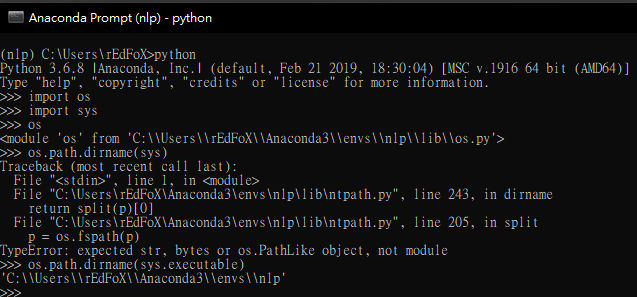
**第一週**



路徑查詢

1. 註解為#及’’’

2. range(a, b)：不含b數字，range(a)為0~a

3. continue：會跳回一開始的迴圈

4. \_\_XXXXX \_\_：系統內建、系統預設

5. import：可以引用同資料夾的py

6. len(陣列)：回傳陣列長度

7. array.sort()：排列陣列

8. del array[X]：從陣列刪除第X項

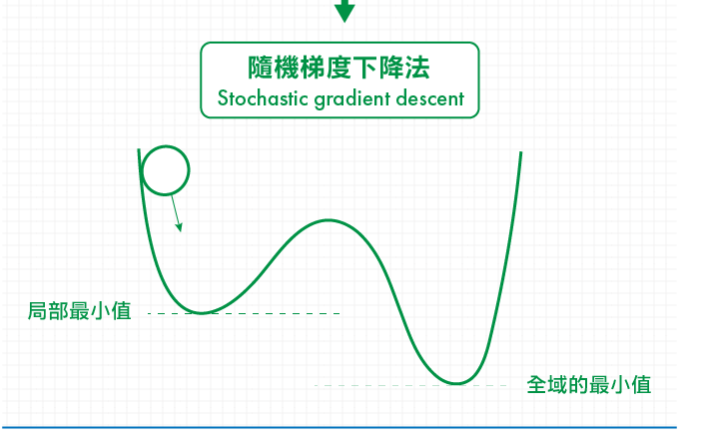
9. 陣列[a:b]為陣列a到b項

**第二週**

1. 爬山演算法：比較周圍尋找最低或最高點

2. 缺點：慢，可能會略過較遠的更高點

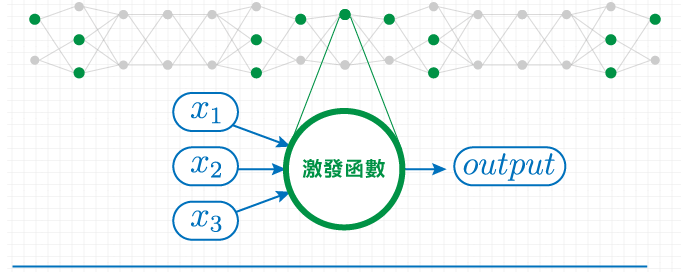
3. 隨機爬山演算法：隨機挑一個變數比較週圍，直到找到最高點後再次跳到隨機位置重複比較，較能避免相鄰兩側的山谷只能找到其一的情形。



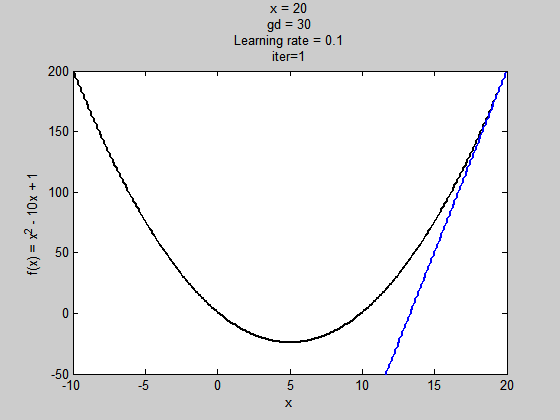
取自：<https://www.stockfeel.com.tw/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92%E7%9A%84%E8%A1%B0%E9%A0%B9%E8%88%88%E7%9B%9B%EF%BC%9A%E5%BE%9E%E9%A1%9E%E7%A5%9E%E7%B6%93%E7%B6%B2%E8%B7%AF%E5%88%B0%E6%B7%BA%E5%B1%A4%E5%AD%B8%E7%BF%92/>

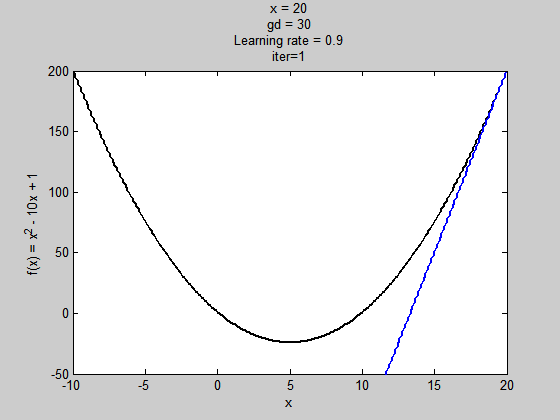
**第三週**

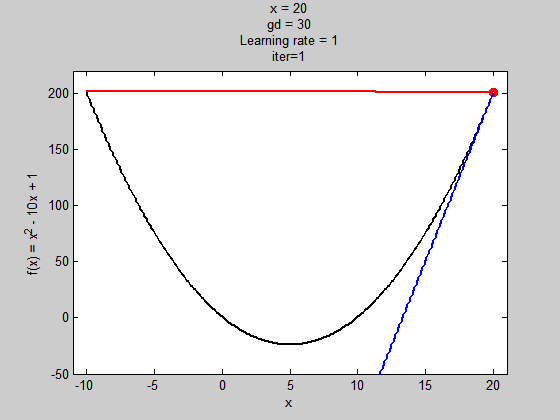
1. 類神經網路：類神經網路是利用電腦去模擬人的神經系統做出來的神經元群組。



取自：<https://www.stockfeel.com.tw/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92%E7%9A%84%E8%A1%B0%E9%A0%B9%E8%88%88%E7%9B%9B%EF%BC%9A%E5%BE%9E%E9%A1%9E%E7%A5%9E%E7%B6%93%E7%B6%B2%E8%B7%AF%E5%88%B0%E6%B7%BA%E5%B1%A4%E5%AD%B8%E7%BF%92/>

梯度下降法：梯度指的是斜率最大(oo)的方向，也就是說，梯度下降法便是朝著斜率最大的方向前進。而正梯度便是朝上走，逆梯度則是朝下走，直到走至梯度為0的點，代表其切線水平，也就是走到谷底為止。取自：<https://medium.com/@chih.sheng.huang821/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E5%9F%BA%E7%A4%8E%E6%95%B8%E5%AD%B8-%E4%BA%8C-%E6%A2%AF%E5%BA%A6%E4%B8%8B%E9%99%8D%E6%B3%95-gradient-descent-406e1fd001f>

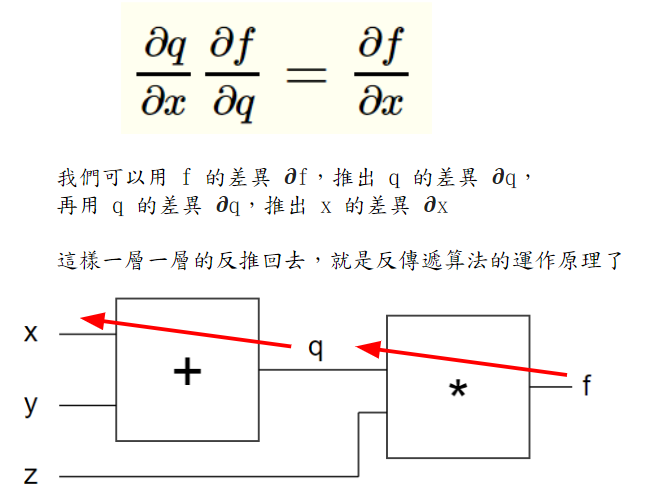
取自：<https://medium.com/@chih.sheng.huang821/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E5%9F%BA%E7%A4%8E%E6%95%B8%E5%AD%B8-%E4%BA%8C-%E6%A2%AF%E5%BA%A6%E4%B8%8B%E9%99%8D%E6%B3%95-gradient-descent-406e1fd001f>



取自：<https://medium.com/@chih.sheng.huang821/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E5%9F%BA%E7%A4%8E%E6%95%B8%E5%AD%B8-%E4%BA%8C-%E6%A2%AF%E5%BA%A6%E4%B8%8B%E9%99%8D%E6%B3%95-gradient-descent-406e1fd001f>

**第四週**

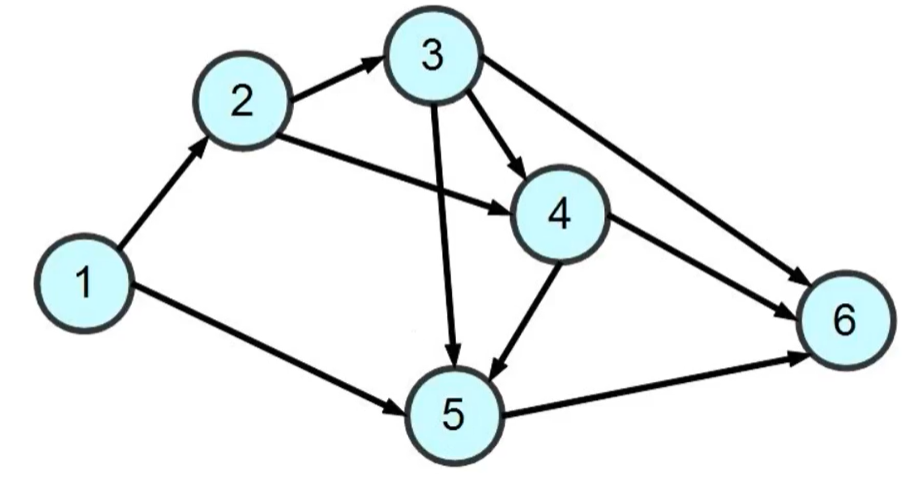
1. 類神經 原理：當我們對神經元進行輸入（Xi）後，經過激發函數與內部迴歸模型對輸入的權重（Wi）加乘，再加入偏誤（b）後，便完成了該節點的輸出。(取自：<https://medium.com/marketingdatascience/%E5%BF%AB%E9%80%9F%E5%8F%8D%E6%87%89%E6%A9%9F%E5%88%B6-%E9%A1%9E%E7%A5%9E%E7%B6%93%E7%B6%B2%E8%B7%AF-a3bbdee4a6f6>)
2. 自動微分-反傳遞演算法



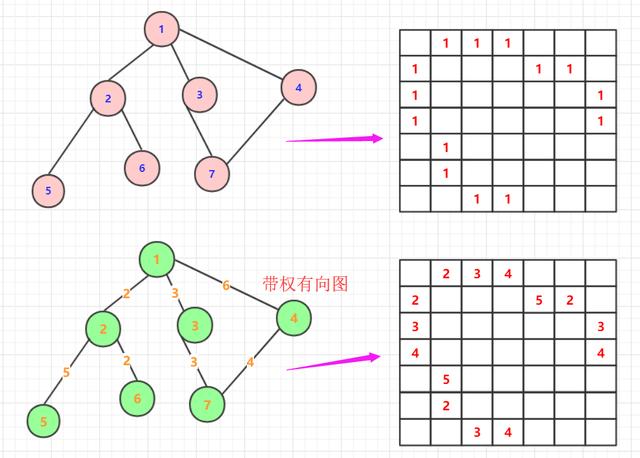
取自：<https://medium.com/@ccckmit/%E4%B8%80%E5%88%86%E9%90%98%E5%AD%B8%EF%BD%81%EF%BD%89-3-%E6%A2%AF%E5%BA%A6%E4%B8%8B%E9%99%8D%E8%88%87%E5%8F%8D%E5%82%B3%E9%81%9E%E7%AE%97%E6%B3%95-90f1ed03ffdd>

**第六週**

1. 圖形搜尋：節點與連線所形成的圖形(取自：<http://programmermagazine.github.io/201406/htm/focus1.html>)

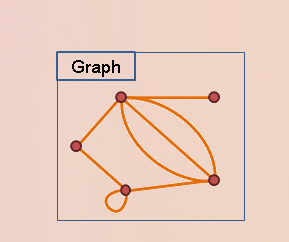


取自：<http://programmermagazine.github.io/201406/htm/focus1.html>



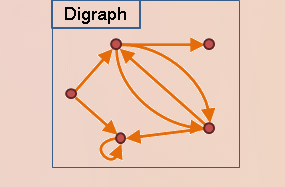
以矩陣形式表示，取自：<https://kknews.cc/code/3453n3y.html>

1. 無向圖



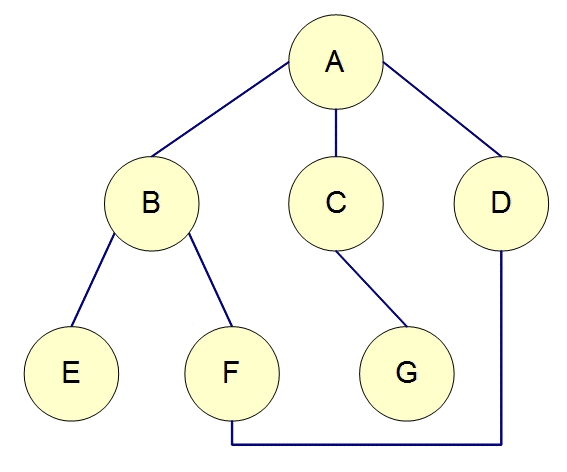
取自：<http://www.csie.ntnu.edu.tw/~u91029/Graph.html>

1. 有向圖



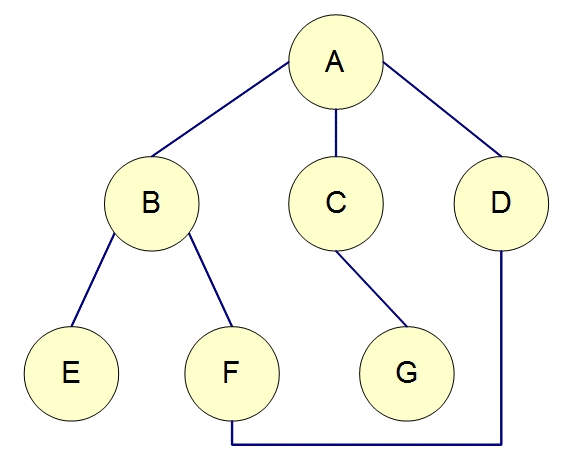
取自：<http://www.csie.ntnu.edu.tw/~u91029/Graph.html>

1. 深度優先搜尋：會先往代號小的搜尋，如果已經搜尋過就往順位次等的方向前進(以遞迴方式)。



取自：<http://simonsays-tw.com/web/DFS-BFS/DepthFirstSearch.html>

1. 廣度優先搜尋： A為起始點，且每一節點由左至右的順序來搜尋下個節點，則結果為: A🡪 B🡪 C🡪 D🡪 E🡪 F🡪 G



取自：<http://simonsays-tw.com/web/DFS-BFS/BreadthFirstSearch.html>

**第七週**

1. 對局搜尋：三個關鍵技巧

第一個是盤面的表達：用程式的方式表達盤面

第二個是評估函數：對每個棋子的重要性計分

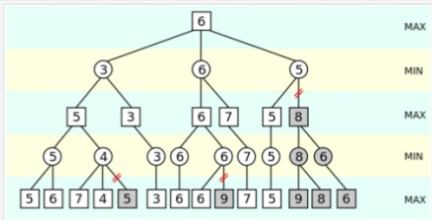
第三個是搜尋技巧：是先判斷未來局勢，非常重視計算速度

1. 電腦下棋主要的算法分別為

盤面評估函數：盤面評估函數首先需要定義一個分數的目標，還要評 估雙方的分數判斷要進攻還是防守。

搜尋多層對局：搜尋多層對局可以藉由運算預判最可能贏得勝利的下法，但是由於可能性有很多的關係，如何減少運算量以算出更多層是非常重要的事。

Alpha-Beta修剪法：減去不可能會下的分支以減少運算量。



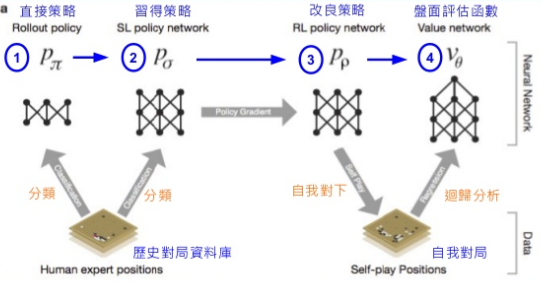
取自：<http://programmermagazine.github.io/201407/htm/focus3.html>

1. Alpha Go的設計原理

兩個主要的神經網路函數：

1. 策略網路：計算某一子的機率(機率隨贏面越大而增加)
2. 價值網路：也就是盤面評估函數

基本上若是兩者取其一也可以。



兩個主要的訓練方式：

1. 歷史對局資料庫：餵棋譜等資料

2.自我對局：自己跟自己對奕(強化式學習)

主要採用深捲積神經網路進行訓練(擷取特徵，例如：斜線)。以蒙地卡羅對局樹搜尋法找出有價值的盤面。

\*深捲積神經網路：常用於影像處理、辨識

\*蒙地卡羅對局樹搜尋法

**第八週**

1. 邏輯推論
2. 計算邏輯

無法寫出一個程式來判斷另一個程式會不會停止(停止問題)

一個問題是否具有可計算性、可證明性

一致性：指公理系統本身不會具有矛盾的現象。

完備性：指一個公理系統，可以《推論出所有的在其中為真的定理》，不會有任何的遺漏，也就是《所有的定理都可以得到證明》的意思。

1. 哥德爾完備定理
2. 哥德爾不完備定理

任何相容的形式系統，只要蘊涵皮亞諾算術公理，就可以在其中構造在體系中不能被證明的真命題，因此通過推演不能得到所有真命題（即體系是不完備的）。

任何邏輯自洽的形式系統，只要蘊涵皮亞諾算術公理，它就不能用於證明它本身的相容性。

一階邏輯系統是一致且完備的，也就是所有的一階邏輯定理都可以透過機械性的推論程序證明出來，而且不會導出矛盾的結論。

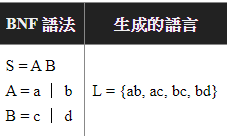
**第九週**

1. 語言處理

自然語言：歷史所衍生出來的語言。

人造語言：為了特定目的所設計、並有固定的規則。

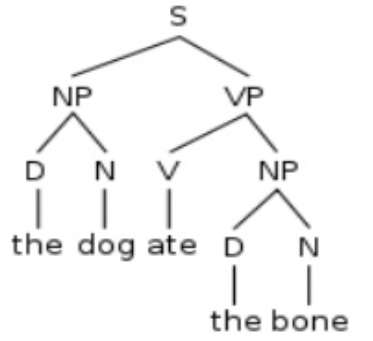
1. BNF 與生成語法



1. 運算式語法

****

1. 英文語法

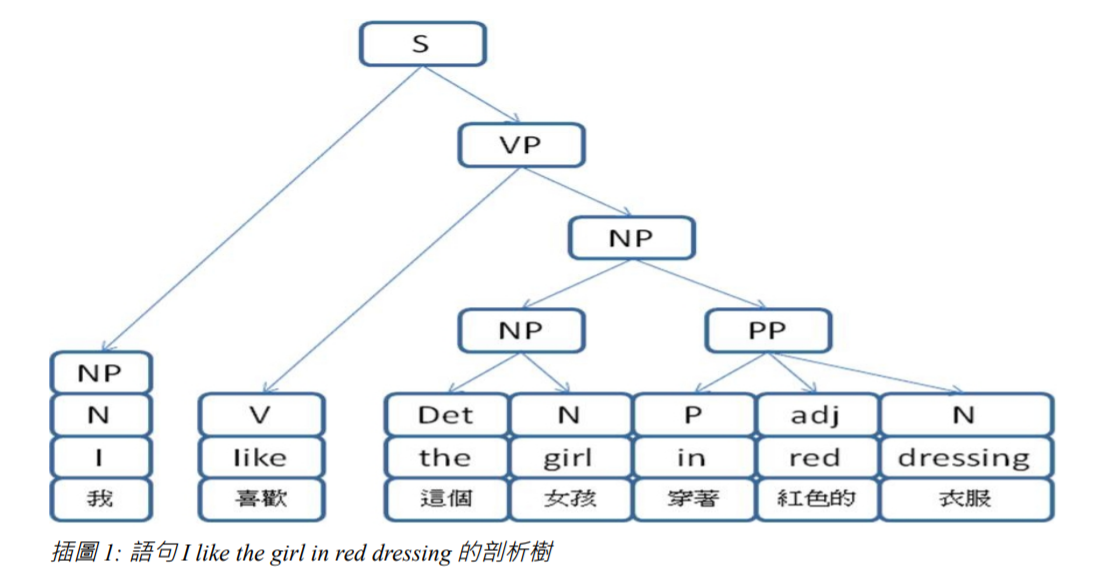
****



**5. 語言合成技術：**使用 遞迴呼叫的方式，撰寫 BNF 語法的生成程式

**6. 機器翻譯**

1. 簡易逐字翻譯系統：對照字典、辭庫直翻
2. 規則式翻譯系統：依靠是語法規則以及語意規則進行翻譯



取自：中研院自然語言處理小組的範例

會遇到的問題：

1. 解歧義問題 (Word Sense Disambiguation)：一字多義的問題。

2. 語法剖析的問題 (Parsing)：如何將語言轉化成語法樹

3.樹狀結構轉換的問題 (Structure Transformation)轉換成更好的語法樹

4.目標語句合成的問題 (Sentence Generation)

**第十週**

1. 科學計算

代數學

sympy. Simplify()可用來簡化算式

**第十一週**

1. 科學計算

2. 迭代法與不動點定理

迭代法必須能寫成x2 =f(x1)

當有x符合x=f(x)，即為不動點

使用迭代法可能會發生下列幾種狀況：

1. 收斂 xn = f(xn)

2. 發散 xn 越來越大

3. 震盪 xn = f(x[n-1]), ...., x[n-1] = f(xn), 且 xn != x[n-1]

4. 混屯 只在有限範圍內移動，且不符合以上三項。

而只有在1的狀況，我們才能找到不動點

2. EM 演算法：用迭代法來求最大機率模型的方法。

3. KMEAN 演算法：用迭代進行機器學習的方法。

**第十二週**

1. 中央極限定理：是機率理論及統計學中最重要且常用的結果之一。對許多初學者而言，卻是一個不容易瞭解的抽象概念。為了讓初學者比較容易瞭解及掌握中央極限定理的基本概念，這裡將藉由網路互動式模擬程式，來讓初學者從互動的實驗中理解中央極限定理的基本概念。

取自：<http://www.math.nsysu.edu.tw/StatDemo/CentralLimitTheorem/CentralLimit.html>

1. k-近鄰演算法(KNN演算法)

又稱為k-近分類演算法

根據不同特徵值之間的距離進行分類

是較為簡單的機器學習方法

取自：<https://zh.wikipedia.org/wiki/K-%E8%BF%91%E9%82%BB%E7%AE%97%E6%B3%95>

1. K-平均演算法(k-means)

將n個點依其欲分類的類別畫分

”物以類聚”的概念，越加相似的點距離越近

屬於非監督式學習

<https://medium.com/@chih.sheng.huang821/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E9%9B%86%E7%BE%A4%E5%88%86%E6%9E%90-k-means-clustering-e608a7fe1b43>

**第十三週**

**第十四週**

1. 隱馬可夫模型(hmm)

馬可夫模型：選一個狀態作為起點，然後沿著邊隨意走訪任何一個狀態，一直走一直走，沿途累計機率，走累了就停在某個狀態。

取自：<http://www.csie.ntnu.edu.tw/~u91029/HiddenMarkovModel.html>

1. 維特比演算法：係常用在交織碼解碼的一種快速最大可能性信號路徑搜尋方法。它利用交織碼為一階馬可夫鏈的特性，在搜尋最佳解碼路徑時，對每個時間點上的每一個路徑節點，比對所有進入此節點的路徑的可能性後，只保留一條最可能路徑往下傳遞，因此與搜尋所有可能路徑的方式相比，具有複雜度與解碼長度成線性比例，所需記憶空間小與計算負荷輕的優點。

取自：<https://pedia.cloud.edu.tw/Entry/Detail/?title=%E7%B6%AD%E7%89%B9%E6%AF%94%E6%BC%94%E7%AE%97%E6%B3%95>

用於馬可夫模型上

用於計算HMM最有可能的隱序列

1. 遺傳演算法genetic algorithm (GA)：是計算數學中用於解決最佳化的搜尋演算法，是進化演算法的一種。 進化演算法最初是借鑑了進化生物學中的一些現象而發展起來的，這些現象包括遺傳、突變、自然選擇以及雜交等。 遺傳演算法通常實現方式為一種電腦類比。

取自：<https://zh.wikipedia.org/zh-tw/%E9%81%97%E4%BC%A0%E7%AE%97%E6%B3%95>

模仿自然界演化現象的算法

主要為模仿兩性生殖的機制

**第十五週**

1. EM 演算法

又稱期望最大化算法

希望找到一個模型，且模型的參數能使樣本的機率對最大化

期望最大化的目的在於尋找不動點

算法主要流程如下：

1.初始化參數

2.重複E、M步驟直到收斂(找到不動點)

2. 深度學習

Pytorch

\*架設venv、torch、pytorch

**第十六週**

1. 深度學習即為神經網路

2. 捲積神經網路CNN

常用於影像處理、圖片辨識

從初始特徵開始捲積綜合出上層特徵

3. 循環神經網路RNN

nlp

循環的特性可以造成記憶效應，可以記住歷史訊息並預測

LSTM長短期記憶網路：重要的記憶可以保存得更久，不重要的較早遺忘

包含了鎖鏈網路的拆解與數學原理

4. 生成對抗網路GAN

非監督式

透過兩個神經網路互相博奕的方式訓練