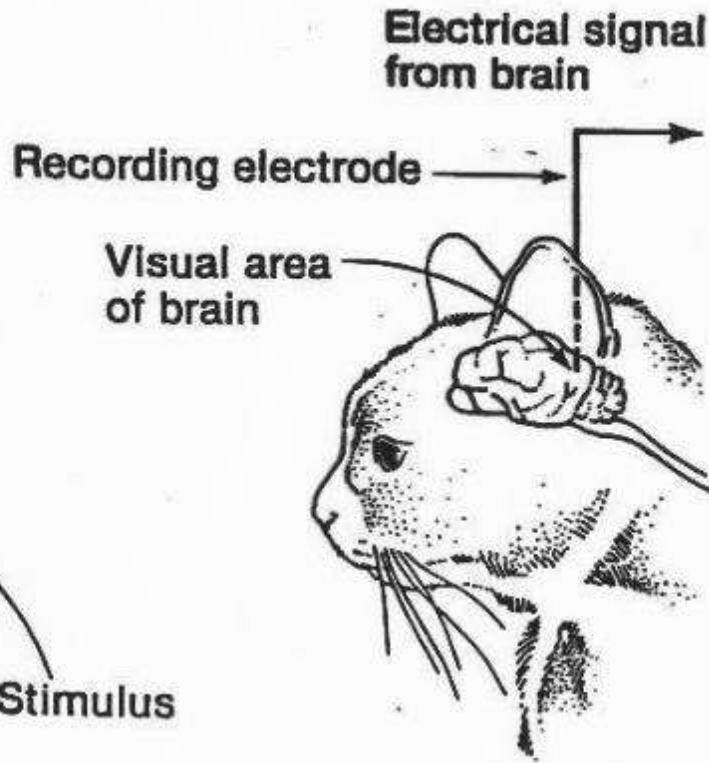
---

# 啟發於神經元的實驗

神經生物學家 David Hubel 和 Torsten Wiesel 在 1981 年發現貓的不同視覺神經元對於不同光影的反應不盡相同，在不同情況下的視覺神經元有著不同的活躍程度，甚至只對圖像的某些特定細節有反應，兩人因此發現而獲得了諾貝爾醫學獎。

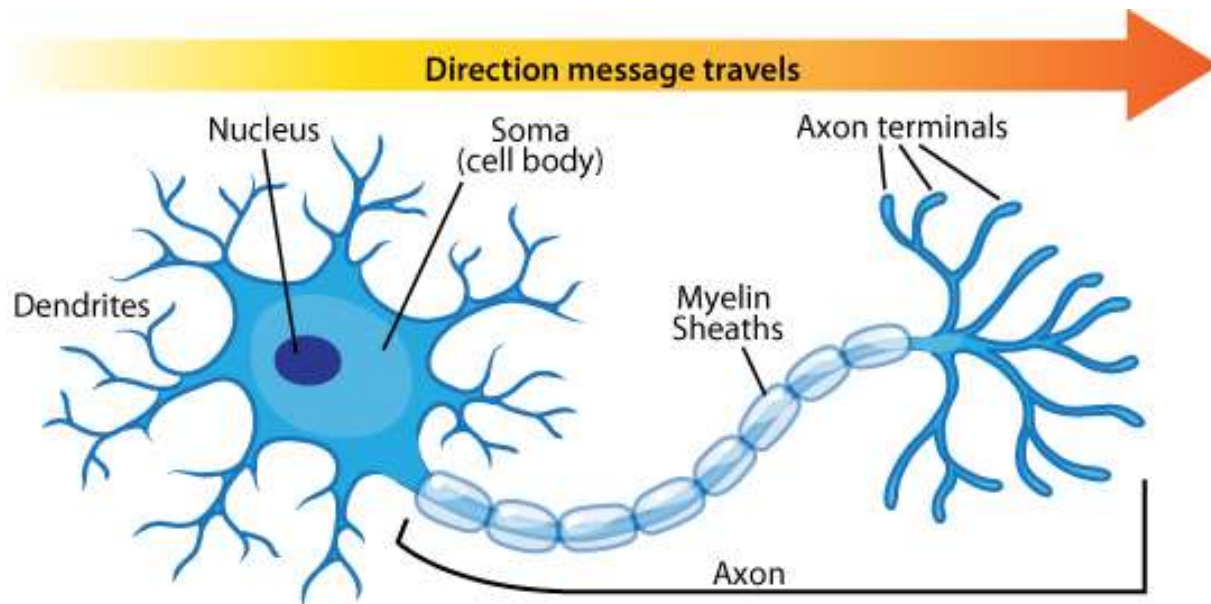


Stimulus



# 人工神經網路 (Artificial Neural Network)

神經系統由神經元構成，彼此間透過突觸以電流傳遞訊號。是否傳遞訊號、取決於神經細胞接收到的訊號量，當訊號量超過了某個閾值 (Threshold) 時，細胞體就會產生電流、通過突觸傳到其他神經元





# 萬用函數

如果能夠模擬神經元產生一個萬用函數呢？

$f(\text{audio waveform}) = \text{"How are you"}$

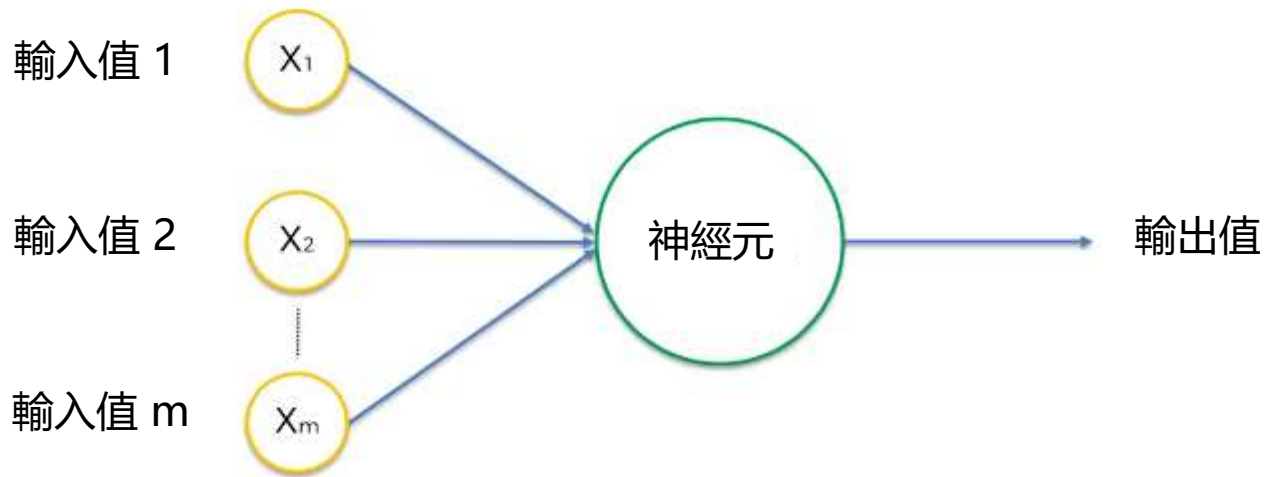
$f(\text{cat image}) = \text{"Cat"}$

$f(\text{chessboard}) = \text{"5-5"}$

是否就可以像人類一樣解決所有問題了？

# 感知機

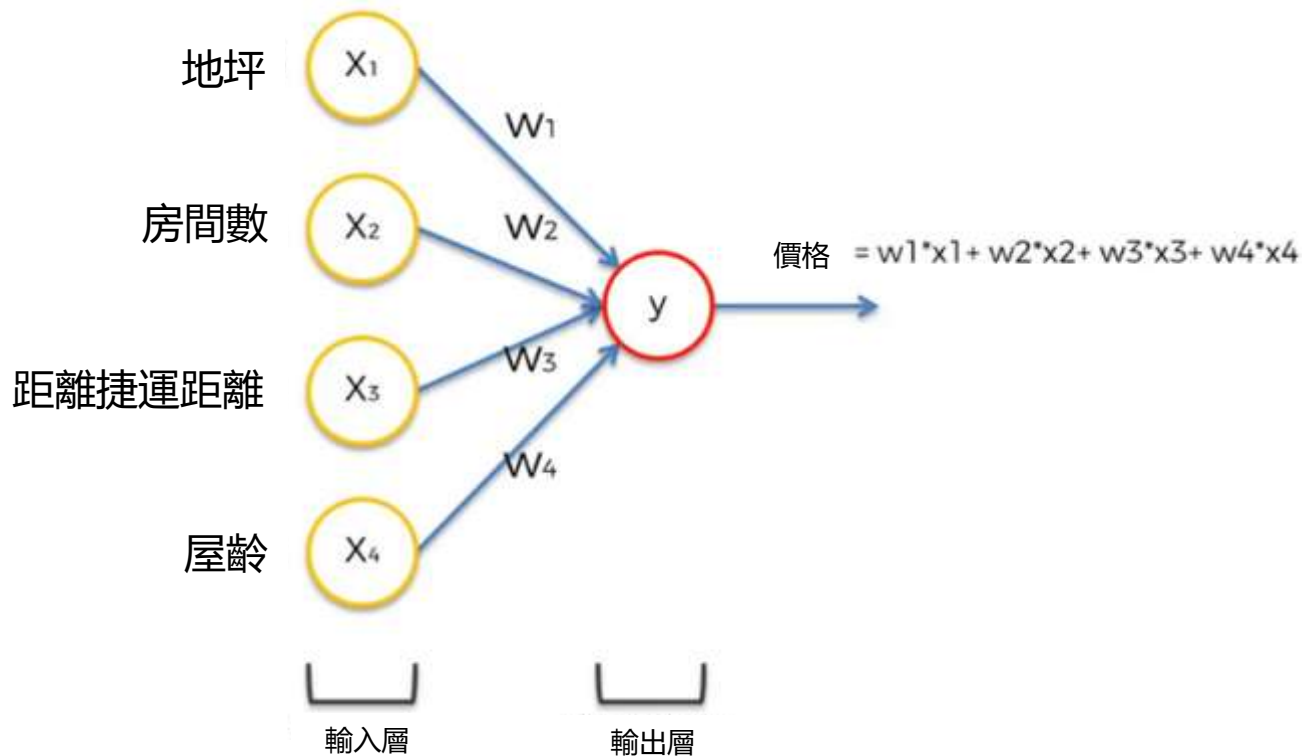
1957 年，Rosenblatt 提出了感知機(Perceptron)模型



1. 加總收集到的訊號
2. 線性或非線性轉換
3. 產生一個新的信號

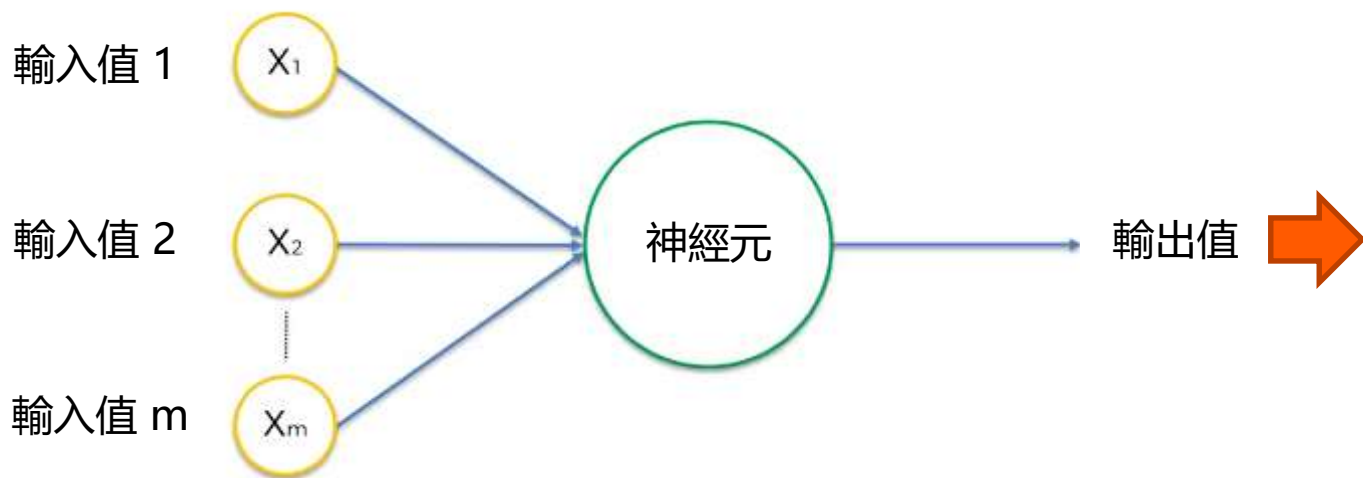
# 單層感知機

假設要預測房屋價格



# 邏輯門

邏輯門是組成數位系統的基本結構，通常組合使用實現更為複雜的邏輯運算



**AND**

a	b	out
0	0	0
0	1	0
1	0	0
1	1	1

**OR**

a	b	out
0	0	0
0	1	1
1	0	1
1	1	1

**XOR**

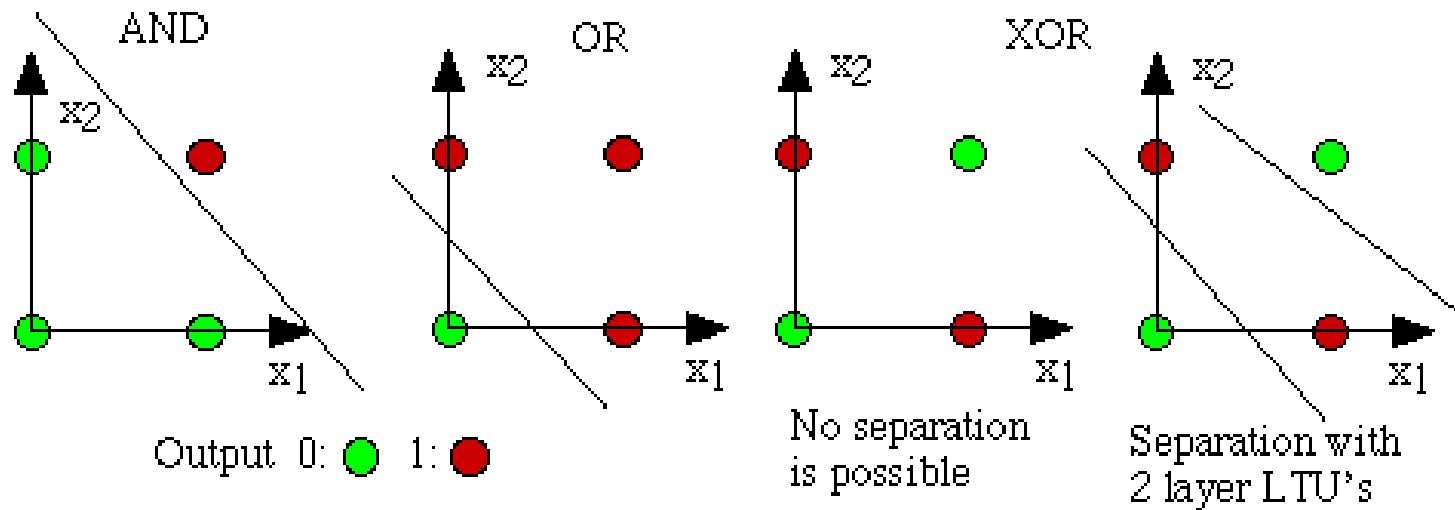
a	b	out
0	0	0
0	1	1
1	0	1
1	1	0

**NOT**

in	out
0	1
1	0

# 單層感知機的缺點

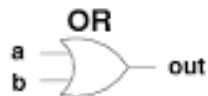
1969年，MIT人工智慧實驗室的Marvin Minsky 和 Seymour 出版了Perceptrons 一書，證明單層感知機無法處理XOR 問題



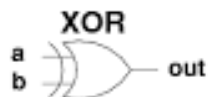
# 利用AND、OR、NOT組合XOR



a	b	out
0	0	0
0	1	0
1	0	0
1	1	1



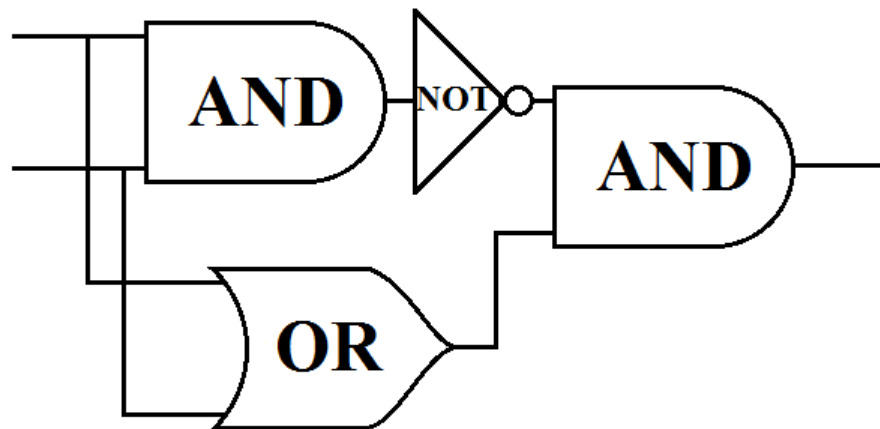
a	b	out
0	0	0
0	1	1
1	0	1
1	1	1



a	b	out
0	0	0
0	1	1
1	0	1
1	1	0

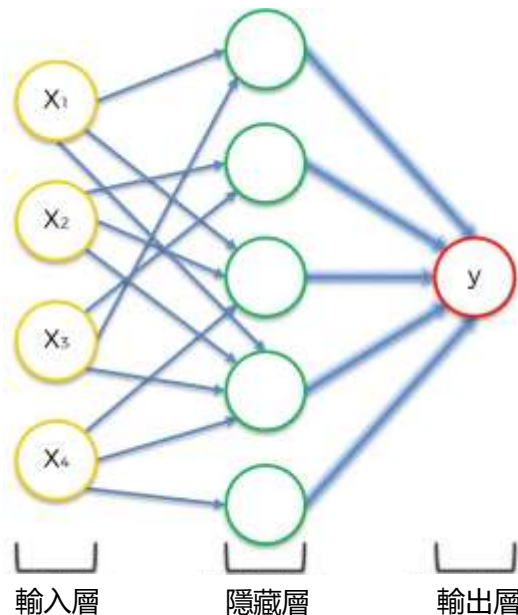


in	out
0	1
1	0



# 多層感知機

1986 年，Rumelhart、Hinton 等人提出「反向傳播演算法」(Backpropagation) 訓練神經網路，催生出具備非線性學習能力的**多層感知機 (Multi-Layer Perceptron)**





# Geoffrey Hinton - 深度學習之父

---



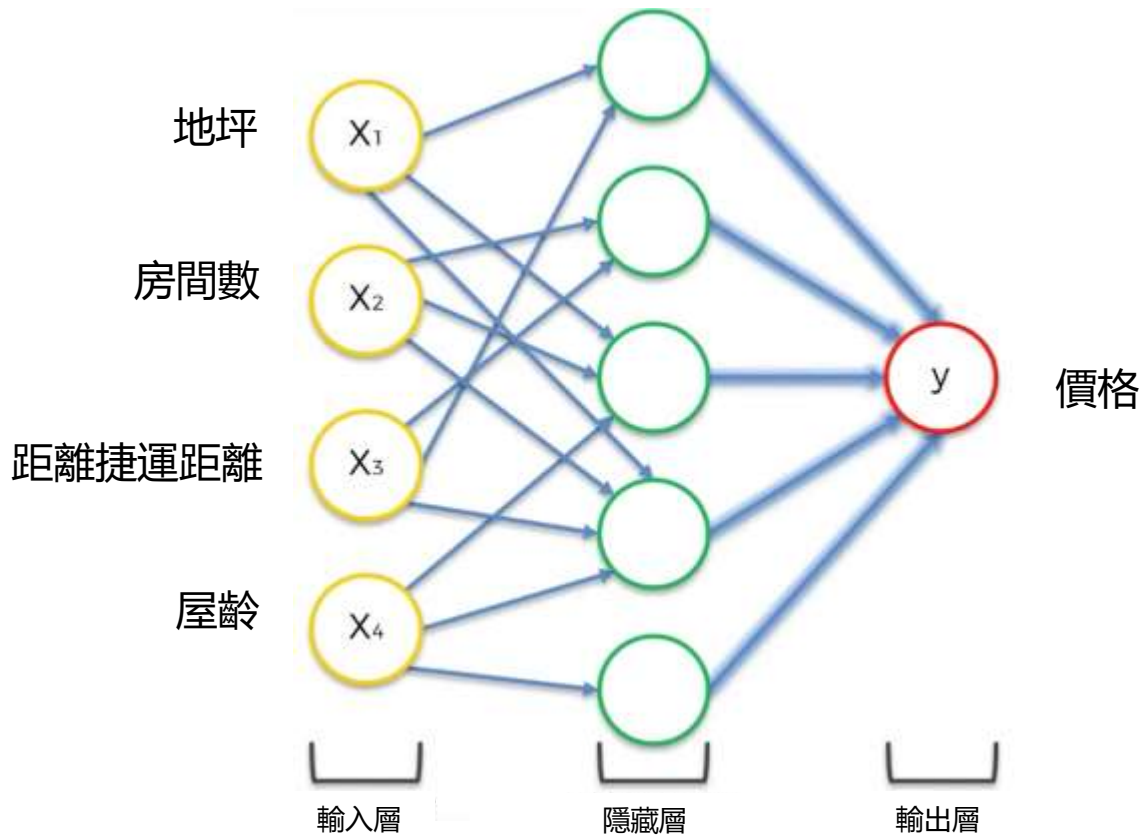
<http://www.cs.toronto.edu/~hinton/>



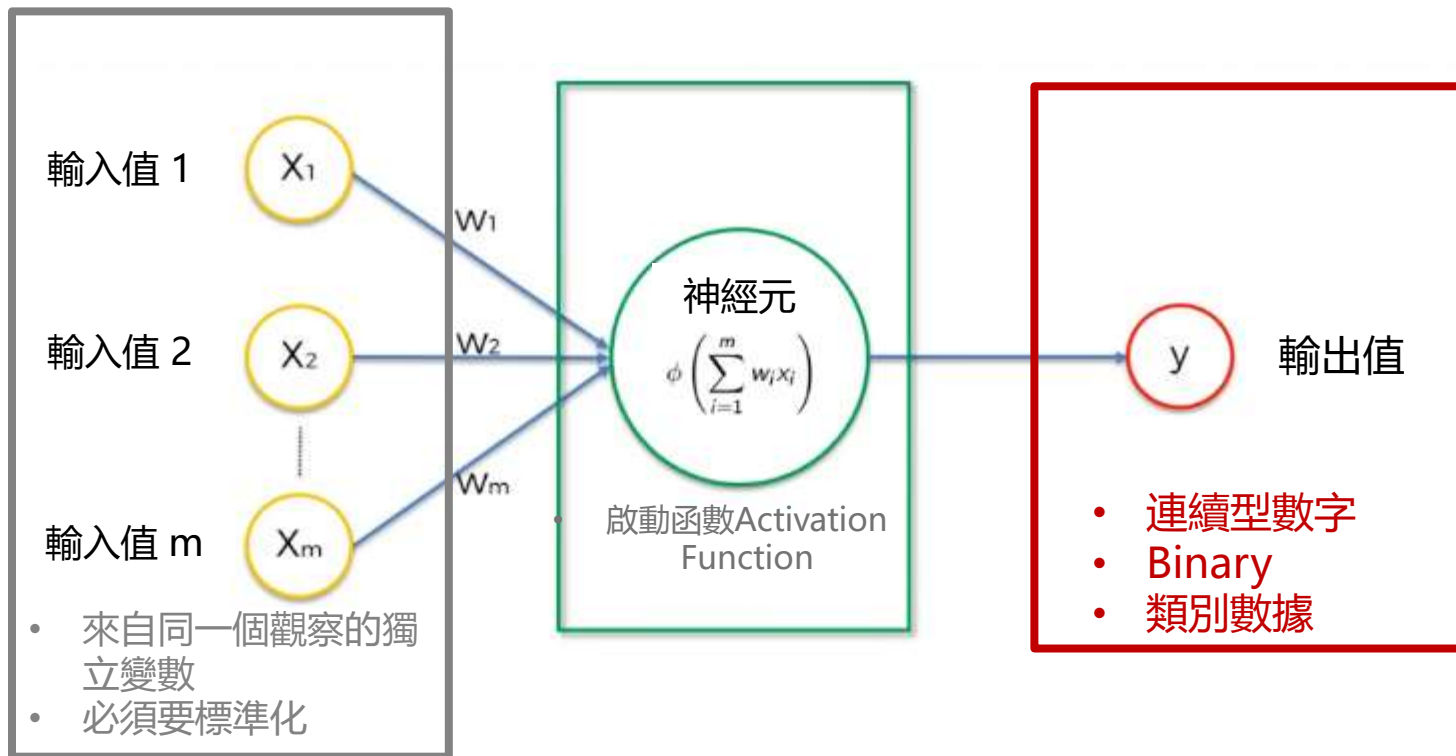
# 建構神經網路

---

# 神經網路



# 使用激活函數調整輸出



# 如同鹿威しの激活函數

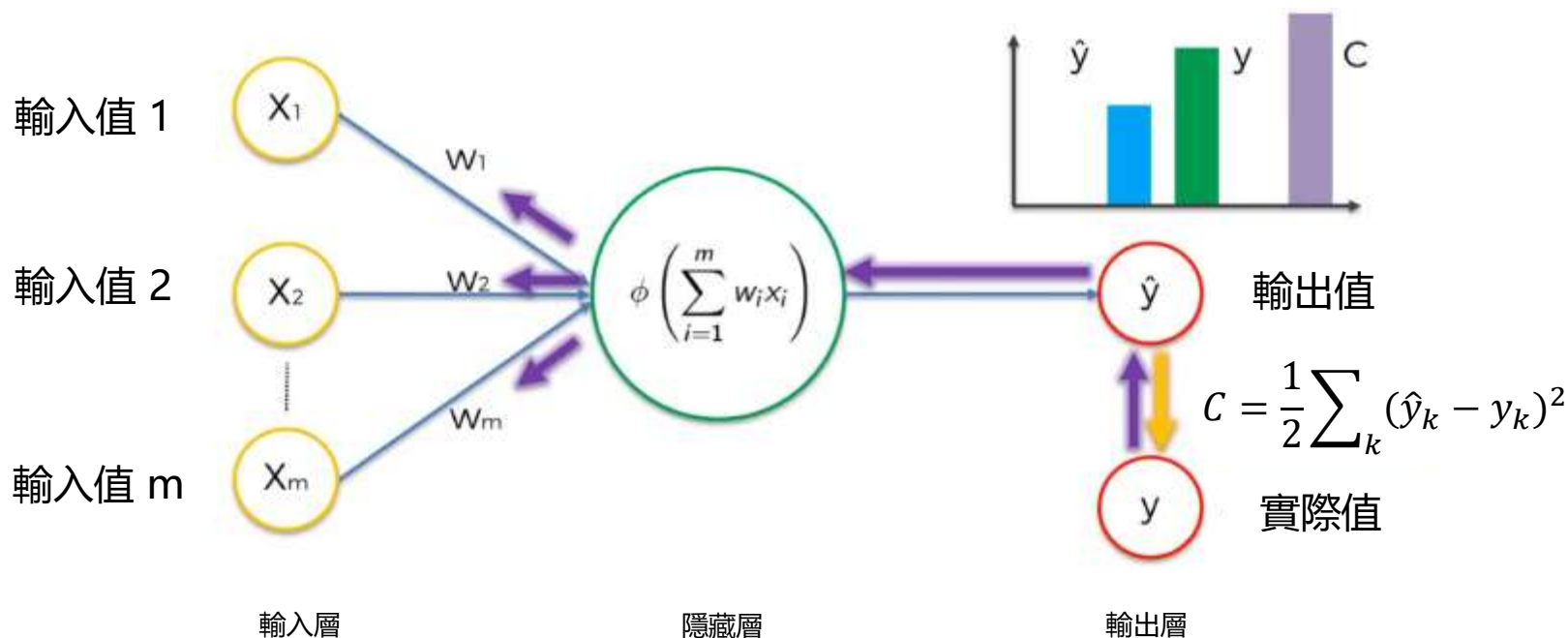
---





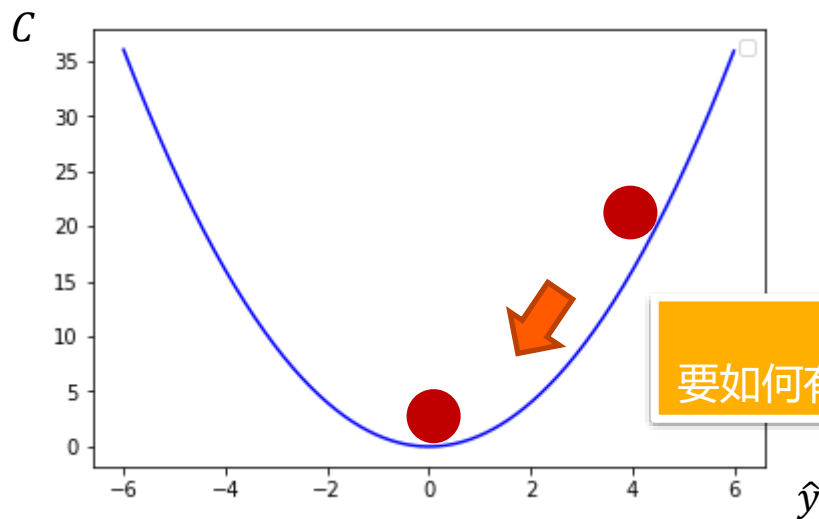
# 類神經網路如何運作

「代價函數」(Cost Function) 或「損失函數」(Loss Function)。  
代價函數是預測結果和真實結果之間的差距



# 梯度法

- Gradient Descent Method  
找出代價函數的最小值
- Gradient Ascent Method  
找出代價函數的最大值

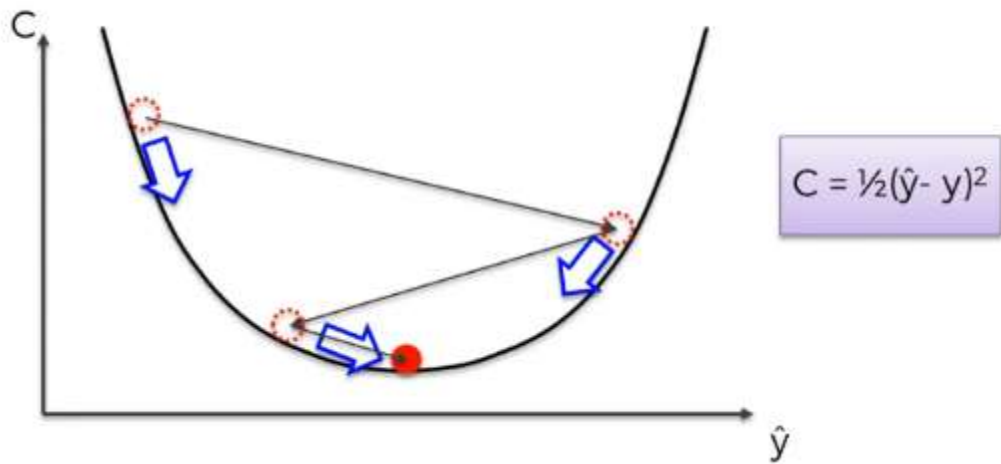


但參數這麼多  
要如何有效搜尋代價函數的最小值

# 計算偏微分

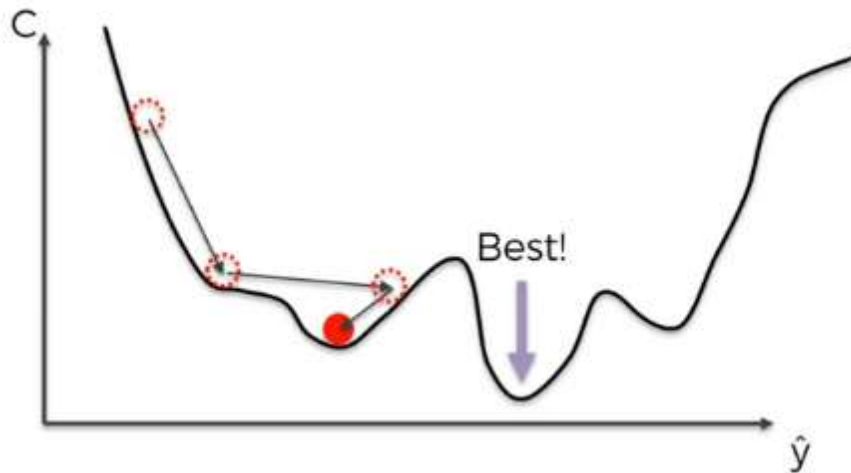
要快速到達目的地，就必須知道要往哪個方向做調整？以及調整幅度的範圍？

- 斜率為正, 往左調整
- 斜率為負, 往右調整
- 斜率越大, 調整幅度越大



# 局部最佳解(Local Minimum)

一路從頭找到最低點可能會陷入局部最佳解(Local Minimum) 等問題  
神經網路只要超過 3 層以上就幾乎沒有效果



「 梯 度 消 失 」  
(Vanishing Gradient)

代價函數為非凸函數，  
求解時容易陷入局部最  
優解、而非全域最優解

# 傳統神經網路只能處理淺層結構

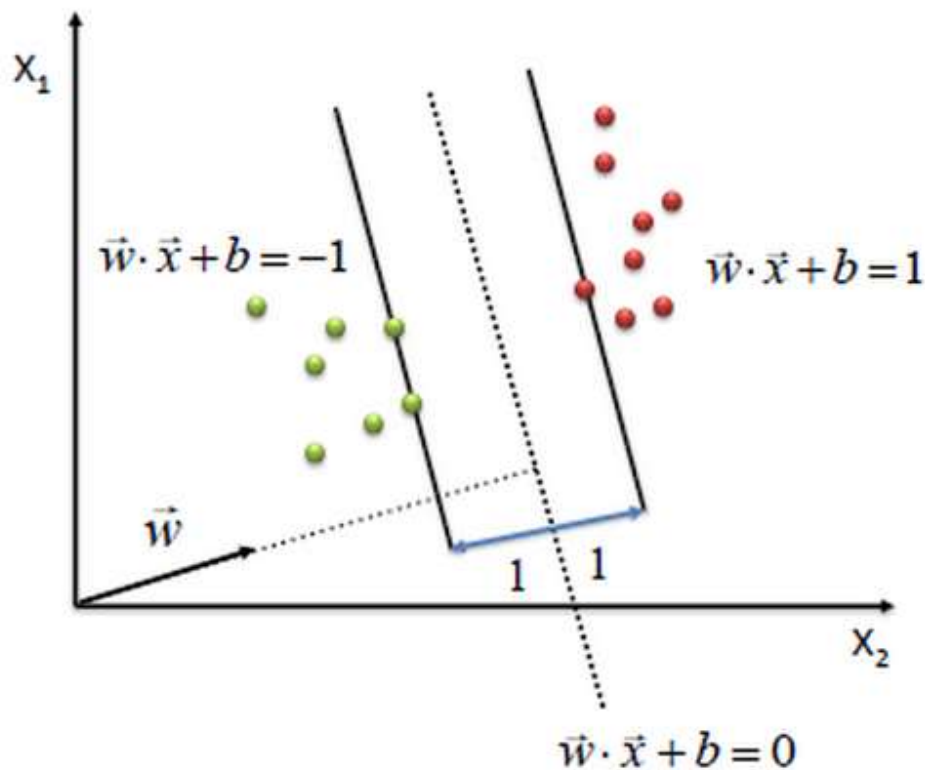
---

梯度消失問題會隨著神經網路層數的增加而更加嚴重，所以傳統類神經網路只能處理淺層結構（比如 2 層）的網路，從而限制了性能

單層感知機、和多層感知機失敗，導致1980的學界認為類神經網路是死胡同

在 1990 年代，支持向量機 (Support Vector Machine, SVM) 等「淺層機器學習模型」成為主流技術，此為機器學習的第二波浪潮

# 支持向量機 (Support Vector Machine)



$$\max \frac{2}{\|\vec{w}\|}$$

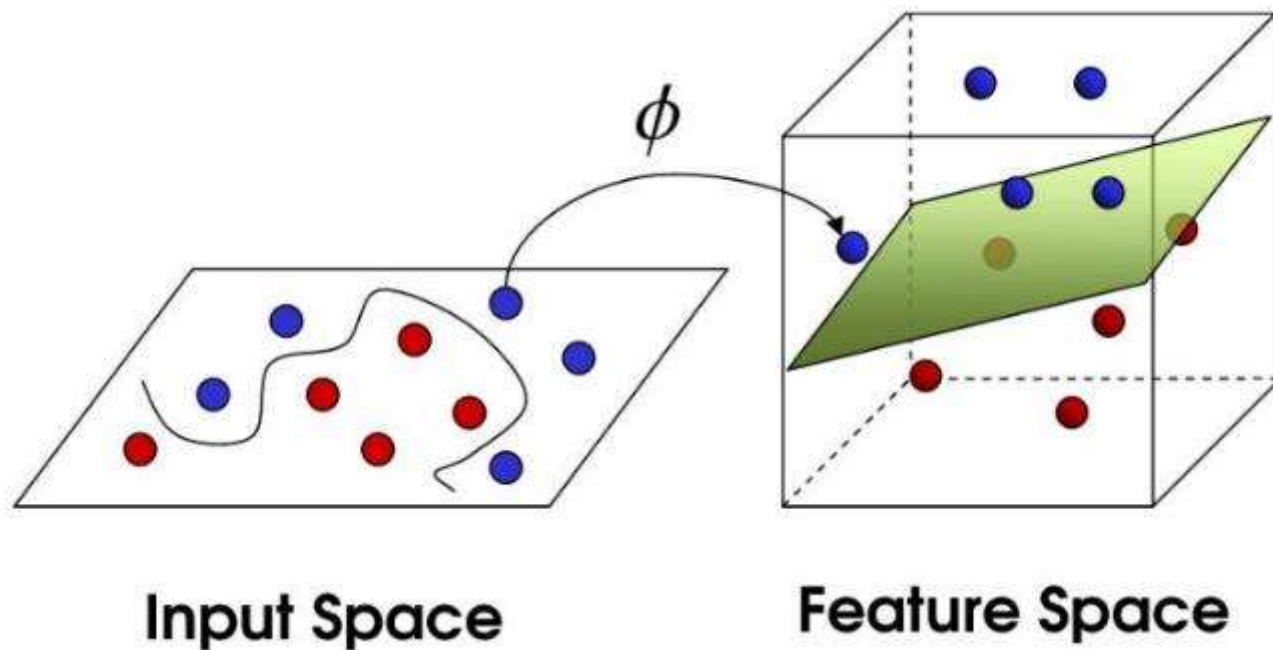
s.t.

$$(\vec{w} \cdot \vec{x} + b) \geq 1, \forall \vec{x} \text{ of class 1}$$

$$(\vec{w} \cdot \vec{x} + b) \leq -1, \forall \vec{x} \text{ of class 2}$$



# 解決高維度資料切分問題





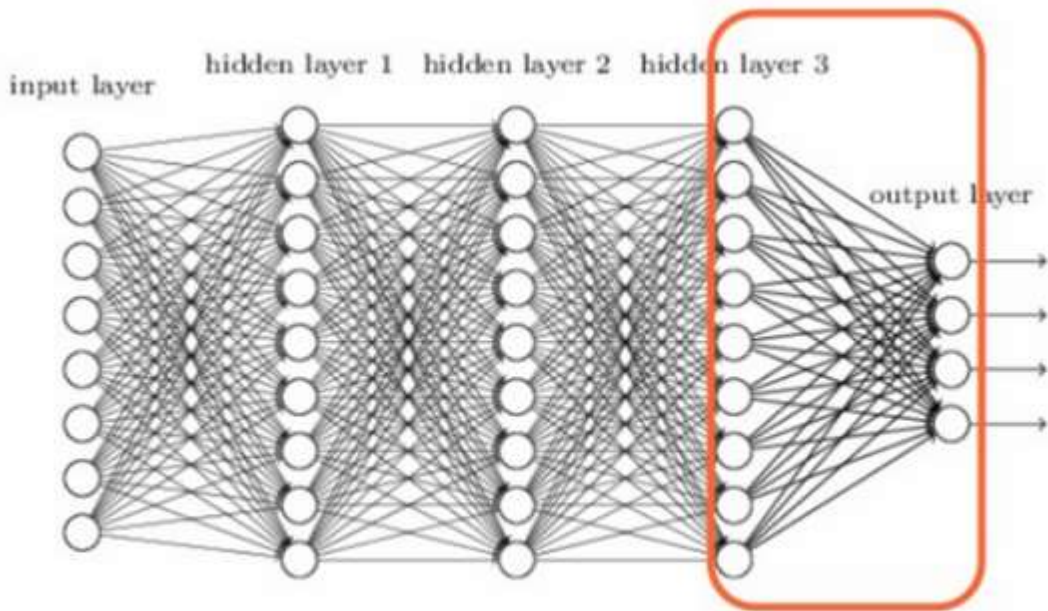
# 深度學習

---

# 類神經網路的曙光

## 2006 – A fast learning algorithm for deep belief nets

Hinton提出用神經網路的非監督式學習來做為神經網路初始權重的指派

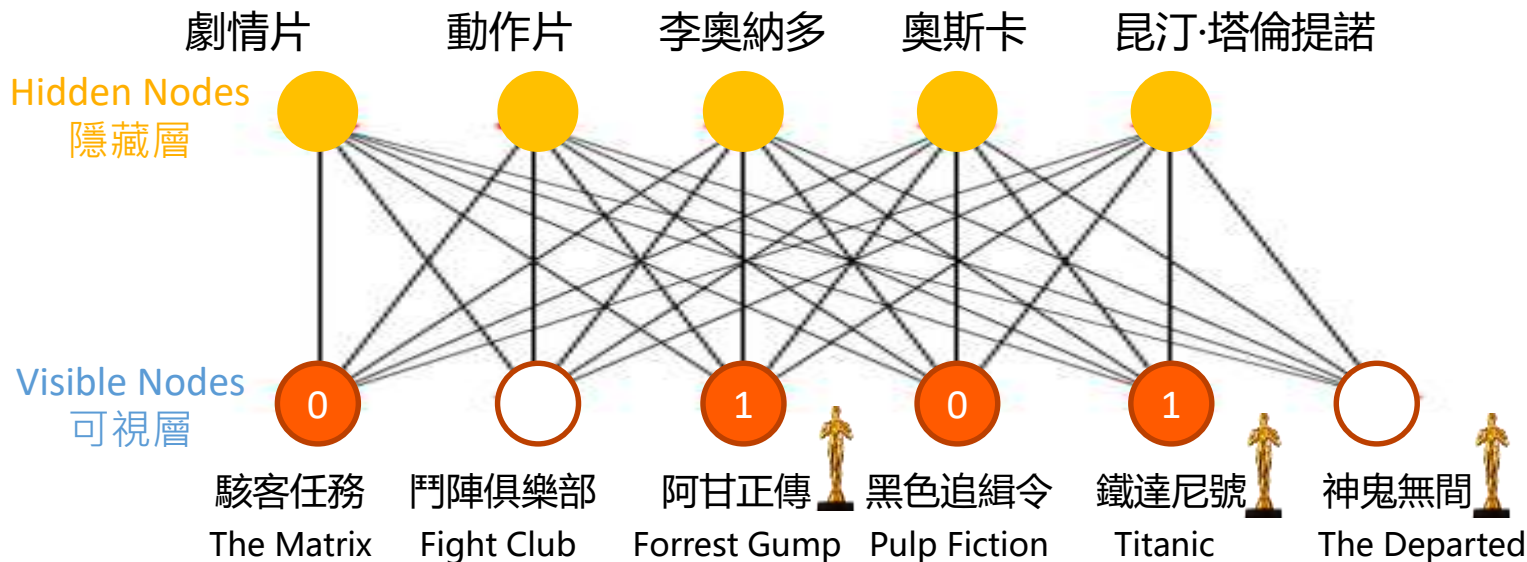


非監督式學習



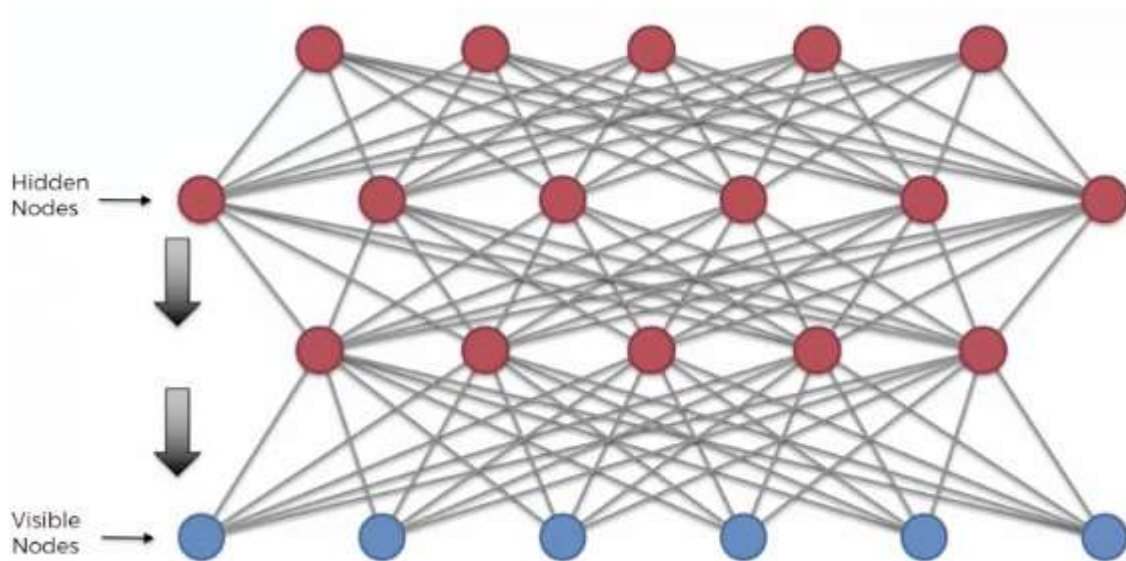
# 限制玻爾茲曼機 (Restricted Boltzmann Machines)

同一層的神經元彼此間沒有連結，不同層的神經元彼此間會連接在一起，並採取隨機決策 (stochastic decisions) 來決定一個神經元要傳導或不傳導



# Deep Belief Networks

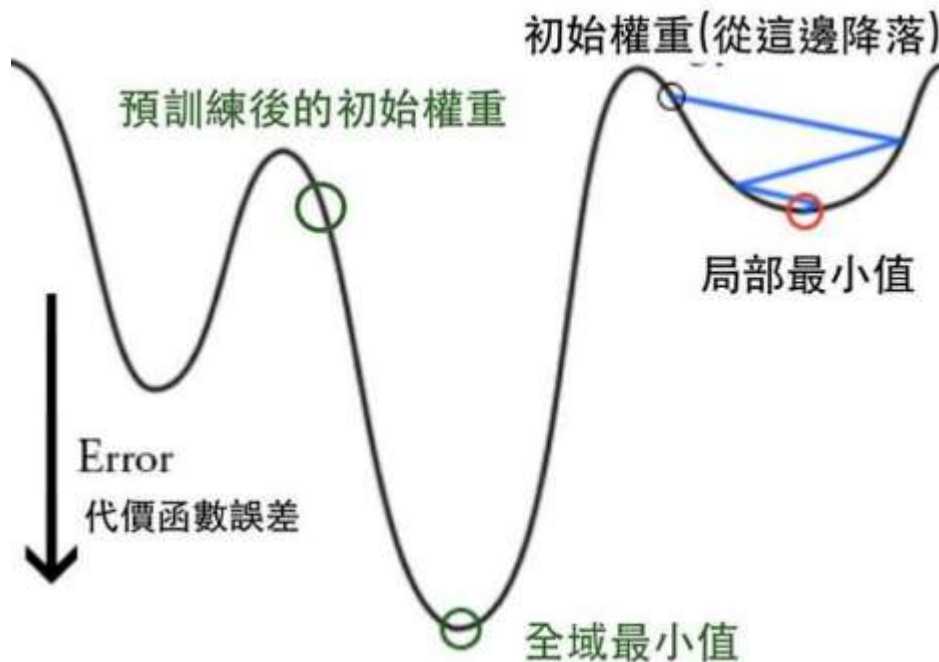
深度信念網路會一次訓練 2 層RBM ；將數個 RBM 模型堆疊起來，每一層模型的輸出值即為下一層模型的輸入值，直到抵達最後一層輸出層為止





# 解決反向傳播的優化問題

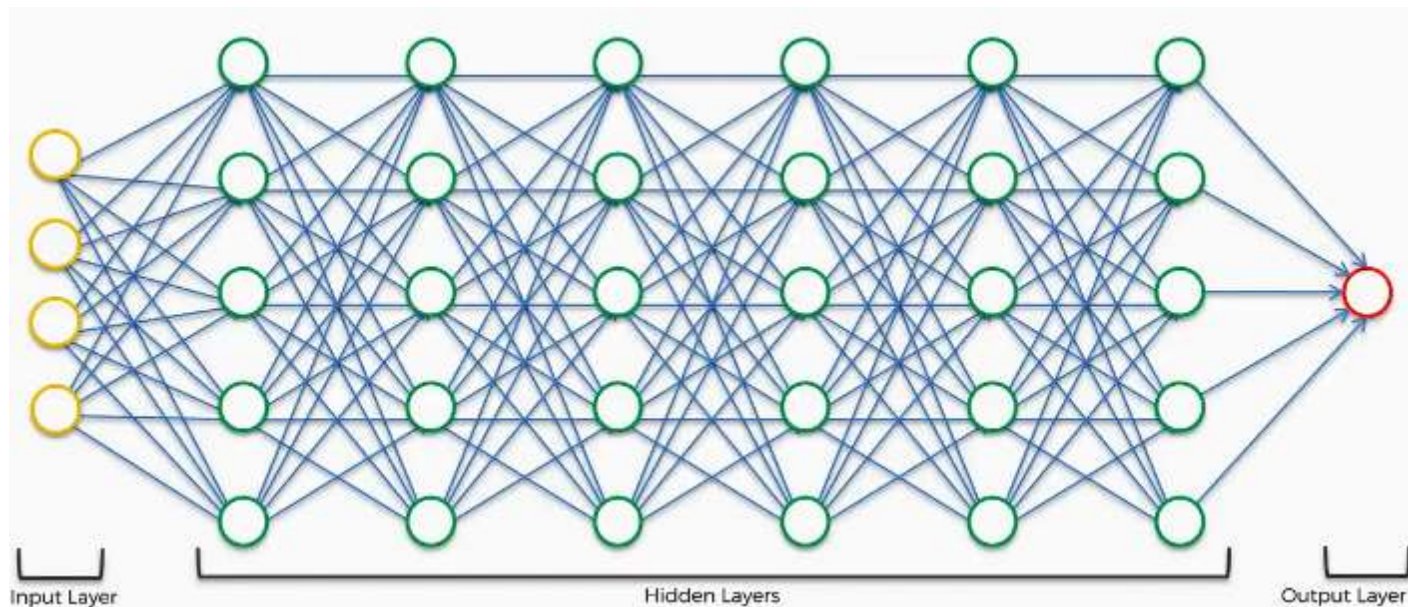
傳統的神經網路隨機初始化網路中的權值，導致網路很容易收斂到局部最小值。找到一個理想的起始點開始梯度下降，將能夠更快、更容易找到全域最小值。





# 深度學習網路

深度學習網路突破以往只能用兩層網路的限制，但由於 Neural Network 長久以來太過惡名昭彰，Hinton 決定把多層的神經網路 (Deep Neural Network) 重命名為深度學習 (Deep Learning)



# ImageNet

IMAGENET

14,197,122 Images, 21841 synsets indexed

[Explore](#) [Download](#) [Challenges](#) [Publications](#) [CoolStuff](#) [About](#)

Not logged in, [Login](#) | [Signup](#)

**ImageNet** is an image database organized according to the **WordNet** hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have an average of over five hundred images per node. We hope ImageNet will become a useful resource for researchers, educators, students and all of you who share our passion for pictures.

[Click here](#) to learn more about ImageNet, [Click here](#) to join the ImageNet mailing list.

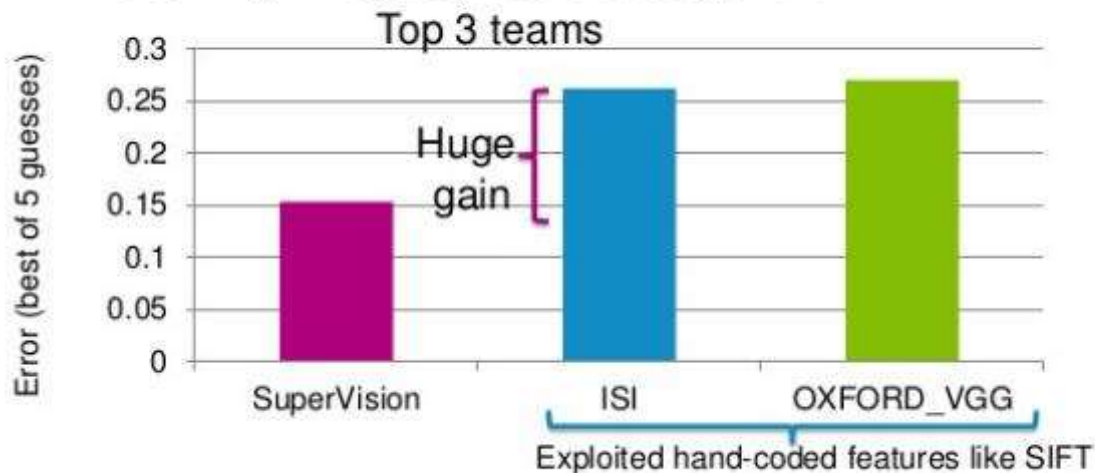


ImageNet 每年都會舉辦圖形識別大賽

# 深度學習在ImageNet 一戰成名

Hinton 的兩個學生以 SuperVision 的隊伍名參賽，以 16.42% 的錯誤率遠勝第二名的 26.22%

ImageNet 2012 competition:  
1.2M training images, 1000 categories



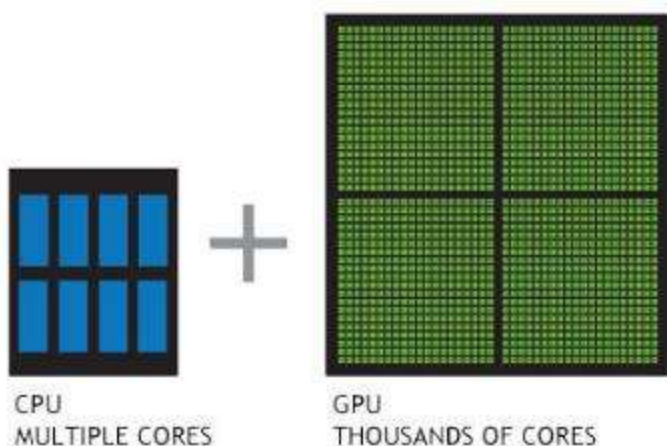
# GPU + 深度學習

---

深度學習會大量用到矩陣運算，最合適的硬體是負責圖形處理的 GPU

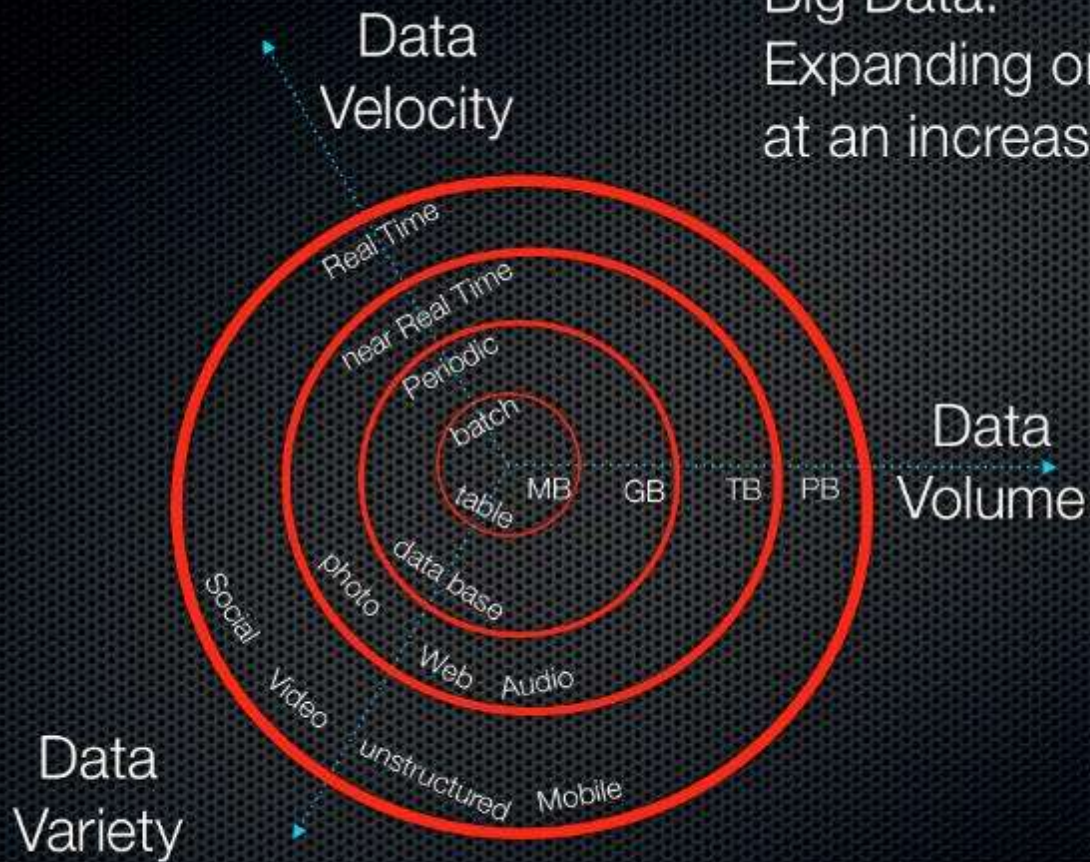
直到 NVIDIA 在 2006 – 2007 年間推出全新運算架構 CUDA — 讓 GPU 成為深度學習運算必用硬體的關鍵。

2012 年 Hinton 的兩位學生利用「深度學習 + GPU」的組合，才真正展現深度學習的威力。

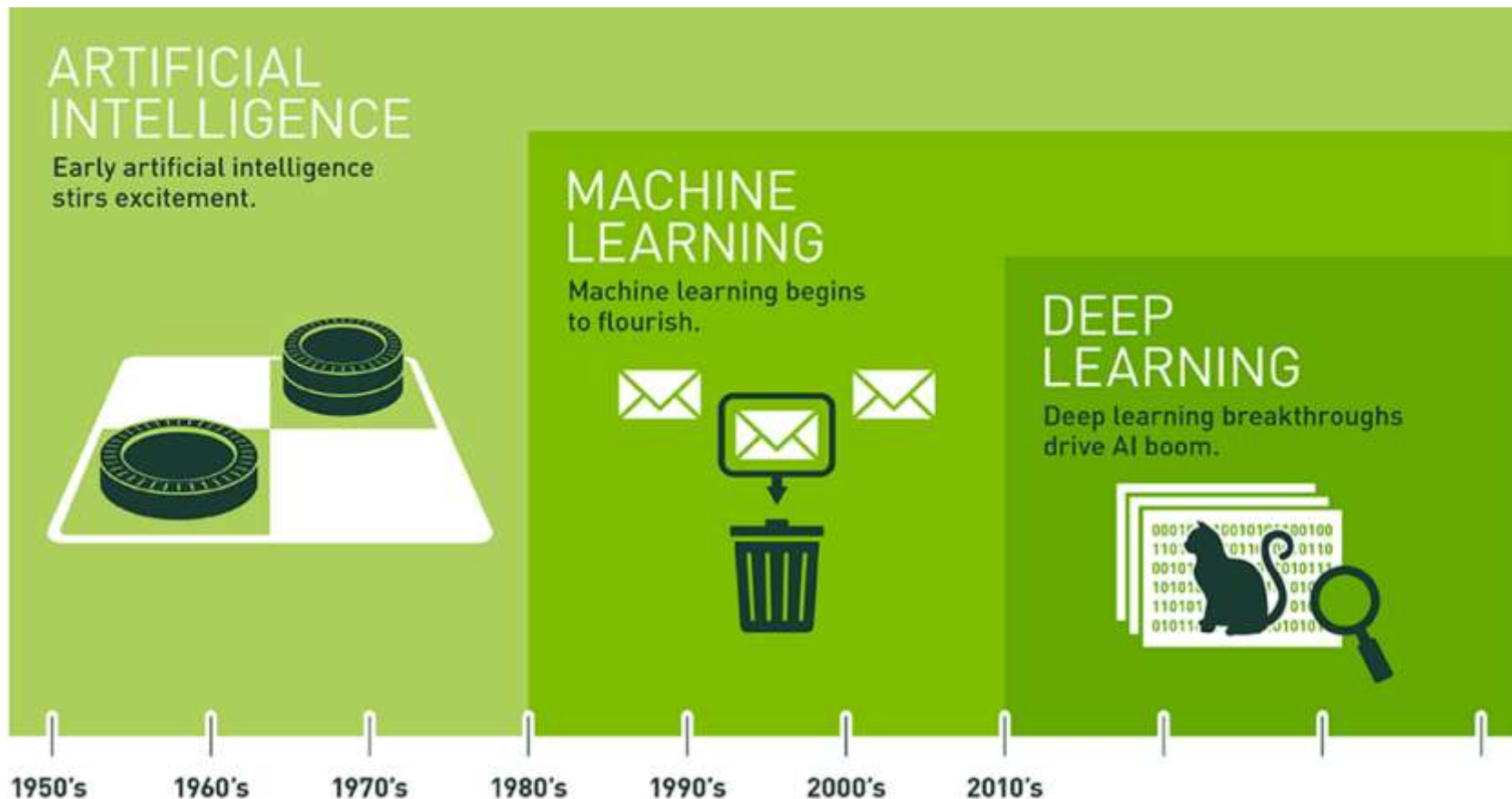




Big Data:  
Expanding on 3 fronts  
at an increasing rate.



# 人工智慧 V.S. 深度學習 V.S. 機器學習



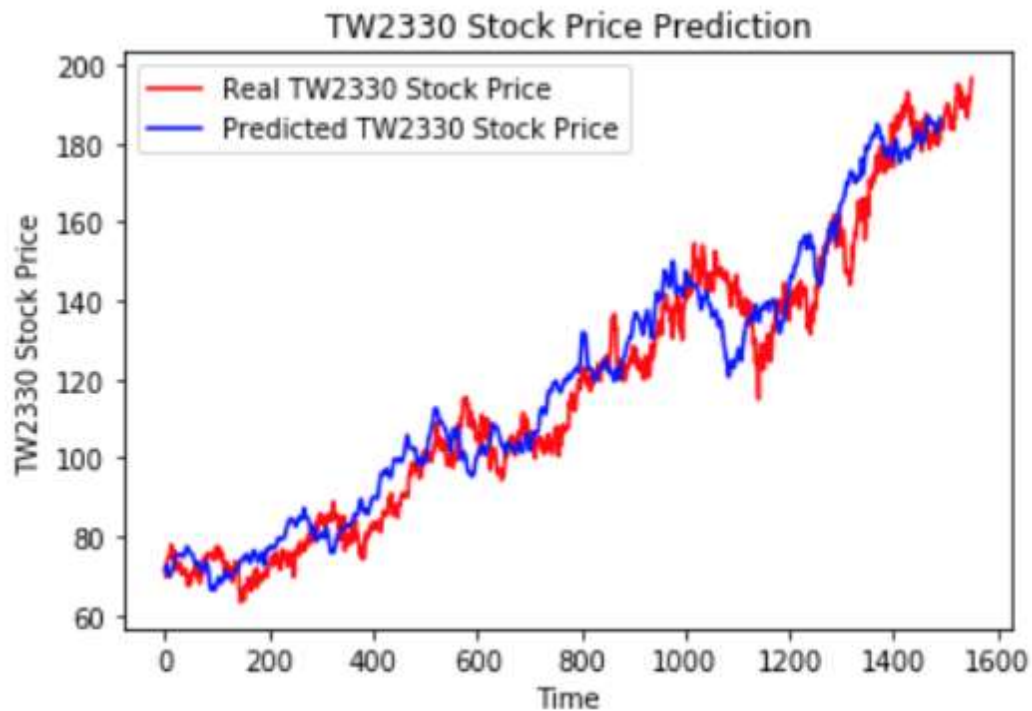


# 人工智慧在金融領域的應用

---



# 股票走勢預測

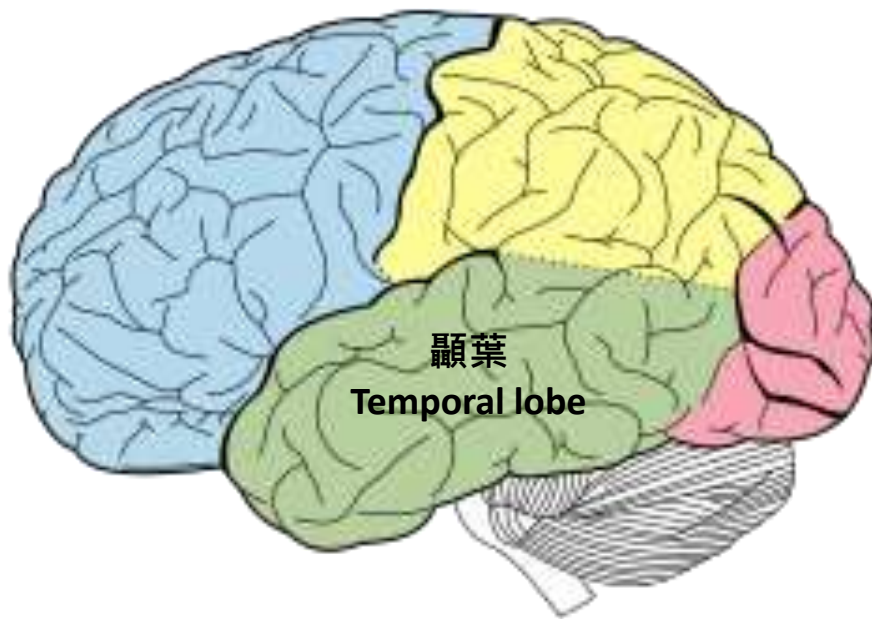


# 大腦的記憶功能

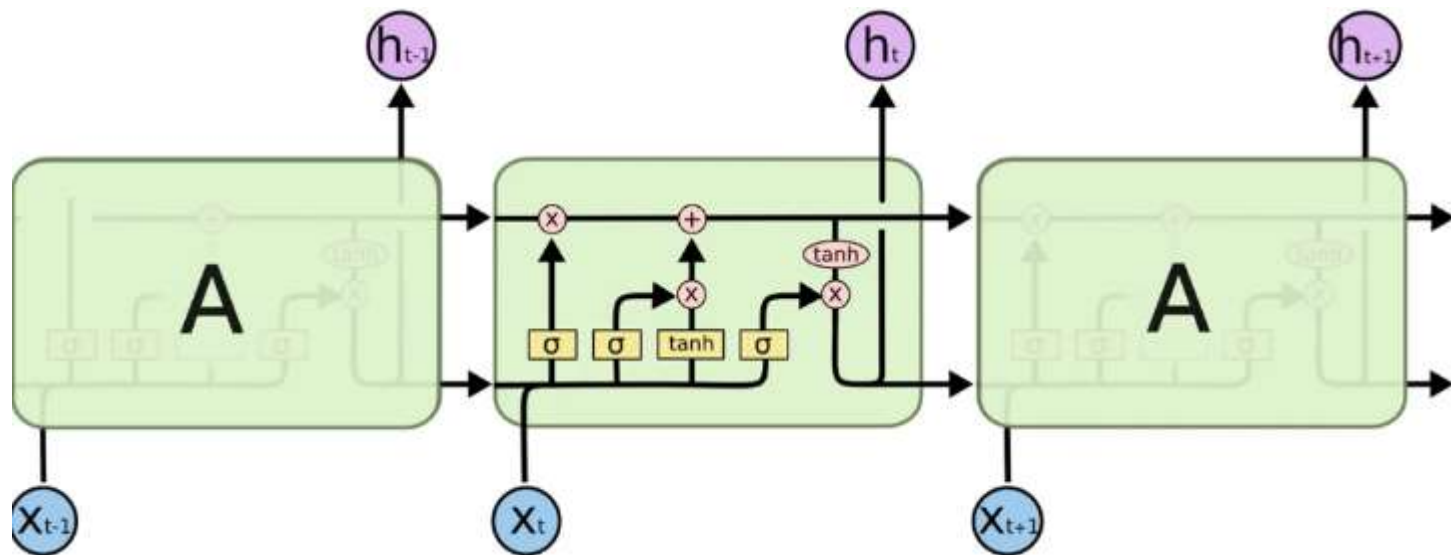
---

顳葉位於兩耳附近的位置，主要提供聽覺與嗅覺上的分辨能力，還有提供新訊息的整理能力

- 右顳葉腦主要在負責視覺化的記憶能力，例如對圖片與臉型的記憶
- 左顳葉腦則負責語言上的記憶能力，例如單字與名字的記憶



# 長短期記憶網路 (Long Short-Term Memory)



# 文字摘要

**Article (truncated):** andy murray came close to giving himself some extra preparation time for his wedding next week before ensuring that he still has unfinished tennis business to attend to . the world no 4 is into the semi-finals of the miami open , but not before getting a scare from 21 year-old austrian dominic *thiem* , who pushed him to 4-4 in the second set before going down 3-6 6-4 , 6-1 in an hour and three quarters . murray was awaiting the winner from the last eight match between tomas berdych and argentina 's juan monaco , prior to this tournament *thiem* lost in the second round of a challenger event to soon-to-be new brit *aljaz* bedene . andy murray pumps his first after defeating dominic *thiem* to reach the miami open semi finals . *murray* throws his *sweatband* into the crowd after completing a 3-6 , 6-4 , 6-1 victory in florida . murray shakes hands with *thiem* who he described as a ' strong guy ' after the game , and murray has a fairly simple message for any of his fellow british tennis players who might be agitated about his imminent arrival into the home ranks : do n't complain , instead the british no 1 believes his colleagues should use the assimilation of the world number 83 , originally from slovenia , as motivation to better themselves .

## Reference Summary:

british no 1 defeated dominic *thiem* in miami open quarter finals .  
andy murray celebrated his 500th career win in the previous round .  
third seed will play the winner of tomas berdych and juan monaco in the semi finals of the atp masters 1000 event in key biscayne .

# Seq2Seq

seq2seq模型最早被用於機器翻譯任務，由於文章摘要與翻譯相似，所以seq2seq模型也被拿來做文章摘要

**seq2seq由三機制組成：**

Encoder部分採用一個單層雙向LSTM，輸入原文的詞向量序列，輸出一個編碼後的隱層狀態序列 $h_i$

Decoder部分採用一個單層單向LSTM，每一步的輸入是前一步預測的詞的詞向量，同時輸出一個解碼的狀態序列 $s_t$ ，用於當前步的預測

Attention是針對原文的概率分佈，目的在於告訴模型在當前的預測過程中，原文中的哪些詞更重要

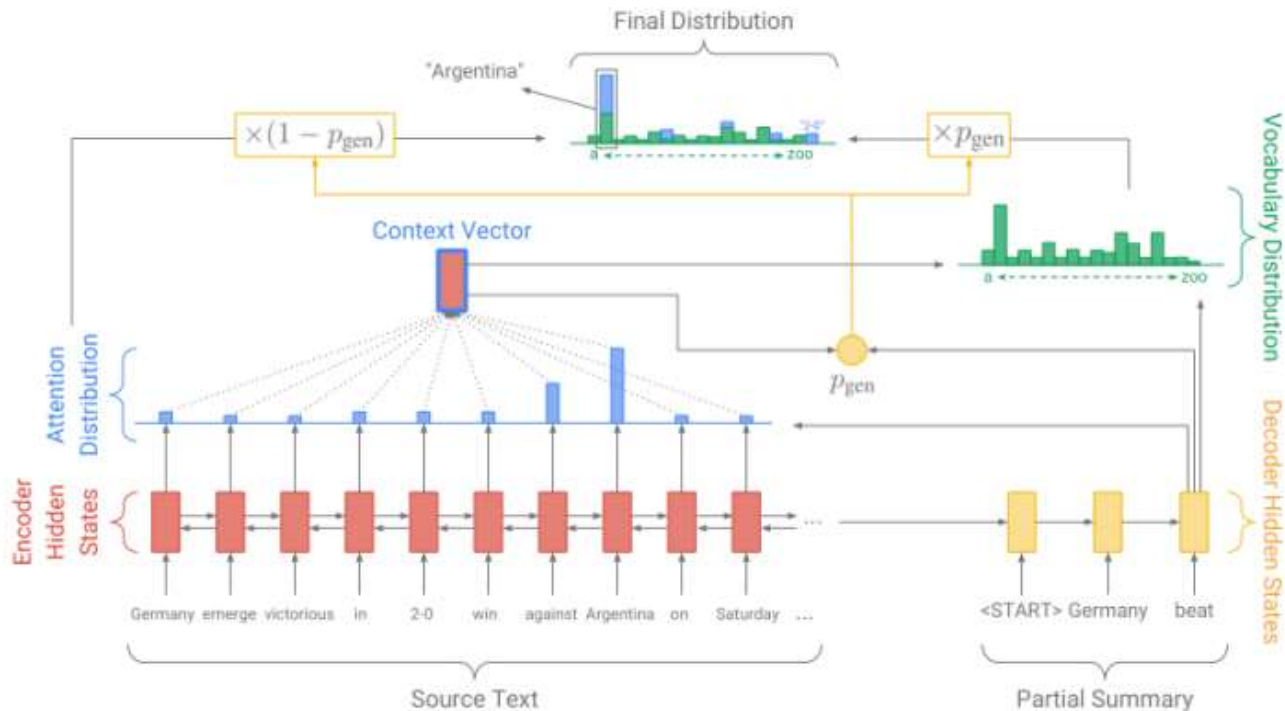
# Pointer-generator network

---

seq2seq模型和pointer network的混合模型，一方面通過seq2seq模型保持文字生成的能力，另一方面通過pointer network直接從原文中取詞，提高摘要的準確度

pointer-generator network相當於在每次摘要生成過程中，都會把原文動態地加入到詞表中去，在每一步的預測過程中，相比於單純的seq2seq模型，選原文中出現的詞作為摘要的概率要更大一些。

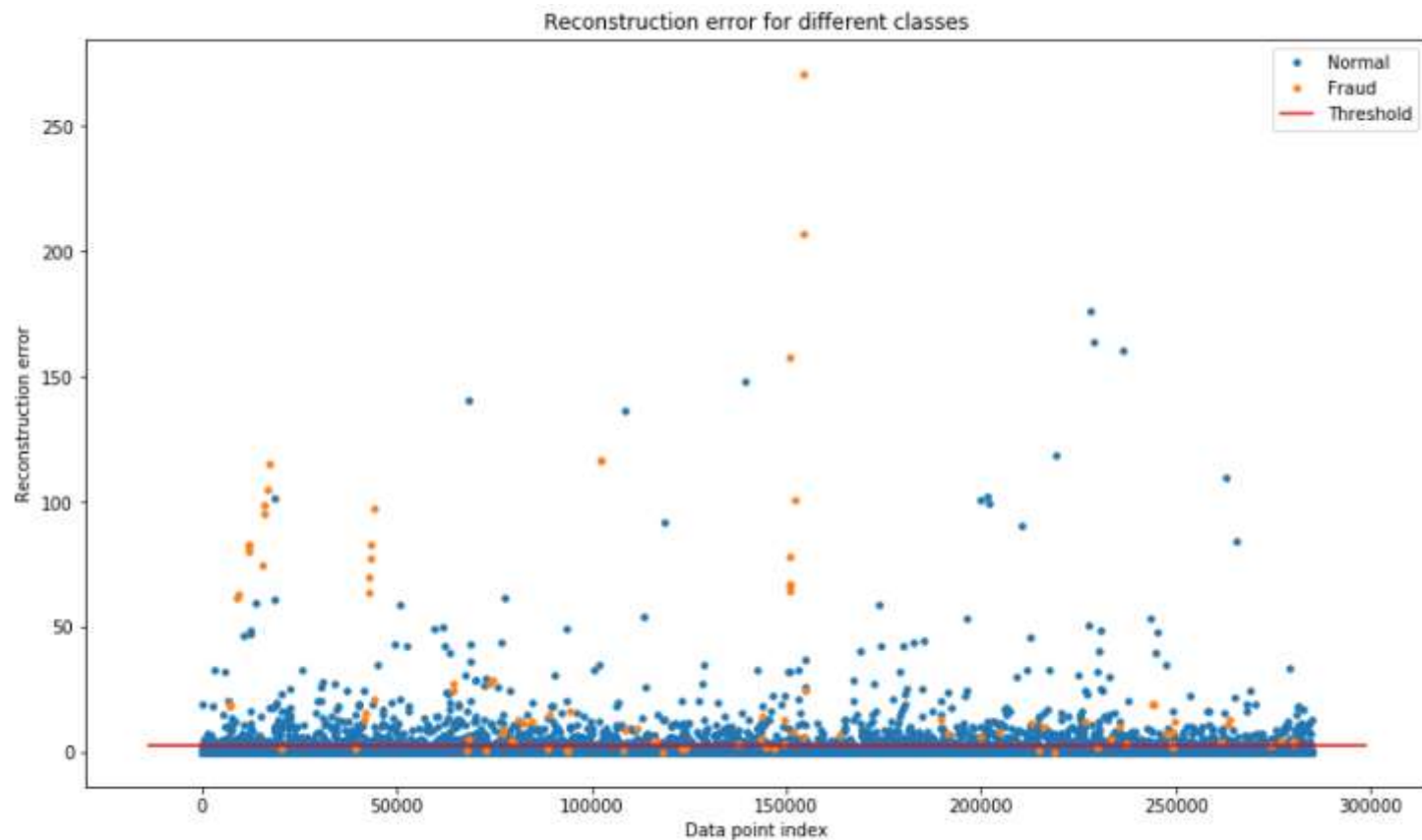
# 文字摘要



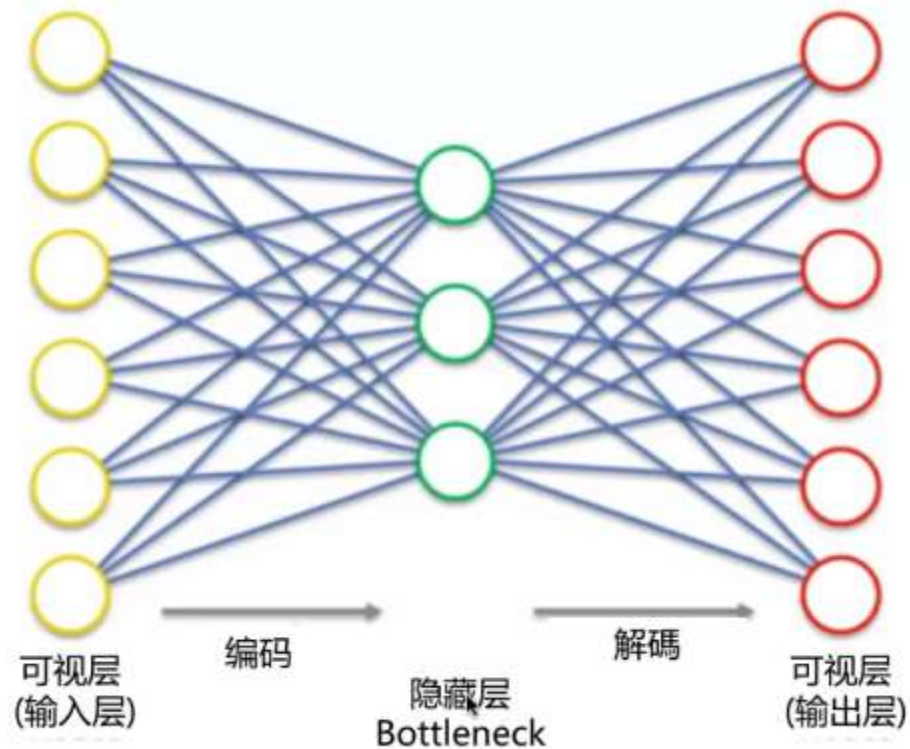
Get To The Point: Summarization with Pointer-Generator Networks, 2017



# 詐欺偵測



# 自編碼網路





# 強化學習

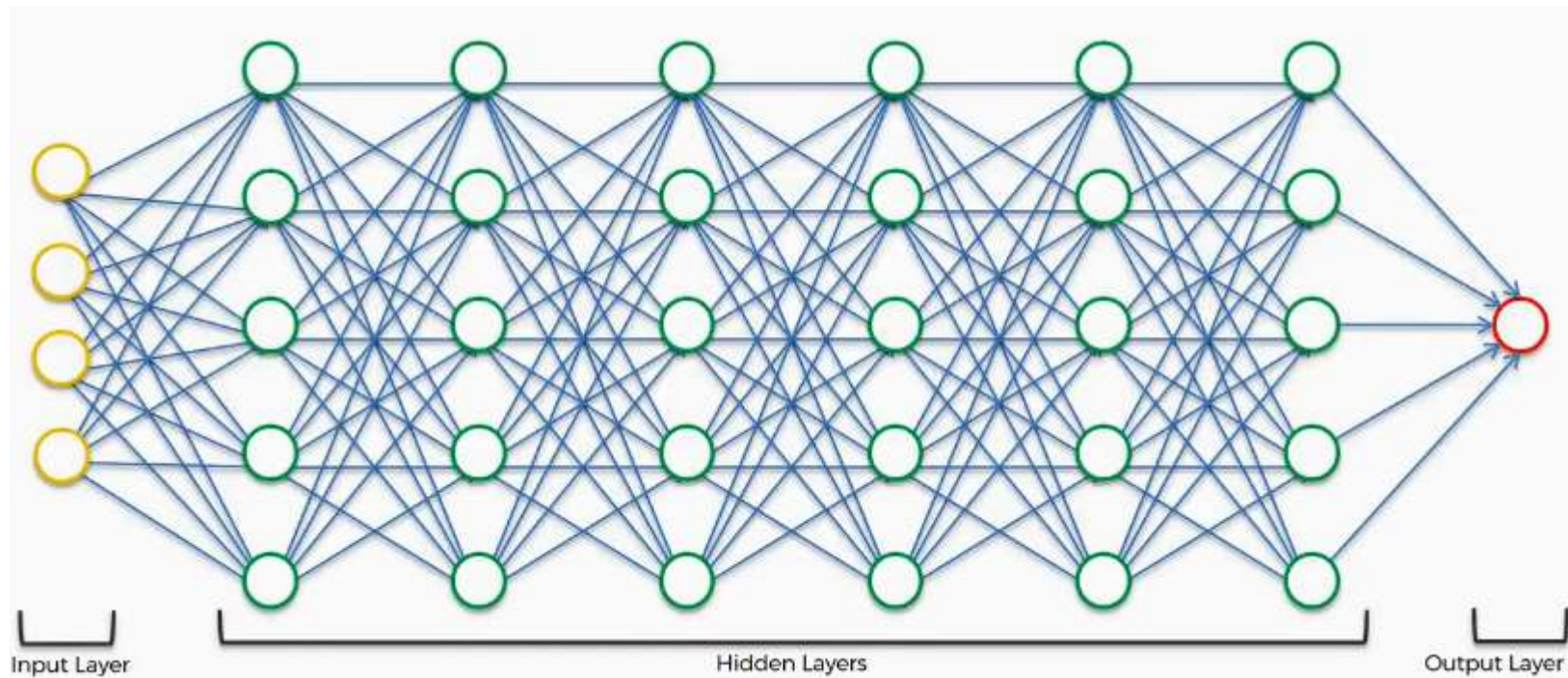
---

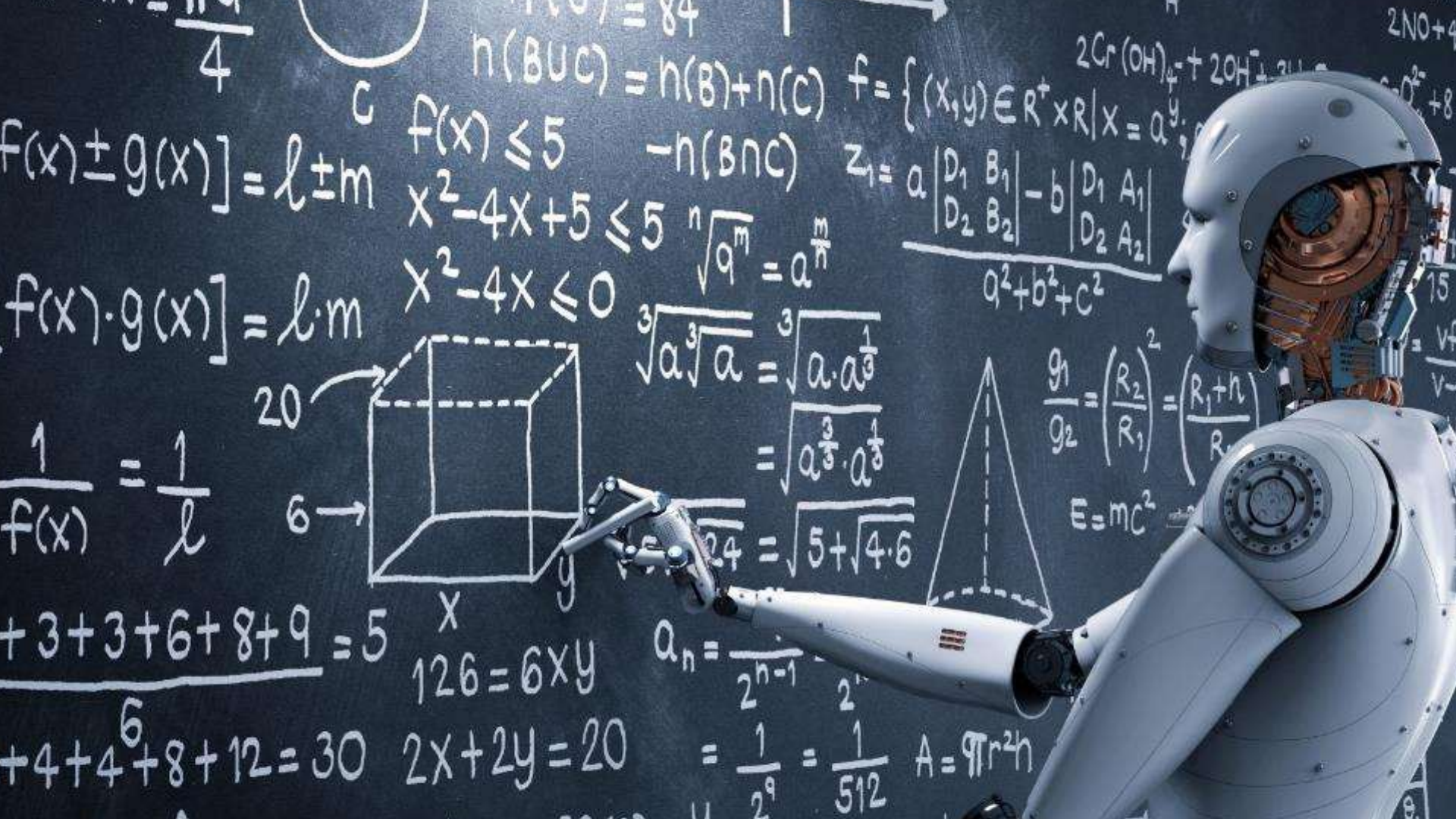




# 深度學習

機器學習的一部份，可以實作監督式學習與非監督式學習





# 深度Q網路

---

DeepMind 於2014 年展示深度Q網路 (DQN, Deep-Q-Network) , 讓電腦自動學習雅達利 (Atari) 49 種遊戲

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529.









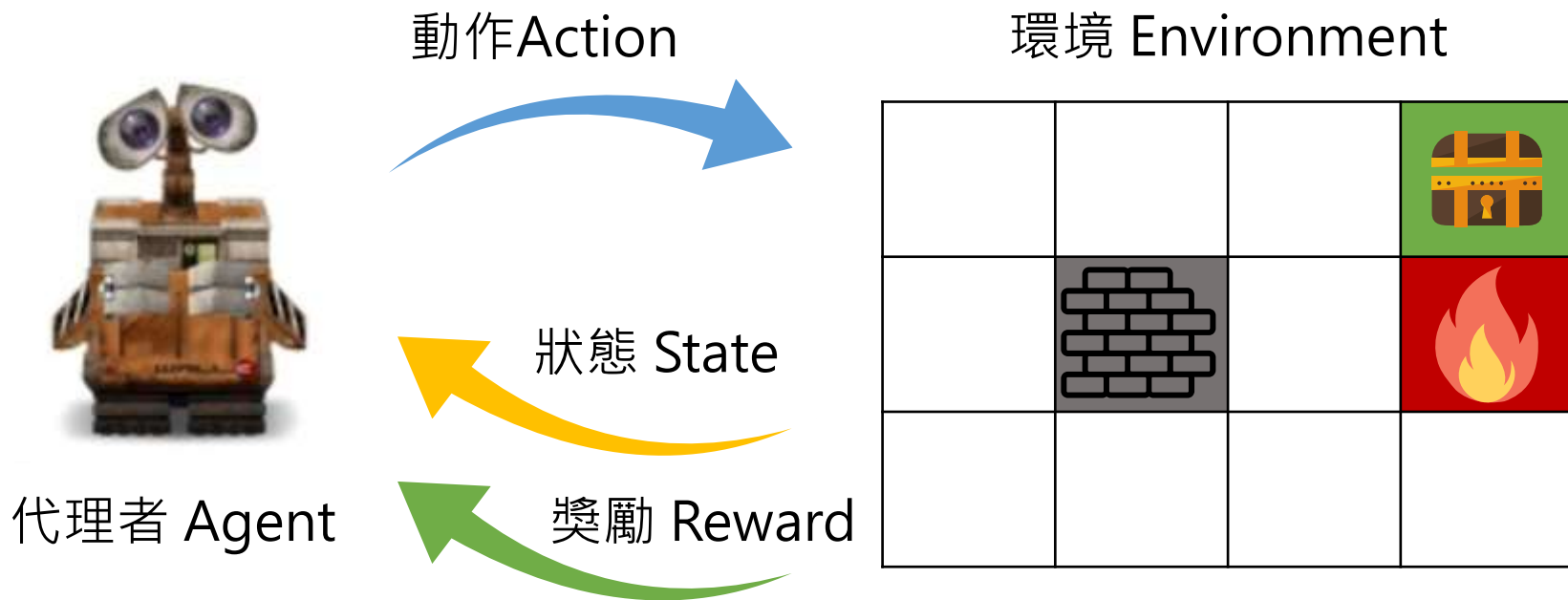
# 什麼是強化學習

---

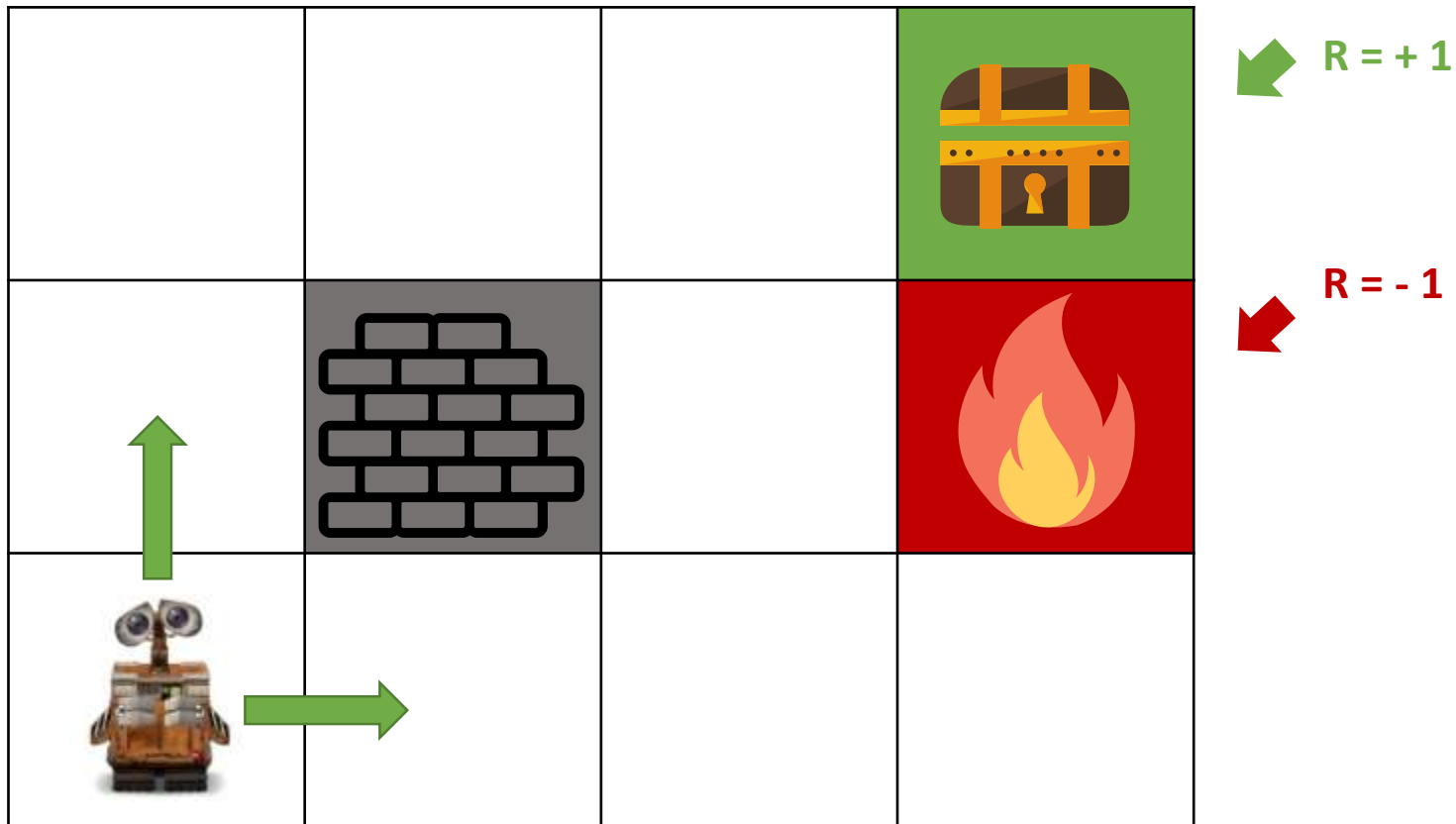


# 強化學習架構

- 代理者在環境中選擇做出某個動作以得到對應的獎勵
- 代理者選擇動作的方式稱為策略(Policy)



# 與環境互動過程





# 貝爾曼方程 ( Bellman Equation )

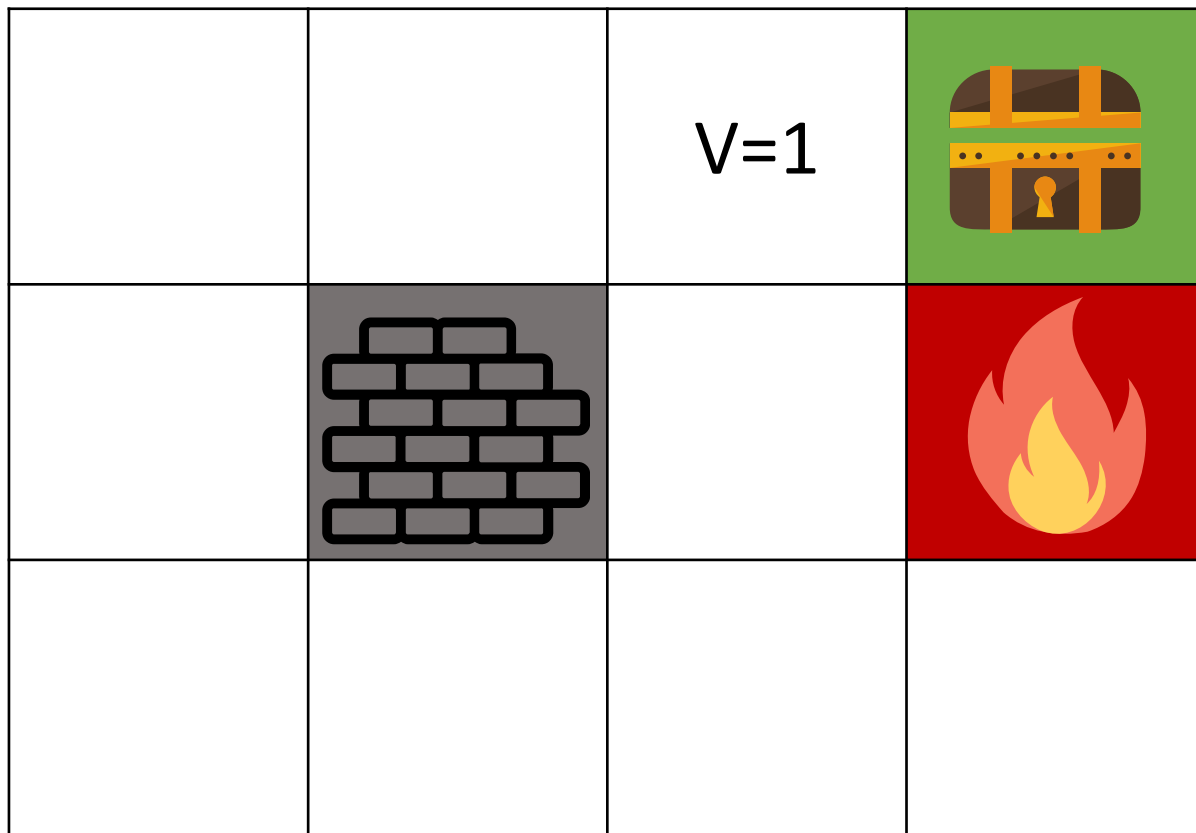
貝爾曼方程 ( Bellman Equation ) 也被稱作 動態規劃方程  
( Dynamic Programming Equation )



An optimal trajectory on the time interval  $[T_1, T]$  must be optimal also on each of the subintervals  $[T_1, T_1 + \epsilon]$  and  $[T_1 + \epsilon, T]$ .



# 尋找最佳路徑


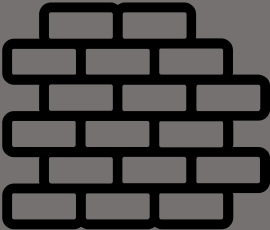



↖ R = + 1

↘ R = - 1



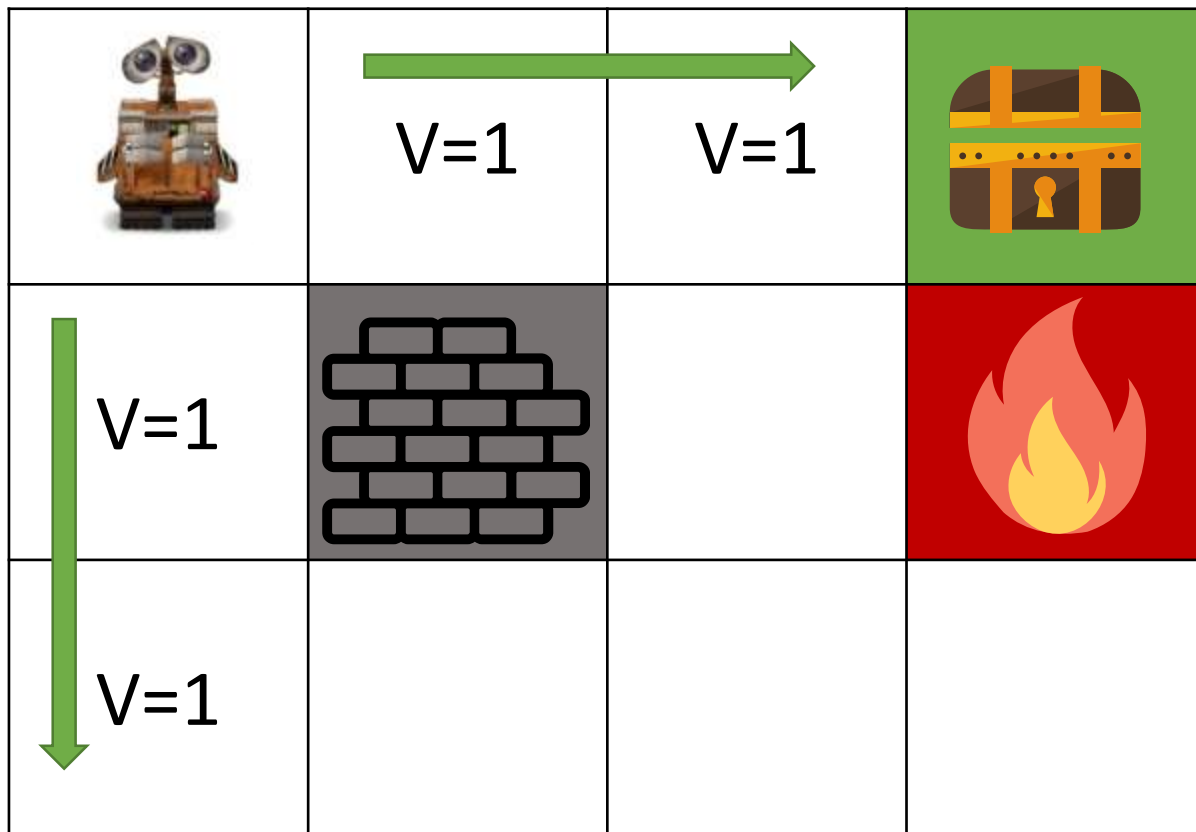
# 尋找最佳路徑

V=1	V=1	V=1	
V=1			
V=1			

↖ R = + 1

↘ R = - 1

# 尋找最佳路徑



# 貝爾曼方程 ( Bellman Equation )

---

$$V(s) = \max_a (R(s, a) + \gamma V(s'))$$

S : 狀態 State



a : 動作 Action

R : 獎勵 Reward

$\gamma$  : 折扣 Discount

# 寻找最佳路径

假设  $\gamma = 0.9$

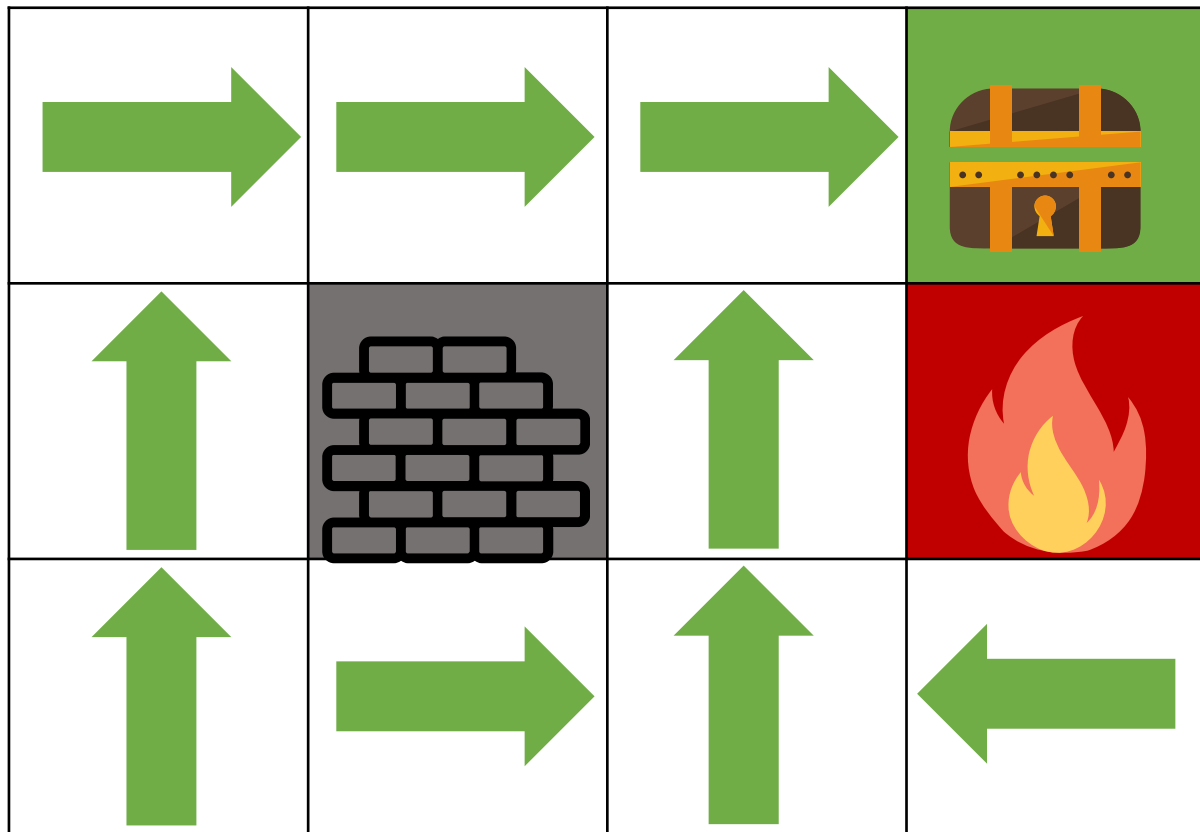
$V=0.81$	$V=0.9$	$V=1$	
$V=0.73$		$V=0.9$	
$V=0.66$	$V=0.73$	$V=0.81$	$V=0.73$



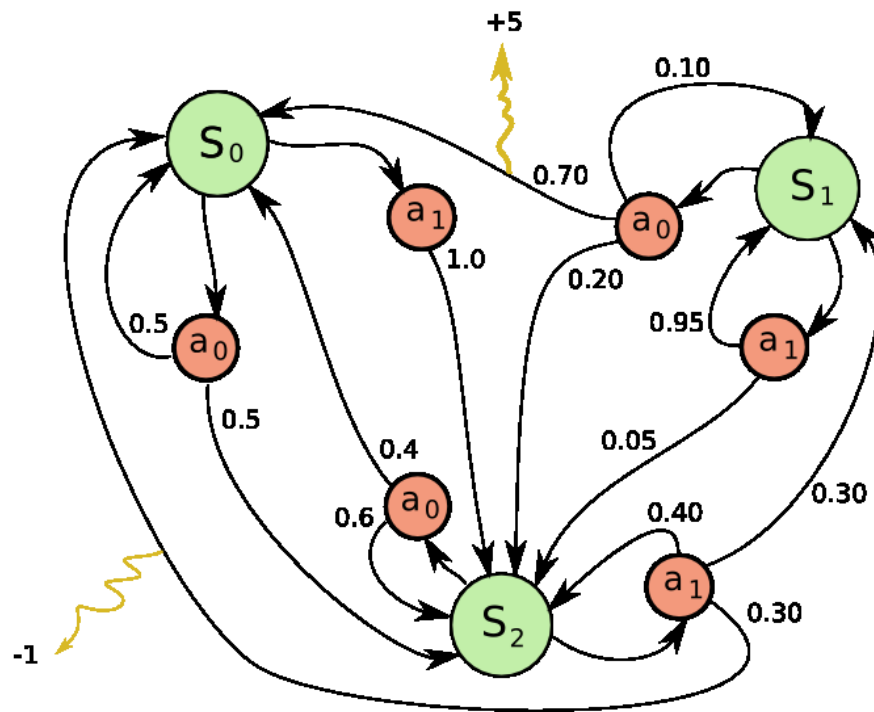
# 馬可夫決策過程

---

# 確定性策略

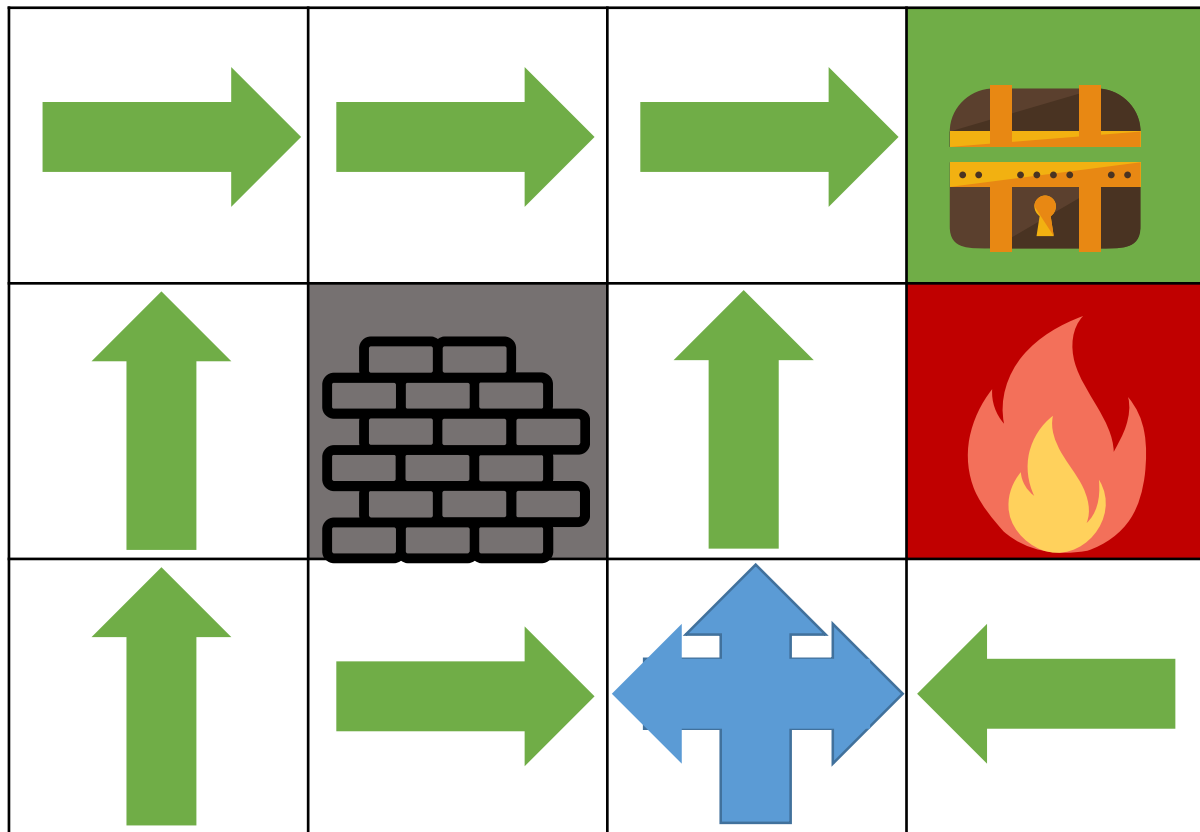


# 馬可夫決策過程





# 隨機策略




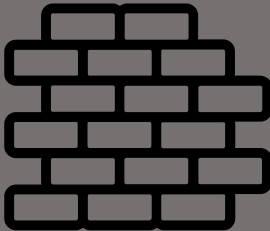

# 貝爾曼方程 ( Bellman Equation )

---

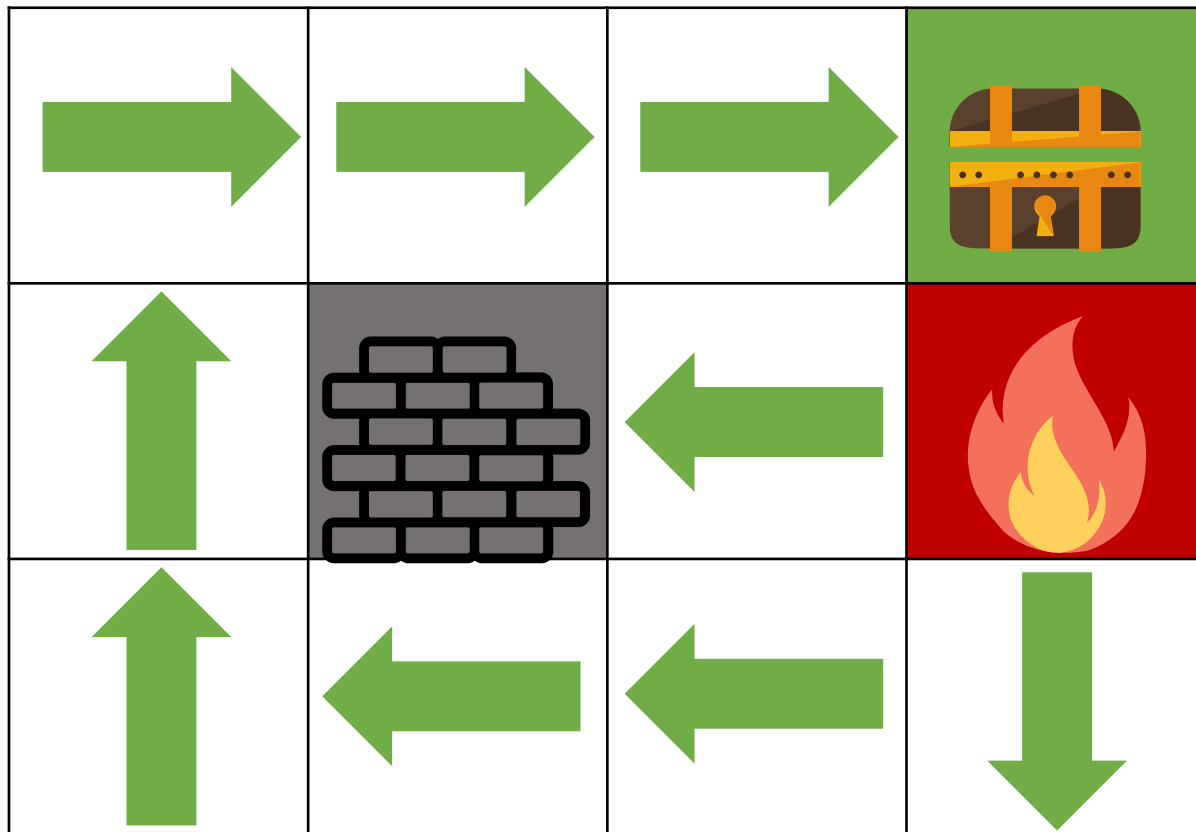
$$V(s) = \max_a (R(s, a) + \gamma \sum_{s'} P(s, a, s') V(s'))$$

- P : 機率 Probability
- S : 狀態 State
- a : 動作 Action
- R : 獎勵 Reward
- $\gamma$  : 折扣 Discount

# 隨機策略

$V=0.7$	$V=0.75$	$V=0.85$	
$V=0.63$		$V=0.36$	
$V=0.55$	$V=0.45$	$V=0.3$	$V=0.2$

# 隨機策略路徑





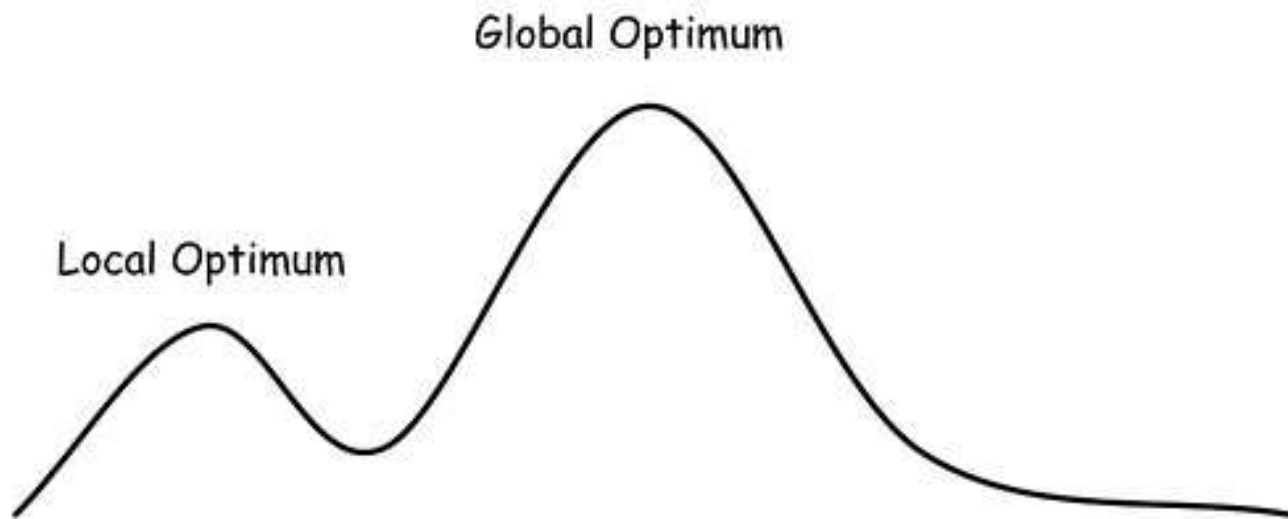
# 未來報酬

---


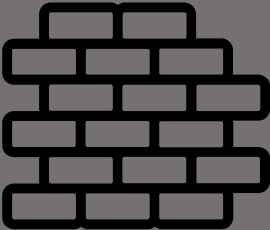

# 未來報酬與即時報酬

---

有時當下看是負面的行為，長遠來看有正面效果

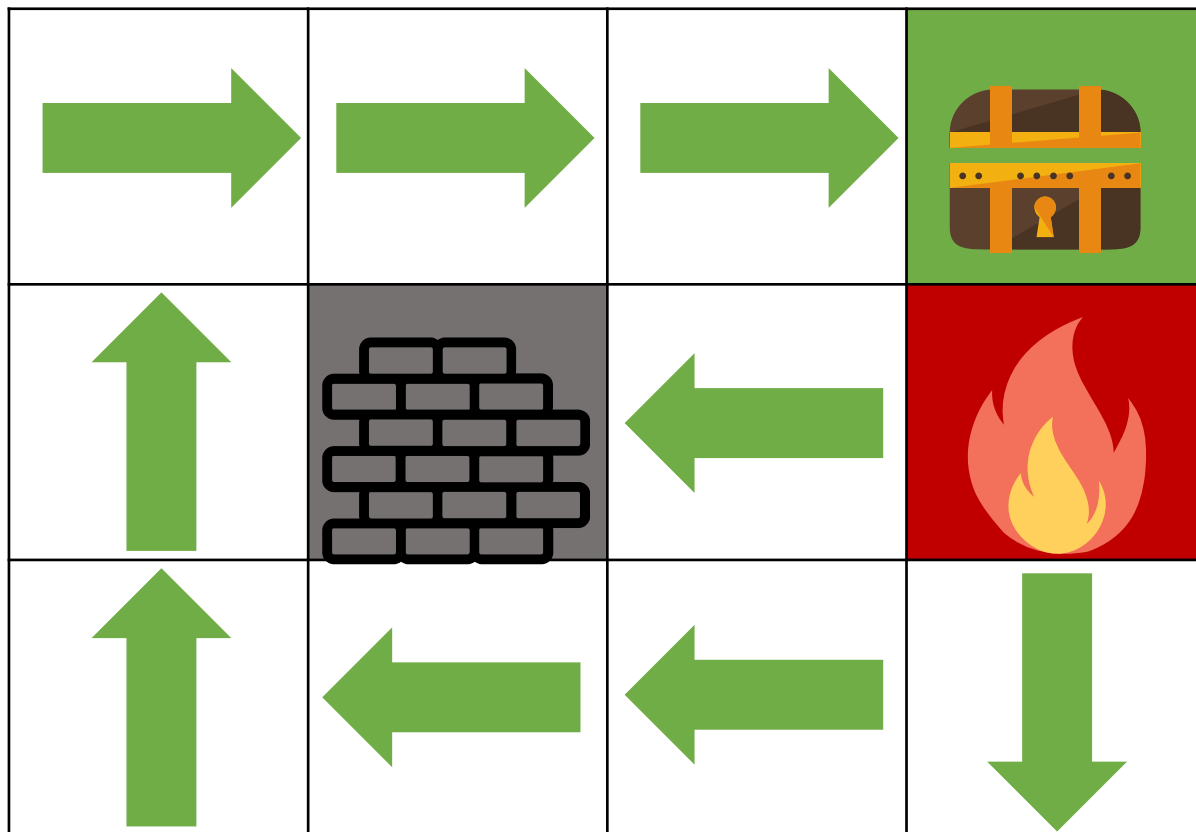


# 考慮即時報酬


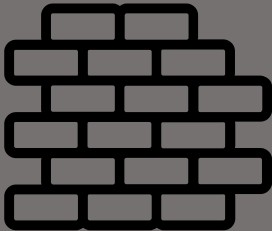

$V=0.7$	$V=0.75$	$V=0.85$	
$V=0.63$		$V=0.36$	
$V=0.55$	$V=0.45$	$V=0.3$	$V=0.2$



# 考慮即時報酬

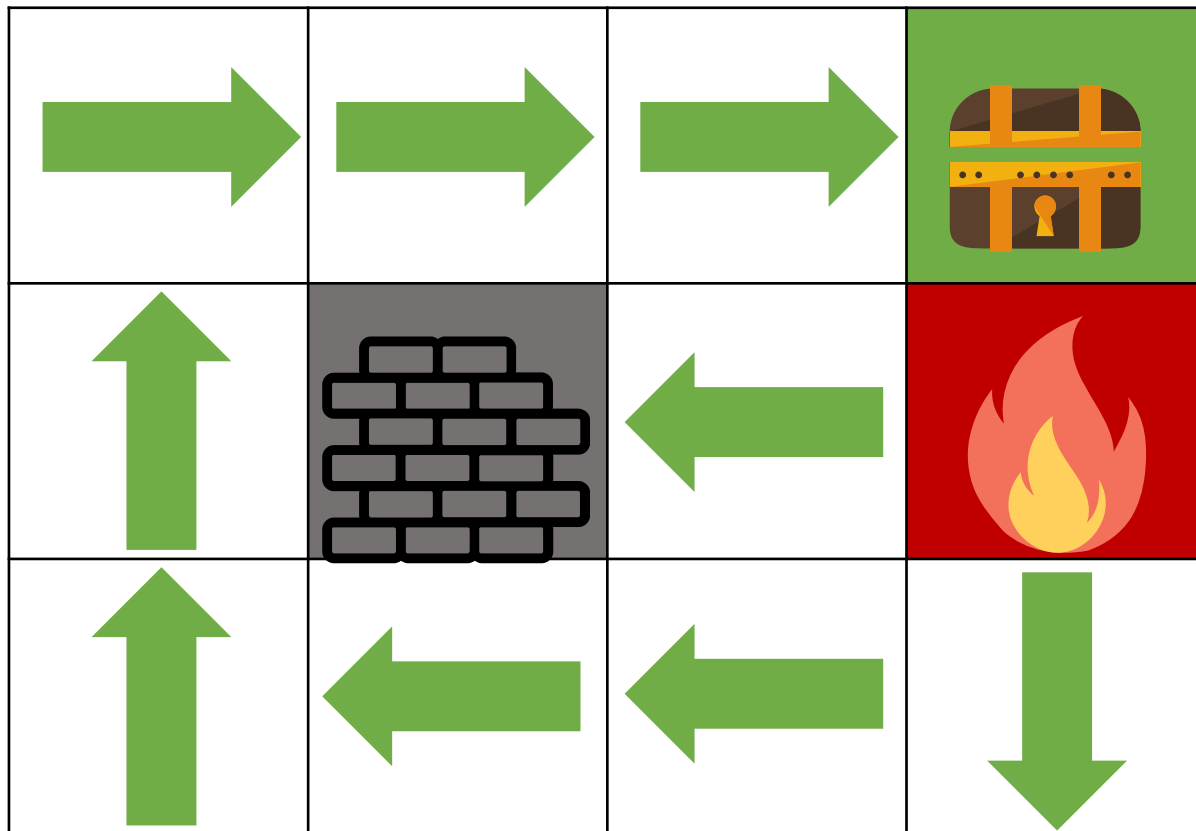


## 增加獎勵折扣

$R=-0.04$	$R=-0.04$	$R=-0.04$	
$R=-0.04$		$R=-0.04$	
$R=-0.04$	$R=-0.04$	$R=-0.04$	$R=-0.04$

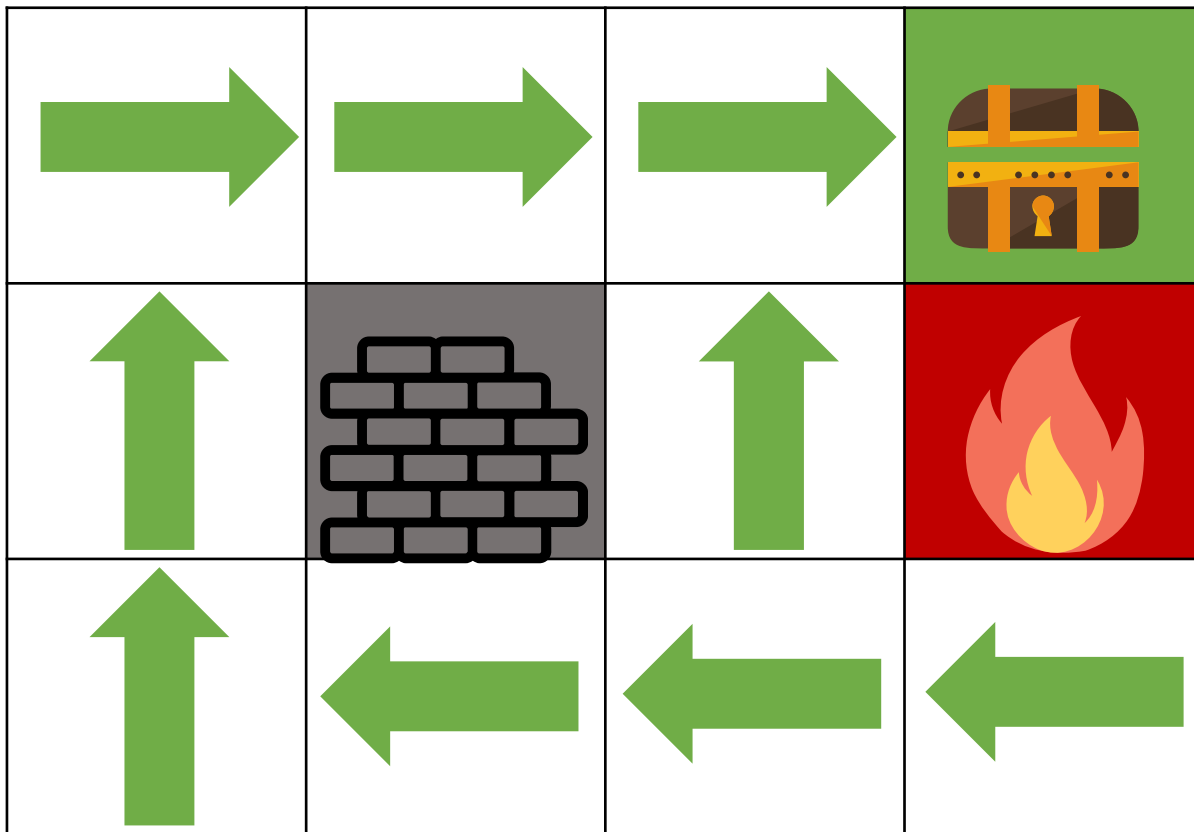
# 折扣未來報酬 $R(S)=0$

假設  $R(S)=0$



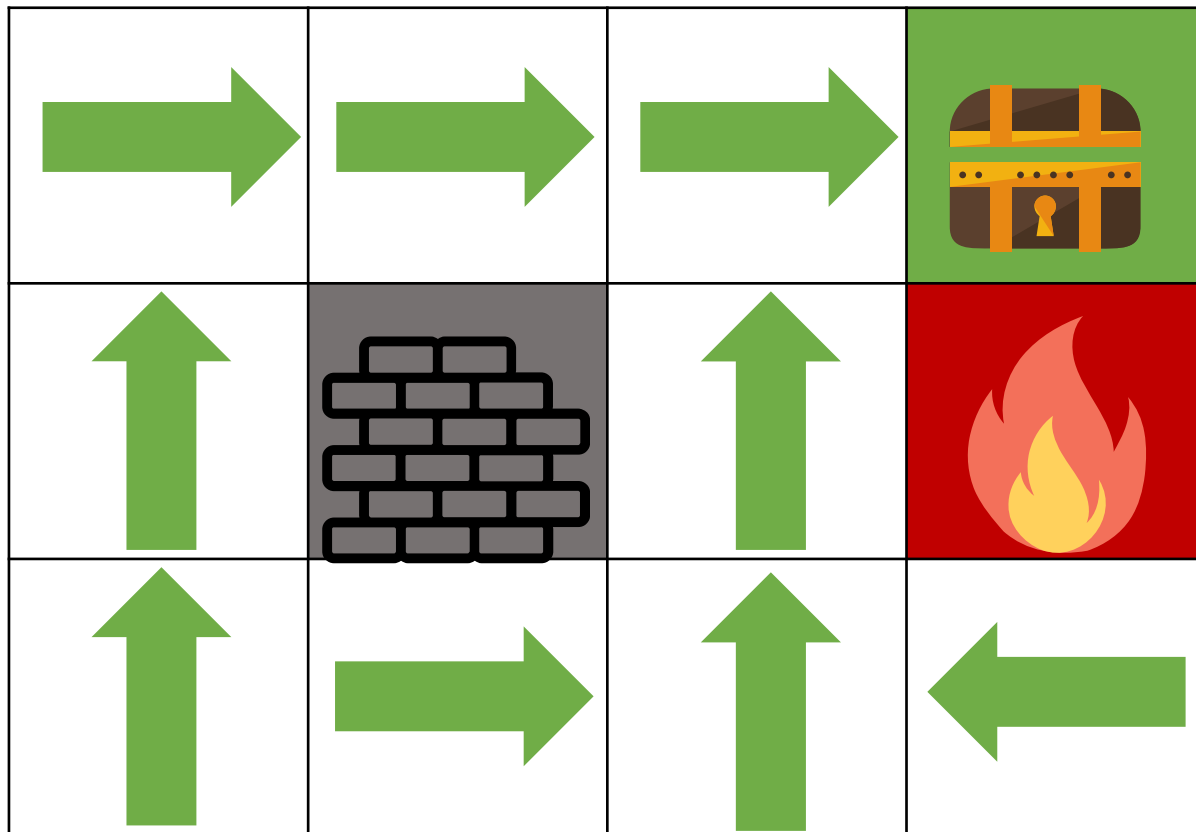
# 折扣未來報酬 $R(S) = -0.04$

假設  $R(S) = -0.04$



# 折扣未來報酬 $R(S) = -0.5$

假設  $R(S) = -0.5$

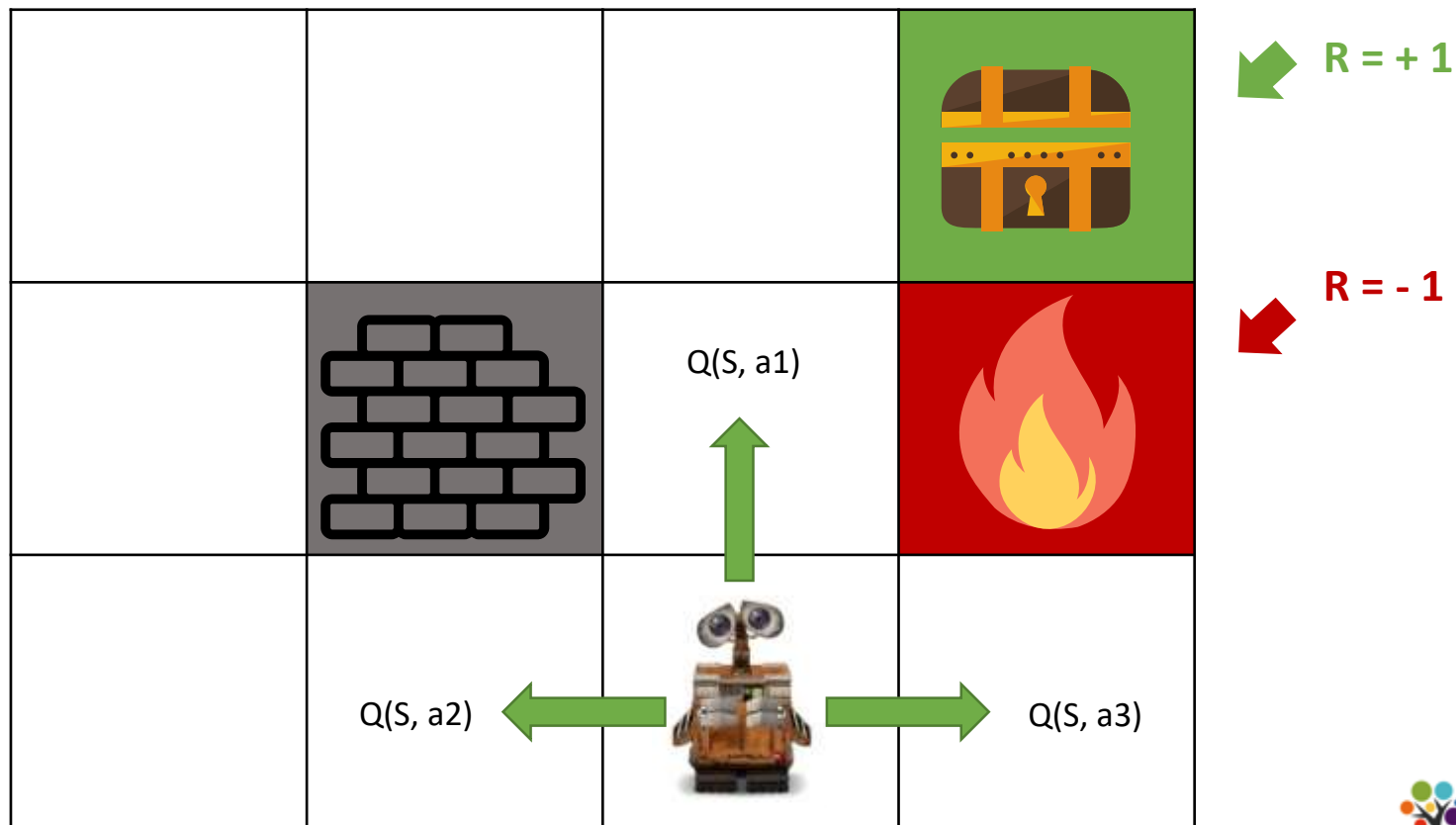




# Q-Learning

---

# 與環境互動過程





# Q 函數

- 計算狀態(State) + 動作(Action) 所對應的品質(Quality)
- 計算在狀態S下做出動作a時，接下來每個動作都是完美的情況(未來報酬最大化)下所對應的結果


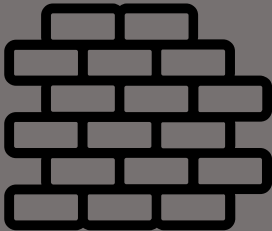

$$Q(s, a) = R(s, a) + \gamma \sum_{s'} P(s, a, s') V(s')$$

$$V(s) = \max_a (R(s, a) + \gamma \sum_{s'} P(s, a, s') V(s'))$$



$$Q(s, a) = R(s, a) + \gamma \sum_{s'} P(s, a, s') \max_{a'} (Q(s', a'))$$

# 隨機策略

$V=0.7$	$V=0.75$	$V=0.85$	
$V=0.63$		$V=0.36$	
$V=0.55$	$V=0.45$	$V=0.3$	$V=0.2$

# 價值反覆運算

- 根據未來的Q更新過去的Q
- 只要知道正確的Q值，該Q就是回合結束之前最後一個動作的Q。在該結果下我們可以確切知道下一個動作的獎勵

$$\widehat{Q}_j \rightarrow \widehat{Q}_{j+1} \rightarrow \widehat{Q}_{j+2} \rightarrow \dots \rightarrow Q^*$$

$$TD(a, s) = R(s, a) + \gamma \max_{a'} Q(s', a') - Q_{t-1}(s, a)$$

$$Q_t(s, a) = Q_{t-1}(s, a) + \alpha TD_t(a, s)$$

$$Q_t(s, a) = Q_{t-1}(s, a) + \alpha (R(s, a) + \gamma \max_{a'} Q(s', a') - Q_{t-1}(s, a))$$

- 透過未來的Q更新現在的Q值，透過價值反覆運算，最後會收斂出較佳的Q

$\alpha$  是學習率

0 代表 永遠不更新Q

1 代表遺忘過去更新到的Q

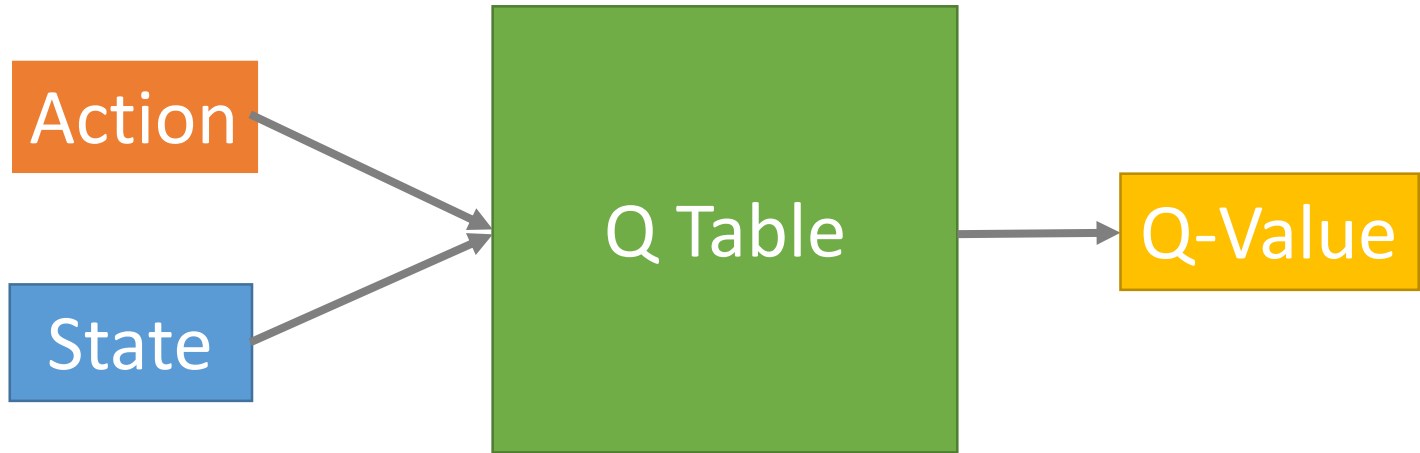


# Deep Q-Learning

---

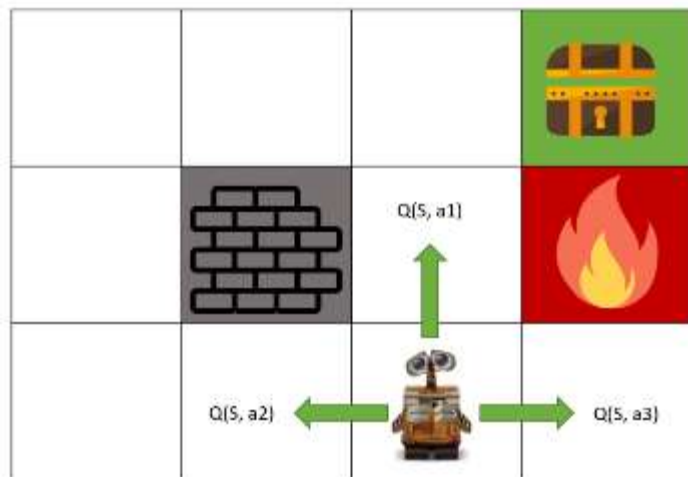
# Q Table

---



# 更複雜的遊戲

狀態: 12種    動作: 4 種



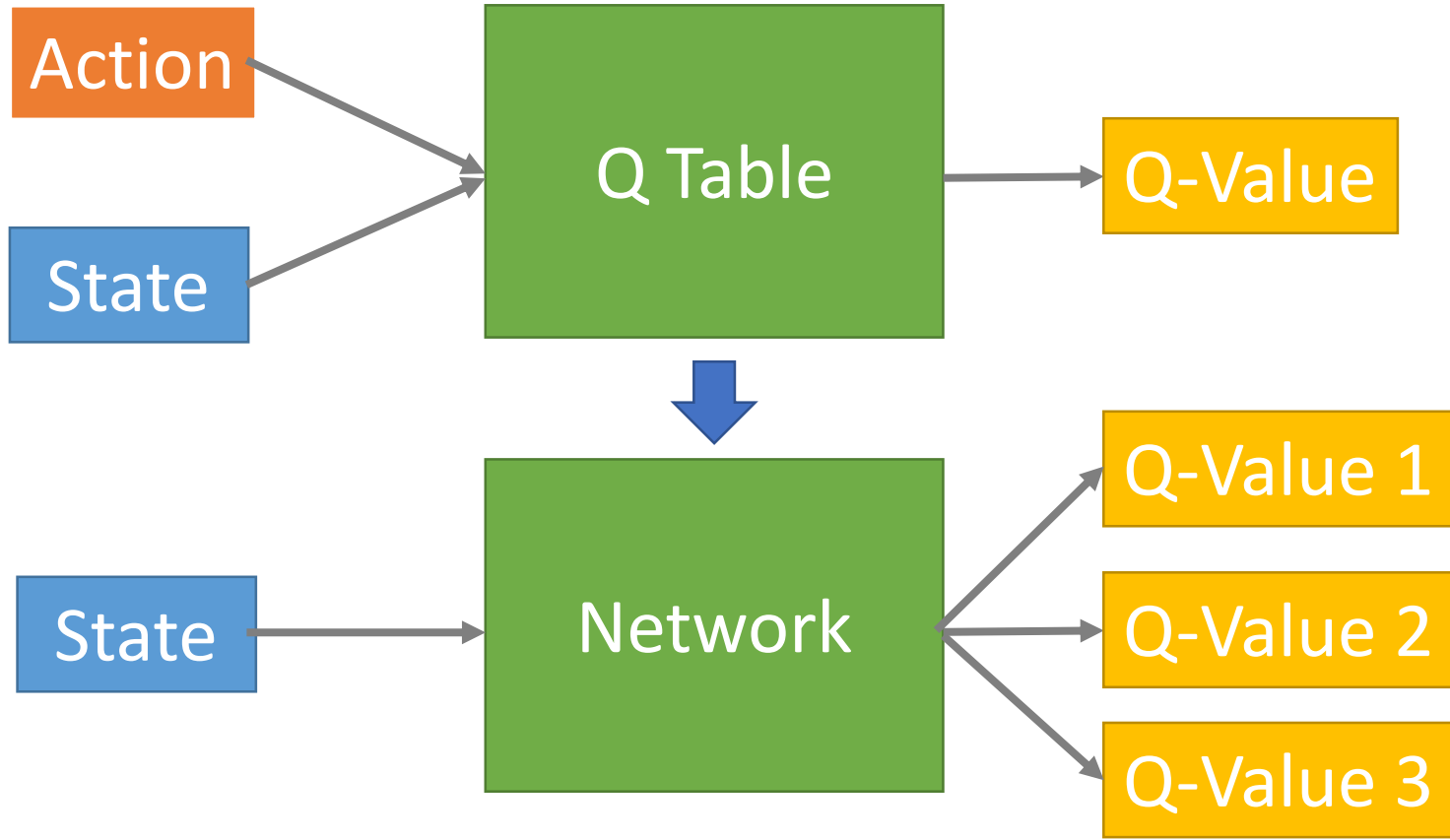
V.S.

狀態: ?種    動作: ? 種



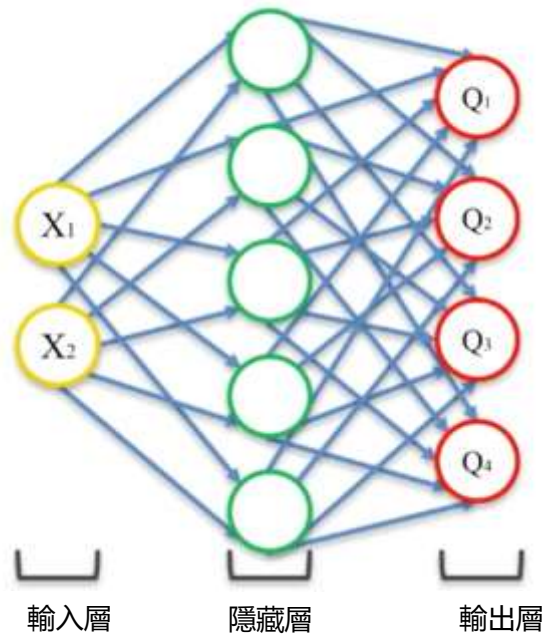
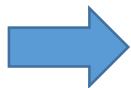
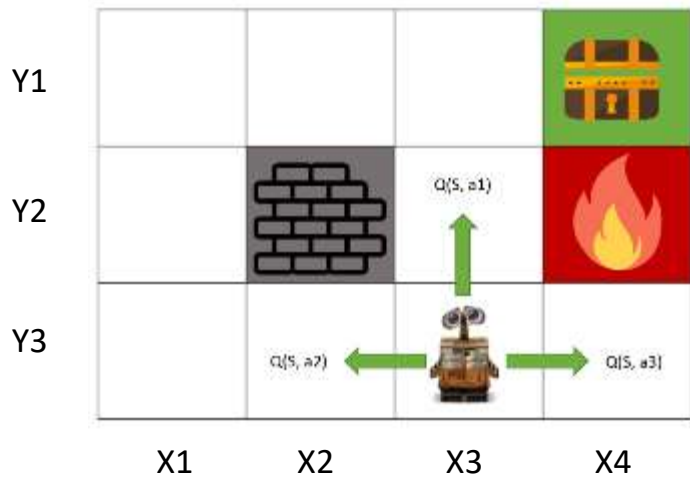
# Deep Q Network

---





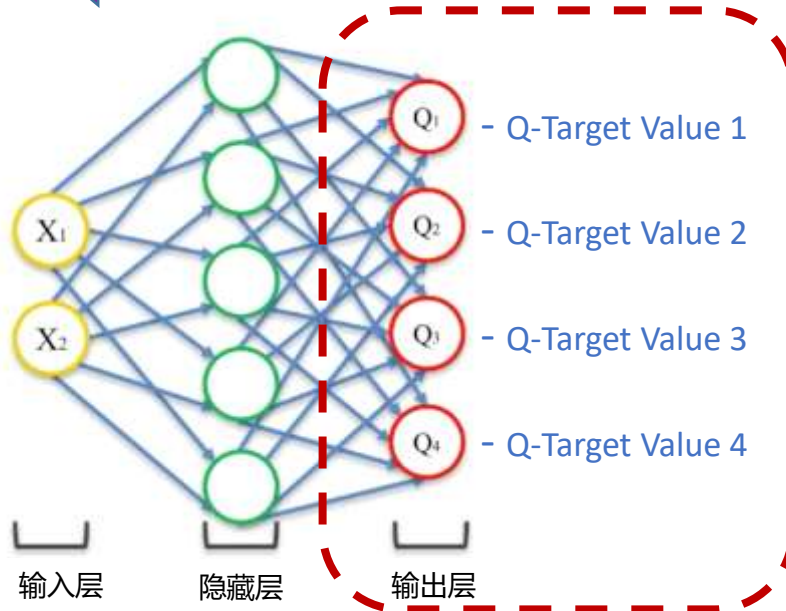
# Deep Q Learning



# Deep Q Learning

Minimize TD

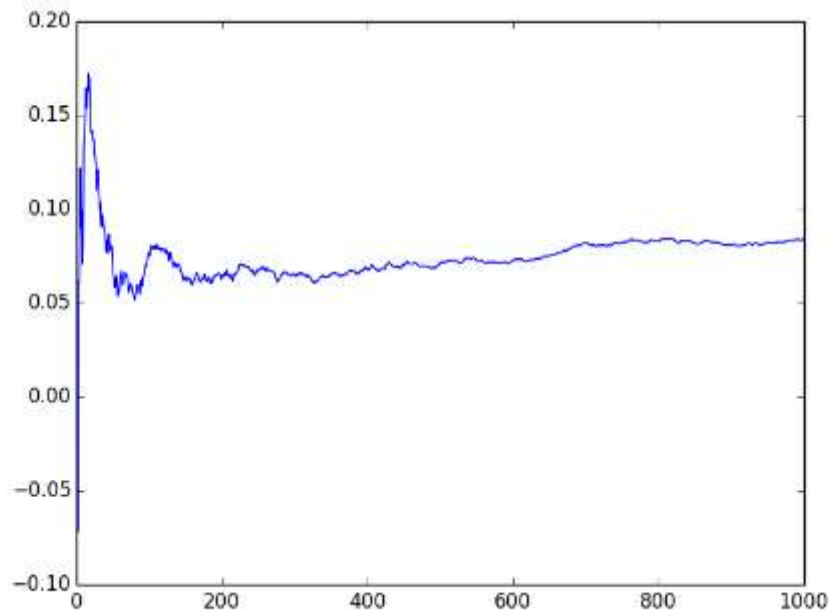
$$TD(a, s) = R(s, a) + \gamma \max_{a'} Q(s', a') - Q_{t-1}(s, a)$$



$$L = \sum Q_{target} - Q$$

# 強化學習與投資機器人

---



- [https://github.com/deependersingla/deep\\_trader](https://github.com/deependersingla/deep_trader)
- <http://nekopuni.holy.jp/2014/08/pythonreinforcement-learning-fx-trading-strategy/>



# THANK YOU

---

EMAIL: [david@largitdata.com](mailto:david@largitdata.com)

網站: [www.largitdata.com](http://www.largitdata.com)

電話: 0929094381