

The background features a complex, abstract design with concentric circles, radial lines, and various geometric shapes like rectangles and triangles. The colors are muted, consisting of light blues, greys, and off-whites, creating a technical or futuristic feel.

人工智慧 學習心得

資工三 110410504 林藝峰

人工智慧的定義與範圍

- 人工智慧是指人類製造出來的機器所表現出來的智慧，其討論研究的範圍很廣。
- 包含演繹、推理和解決問題、知識表示法、規劃與學習、自然語言處理、機器感知、機器社交、創造力等。
- 而我們常常聽到的「機器學習(machine learning)」是屬於人工智慧的一部分，「深度學習(deep learning)」又屬於機器學習的另一部分。

人工智慧

機器學習

深度學習

1950's

1960's

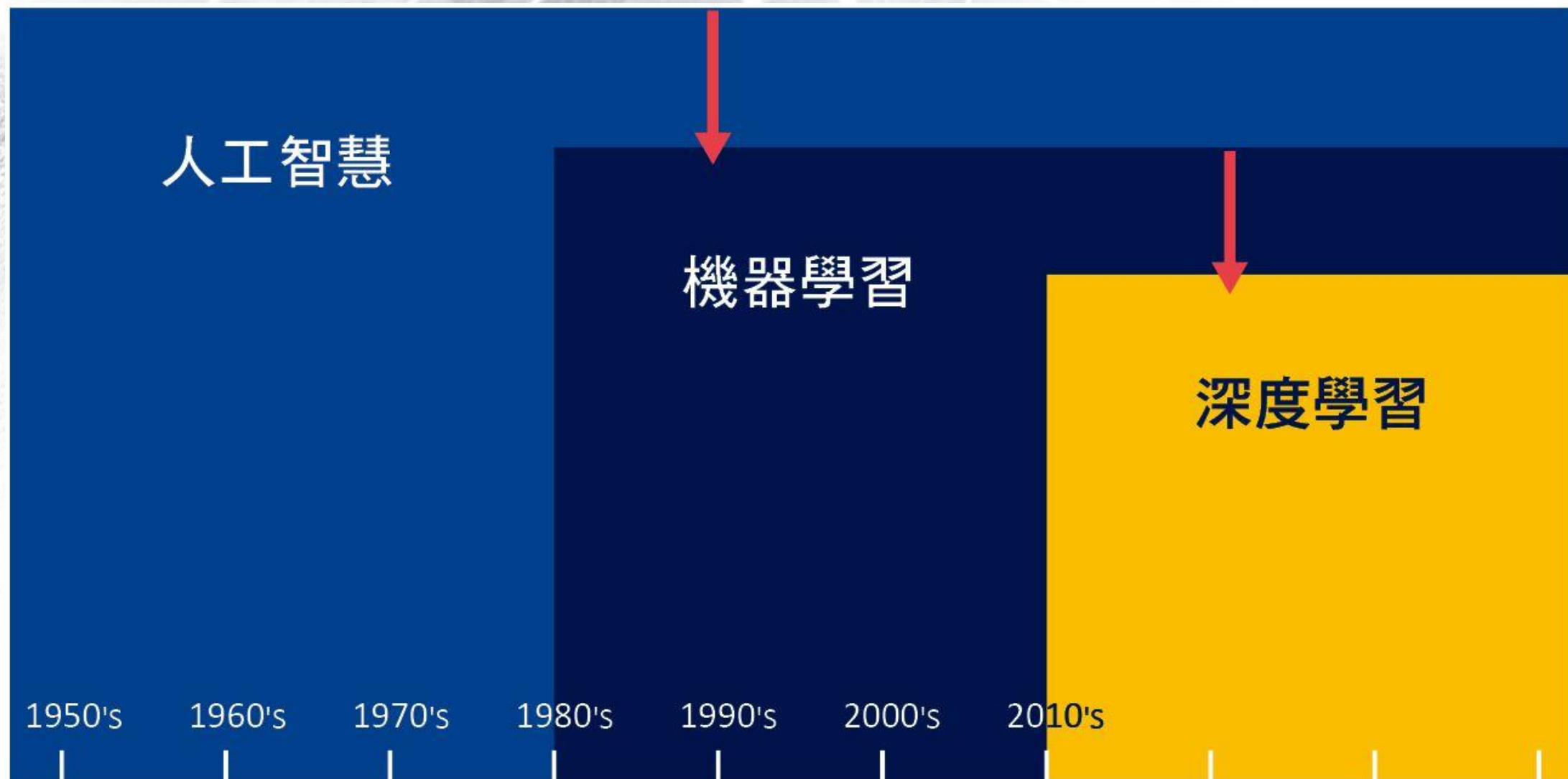
1970's

1980's

1990's

2000's

2010's



人工智慧發展

- 第一級人工智慧 (first level AI) :

電腦擁有自動控制的功能，可以經由感測器偵測外界的溫度、濕度、亮度、震動、距離、影像、聲音等訊號，經由控制程式自動做出相對的反應。

- 第二級人工智慧 (second level AI) :

可以探索推論、運用知識，是基本典型的人工智慧，利用演算法將輸入與輸出資料產生關聯，可以產生極為大量的輸入與輸出資料的排列組合，可能的應用包括拼圖解析程式、醫學診斷程式等。

- 第三級人工智慧(third level AI): 機器學習

指電腦可以根據資料學習如何將輸入與輸出資料產生關聯,「機器學習」是指根據輸入的資料由機器自己學習規則,可能的應用包括搜尋引擎、大數據分析等。

- 第四級人工智慧(fourth level AI): 深度學習

電腦可以自行學習並且理解機器學習時用以表示資料的「特徵值」,因此又稱為「特徵表達學習」,可能的應用包括: Google 教會電腦貓的特徵。

第三級人工智慧處理資料時的「特徵值」必須由人類告訴電腦; 第四級人工智慧處理資料時的「特徵值」可以由電腦自己學習而得。

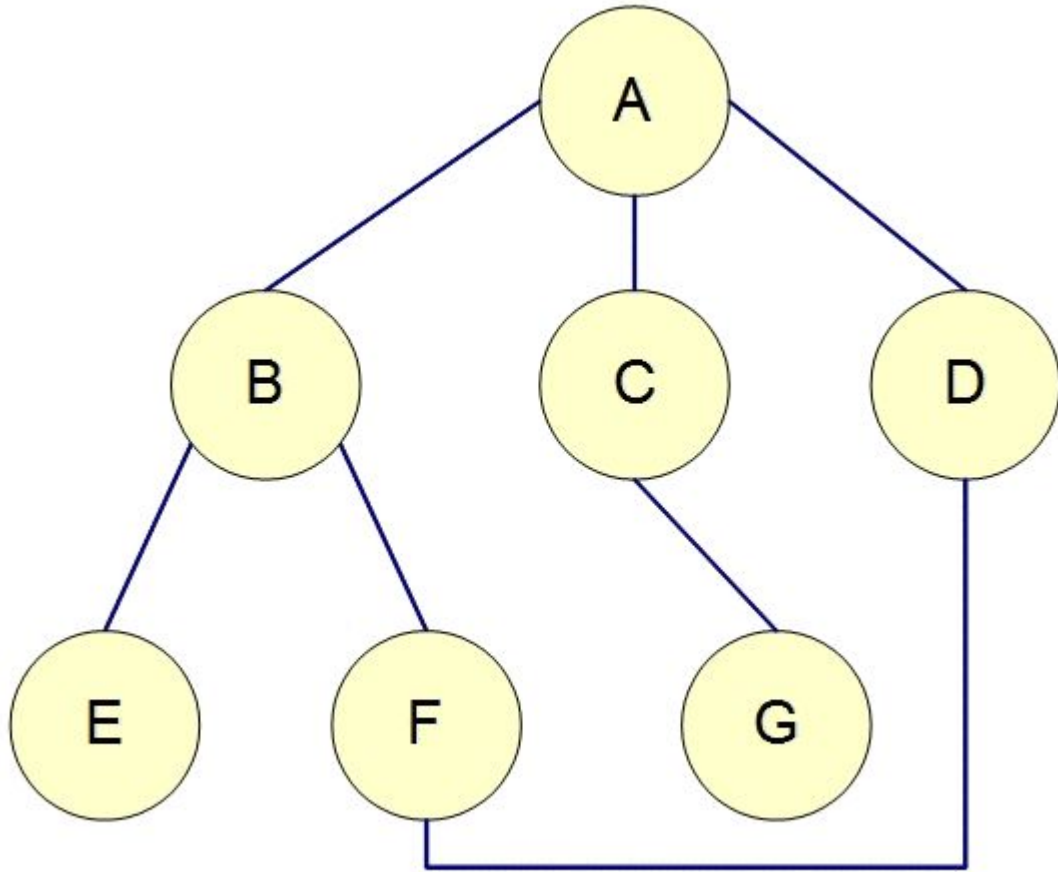
第一級人工智慧就好像是公司裡的工讀生：只是執行老闆交待的命令，進行各種重複性的工作，並不會去思考這個命令是否正確

第二級人工智慧就好像是公司裡的員工，能夠理解老闆交待的規則並且做出判斷，例如老闆說根據箱子長、寬、高分類大小箱子，運用知識留意不同貨物種類

第三級人工智慧就好像是公司裡的經理，能夠學習原則並且自行判斷，例如老闆給予大箱子與小箱子的判斷原則（特徵值），讓經理自己學習如何判斷多大是大箱子？

深度優先學習法

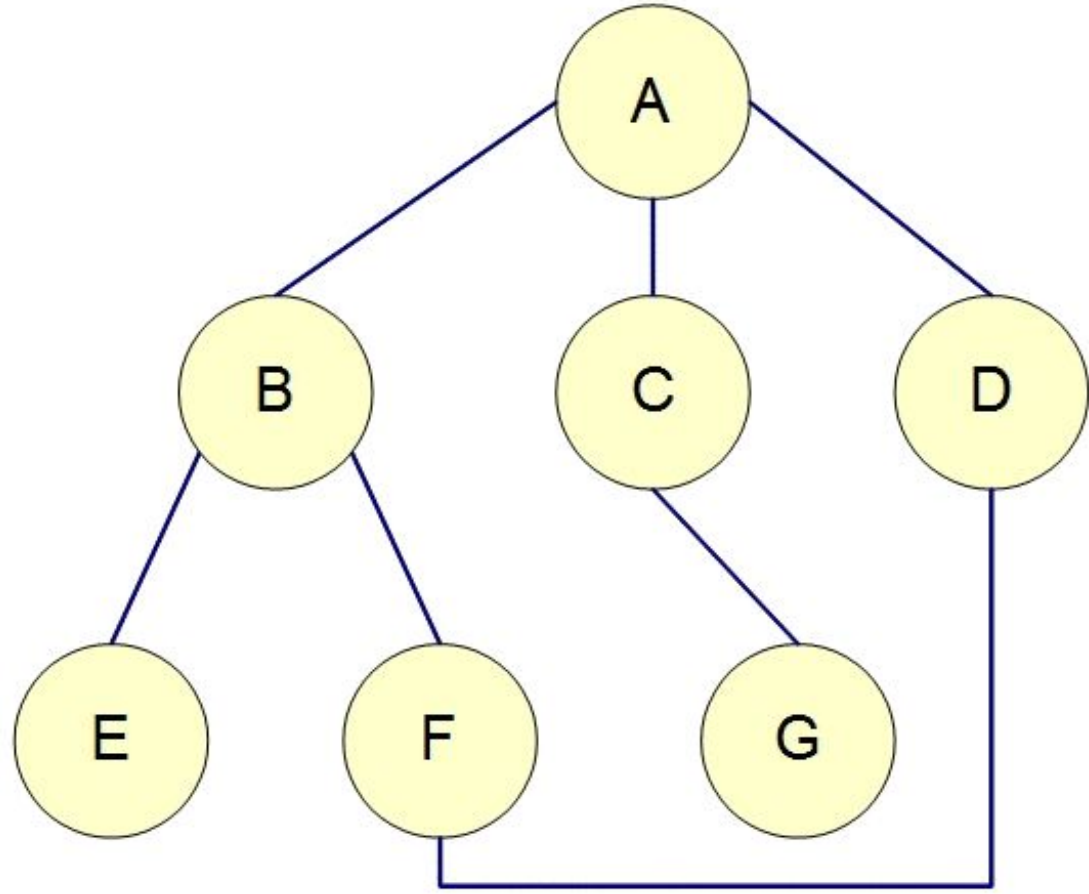
- 一種用來遍尋一個樹或圖的演算法。
- 由樹的根來開始探尋，先探尋邊(edge)上未搜尋的一節點(vertex or node)，並儘可能深的搜索，直到該節點的所有邊上節點都已探尋。
- 該節點全搜尋過後就回溯(backtracking)到前一個節點，重覆探尋未搜尋的節點，直到找到目的節點或遍尋全部節點。



- procedure dfs(vertex v)
 - {
 - mark v as visited
 - for each w adjacent to v {
 - if w unvisited {
 - dfs(w)
 - }
 - }
 - }

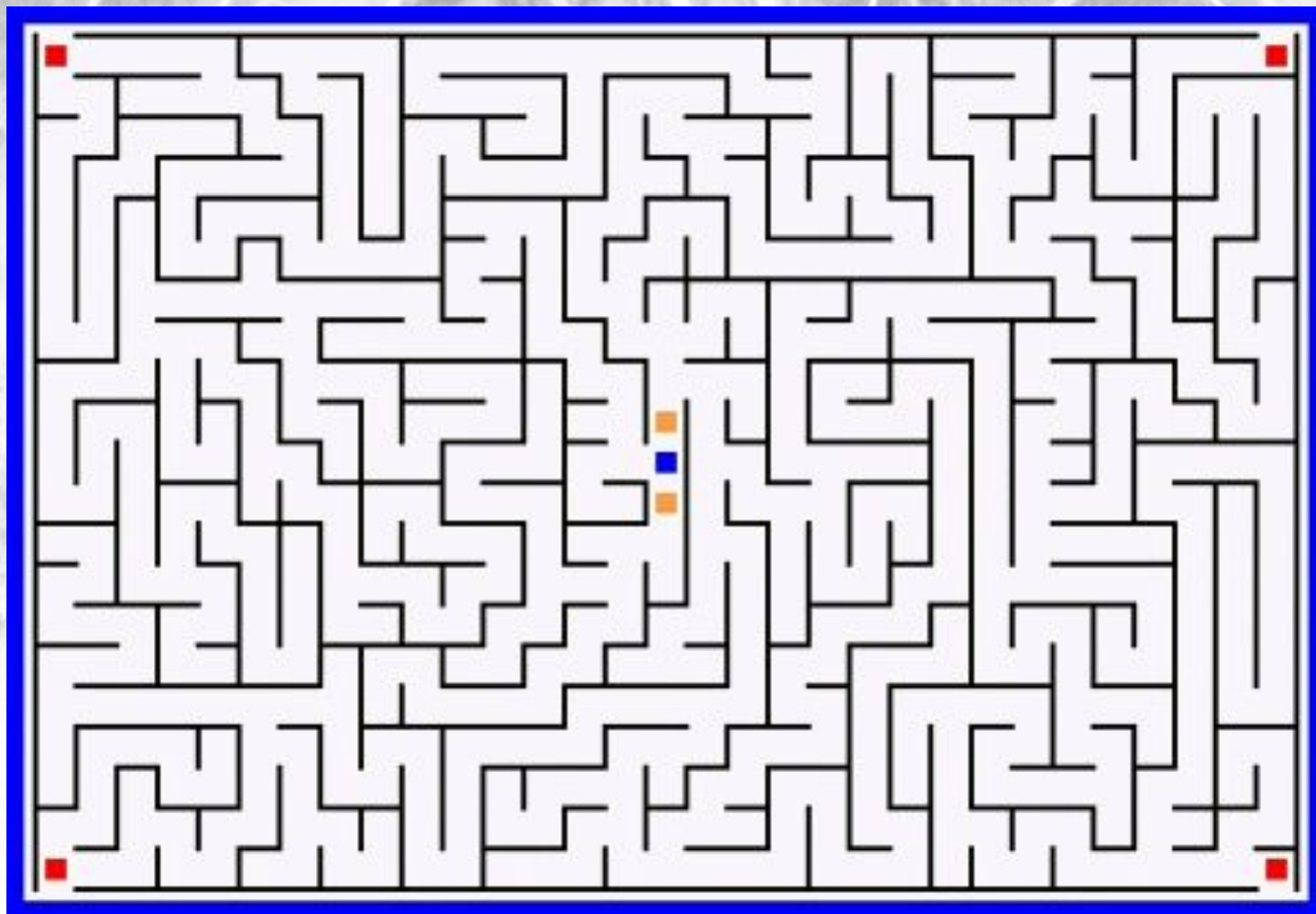
廣度優先學習法

- 是一種圖形搜索演算法。
- 從圖的某一節點(vertex, node)開始走訪，接著走訪此節點所有相鄰且未拜訪過的節點，由走訪過的節點繼續進行先廣後深的搜尋。
- 以樹(tree)來說即把同一深度(level)的節點走訪完，再繼續向下一個深度搜尋，直到找到目的節點或遍尋全部節點。
- 廣度優先搜尋法屬於盲目搜索(uninformed search)，是利用佇列(Queue)來處理，通常以迴圈的方式呈現。



- procedure BFS(vertex s)
 - {
 - create a queue Q
 - enqueue s onto Q
 - mark s as visited
 - while Q is not empty {
 - dequeue a vertex from Q into v
 - for each w adjacent to v {
 - if w unvisited {
 - mark w as visited
 - enqueue w onto Q
 - }
 - }
 - }
 - }

假設起始點在迷宮的中央，而出口在迷宮的四個角落，由於廣度優先搜尋法是將每個方格的下一步全部走完，所以當最先走到出口的路徑即為最短路徑。



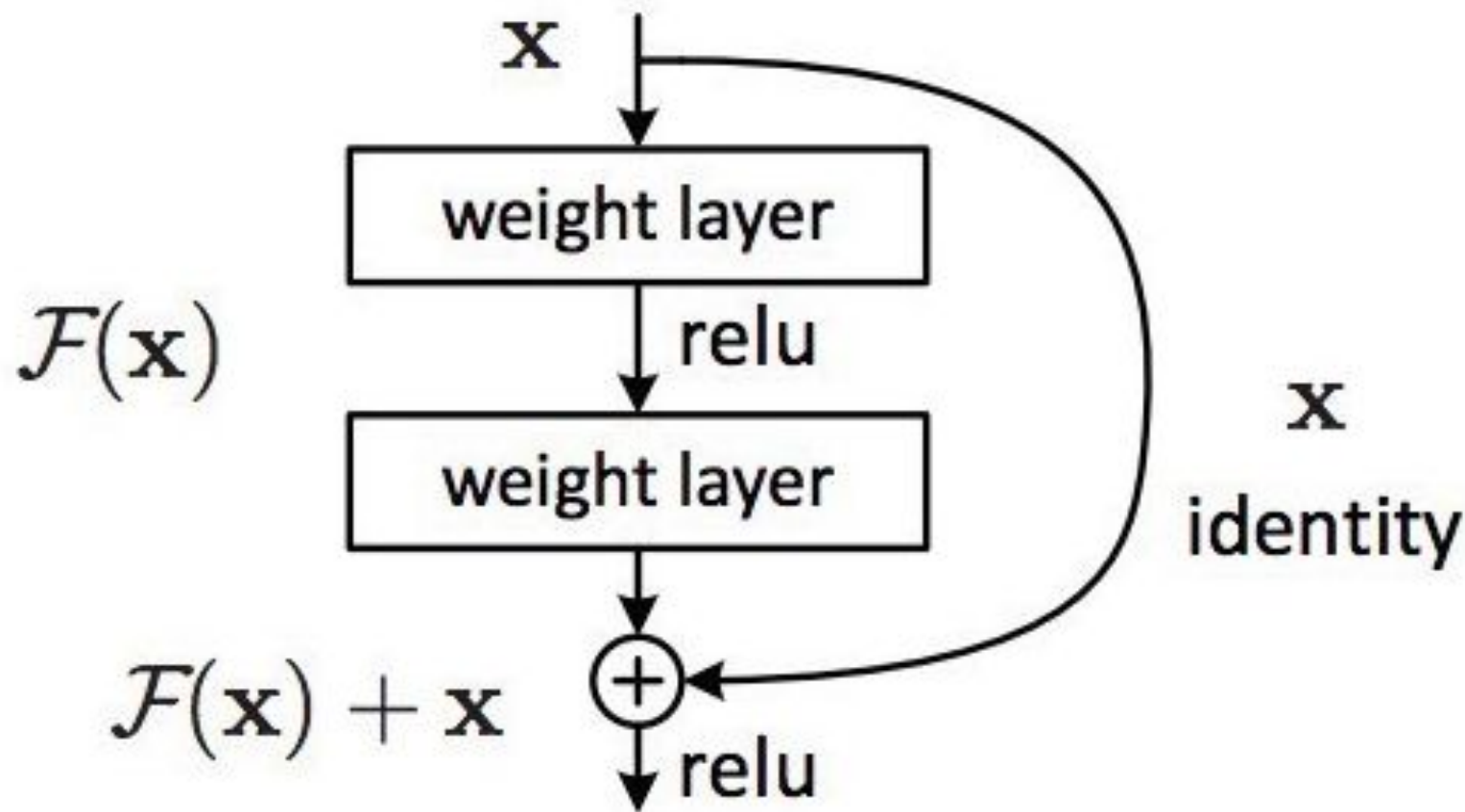
AlphaGo Zero

- AlphaGo Zero用CNN來改進圍棋投子策略。具體到CNN的系統架構，AlphaGo Zero用的是Residual架構ResNet。
- 可以直接使用人類棋手的棋譜來訓練ResNet。理論上來說，用人類棋手的棋譜來訓練ResNet，AlphaGo Zero的水平，會在更短時間內，獲得更快提升。
- 即便不用人類棋手的棋譜，只用機器自我對弈的棋譜，來訓練ResNet，在短短 40 天內，AlphaGo Zero 就已經超越人類棋手的水平。

ResNet

- 假設我們建構了一個 X 層網路，並實現了一定的準確度。那麼一個 $X+1$ 層網路至少也能實現同樣的準確度——只要簡單複製前面 X 層，在最後一層加上一恆等映射即可。同樣的， $X+2$ 、 $X+3$ 的網路都可以繼續增加恆等映射，然後實現同樣的準確度。但在實際情況下，這些深度網路都會表現很差。
- ResNet的作者將這些問題歸咎於：直接映射的學習效率很差。而他提出了解決方法：不再學習 x 到 $H(x)$ 的基本映射關係，而是學習兩者間的差異，也就是「殘差(residual)」。最後，為了計算 $H(x)$ ，我們只須將此殘差加到輸入上即可。

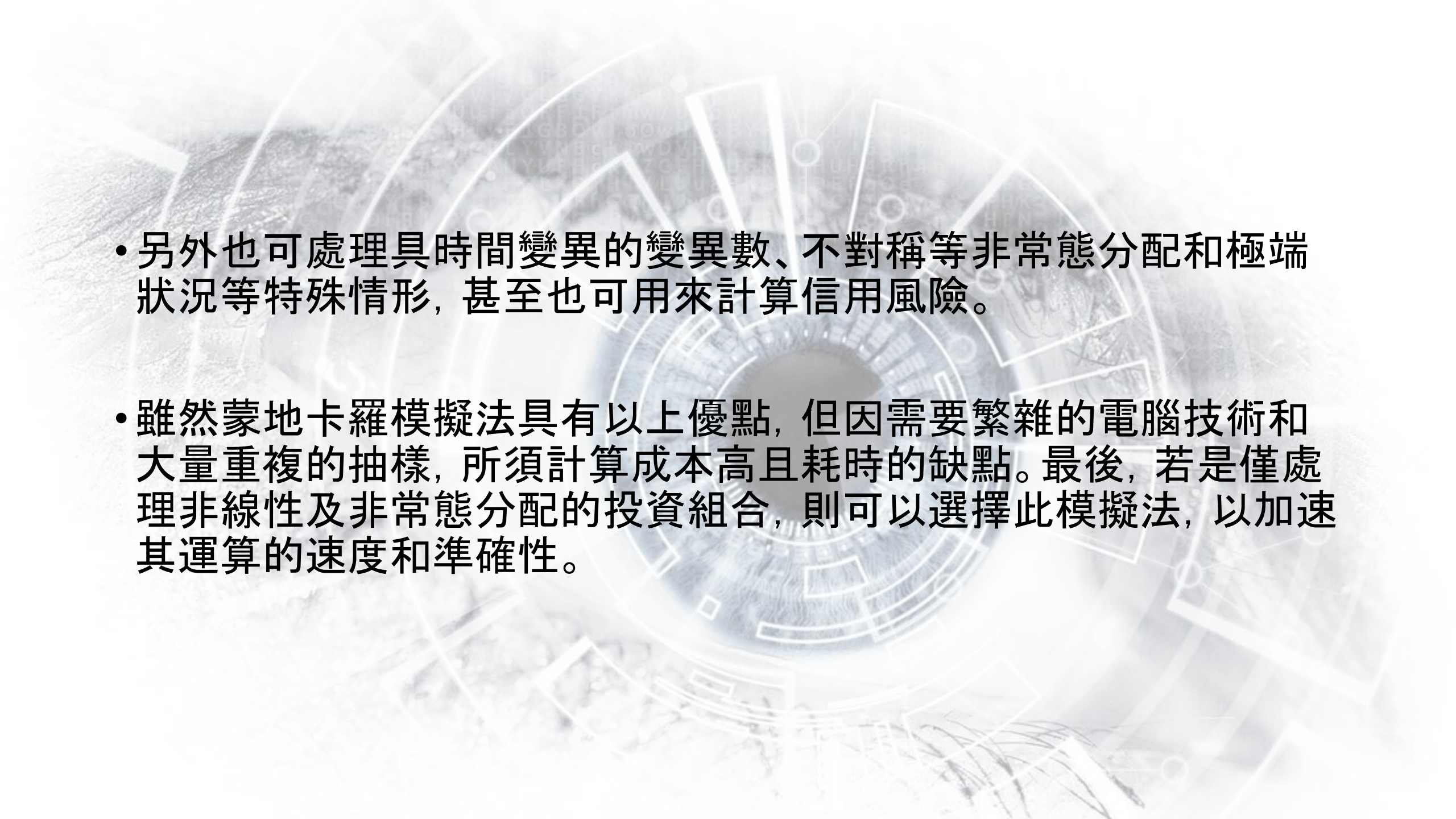
- 假設殘差為 $F(x)=H(x)-x$ ，而現在我們的網路不會學習 $H(x)$ 了，而直接學習 $F(x)+x$



- ResNet的每一個「模塊(block)」都由一系列「捷徑(shortcut)」所組成, 此「捷徑」將模塊的輸入和輸出連接在一起。
- 在層面上執行「加法(add)」運算, 如果輸入輸出大小不同, 那就可使用「零填充」或「投射」(通過 1×1 卷積)來得到匹配的 X 大小。
- 回到上述實驗, 這能大大簡化衡等映射的建構。比起從頭學習衡等變換, 使 $F(x)$ 為0比使輸出為 X 要容易得多。一般來說, ResNet 會給層一個「參考點」 X , 以 X 為基礎學習。

蒙地卡羅演算法

- 是一種數值方法，利用亂數取樣模擬來解決數學問題。
- 舉凡在所有目前具有隨機效應的過程，均可能以蒙地卡羅方法大量模擬單一事件，藉統計上平均值獲得某設定條件下實際最可能測量值。
- 此演算法是基於大數法則的實證方法，當實驗的次數越多，其平均值也就會越趨近於理論值。其法則亦可以估算投資組合的各種風險因子，特別是一些難以估算的非線性投資組合。

- 
- The background of the slide features a complex, abstract design. It consists of several concentric circles and arcs, some of which are highlighted in a light blue or white color. Interspersed among these geometric shapes are various binary digits (0s and 1s) and small, stylized icons, giving the overall appearance a high-tech or digital feel. The text is overlaid on this background in a clear, black, sans-serif font.
- 另外也可處理具時間變異的變異數、不對稱等非常態分配和極端狀況等特殊情形，甚至也可用來計算信用風險。
 - 雖然蒙地卡羅模擬法具有以上優點，但因需要繁雜的電腦技術和大量重複的抽樣，所須計算成本高且耗時的缺點。最後，若是僅處理非線性及非常態分配的投資組合，則可以選擇此模擬法，以加速其運算的速度和準確性。

EM演算法

- 知道一些條件(觀察到的數據), 還有不了解的部分, 需要全部解析出來。
- 先根據假設做猜測(E-Step), 將不知道的部分全部做假設(Parameter)。有了虛構數據後, 持續思考並更新自己的猜測, 使自己觀察到的數據更加正確(Maximize likelihood; M-step); 反覆循環猜測、思考。最後得到可以解釋整個數據的假設。

馬可夫鍊

- 馬可夫鍊，就是同類型的事件（不同的狀態）依序發生的機率。

- 舉例來說，假設天氣有三種狀態：

晴天、陰天跟雨天。如果昨天是雨天，那麼今天是「雨天」的機率，會跟昨天是「晴天」而今天是「雨天」的機率有所不同，這是因為我們相信天氣現象在時間上有某種連續性，前面發生的狀態會影響到後面發生的狀態，而馬可夫模型就是描述這種前後關係的數學語言。

隱馬可夫鍊

- 沿用上則天氣的例子，假設觀測者因腳受傷，必須住在房間裡，看不到外面的天氣，但是我可以觀察到隔壁房間的鄰居每天從事的運動：「跑步」、「健身操」或是「游泳」。
- 如果把鄰居每天從事的運動項目記錄下來，就是「運動」這個事件的馬可夫鍊，這是我可以觀察的到的現象。然後，我又依照過去的經驗，知悉在每種天氣狀況下，他從事各項運動的機率，那麼是否還是可以透過的觀察，去推測每天的天氣？



- 在此範例中，有兩個事件的序列：

自身觀察到的鄰居每天所從事的運動項目；另一個為看不見的，外面每天的天氣。由於我知悉這兩個馬可夫鏈之間的關係，所以便可以由其中一個馬可夫鏈的狀態，去預測另一個馬可夫鏈的狀態。

- 而「隱馬可夫模型」，便是描述這樣的兩個序列的關係的統計模型。