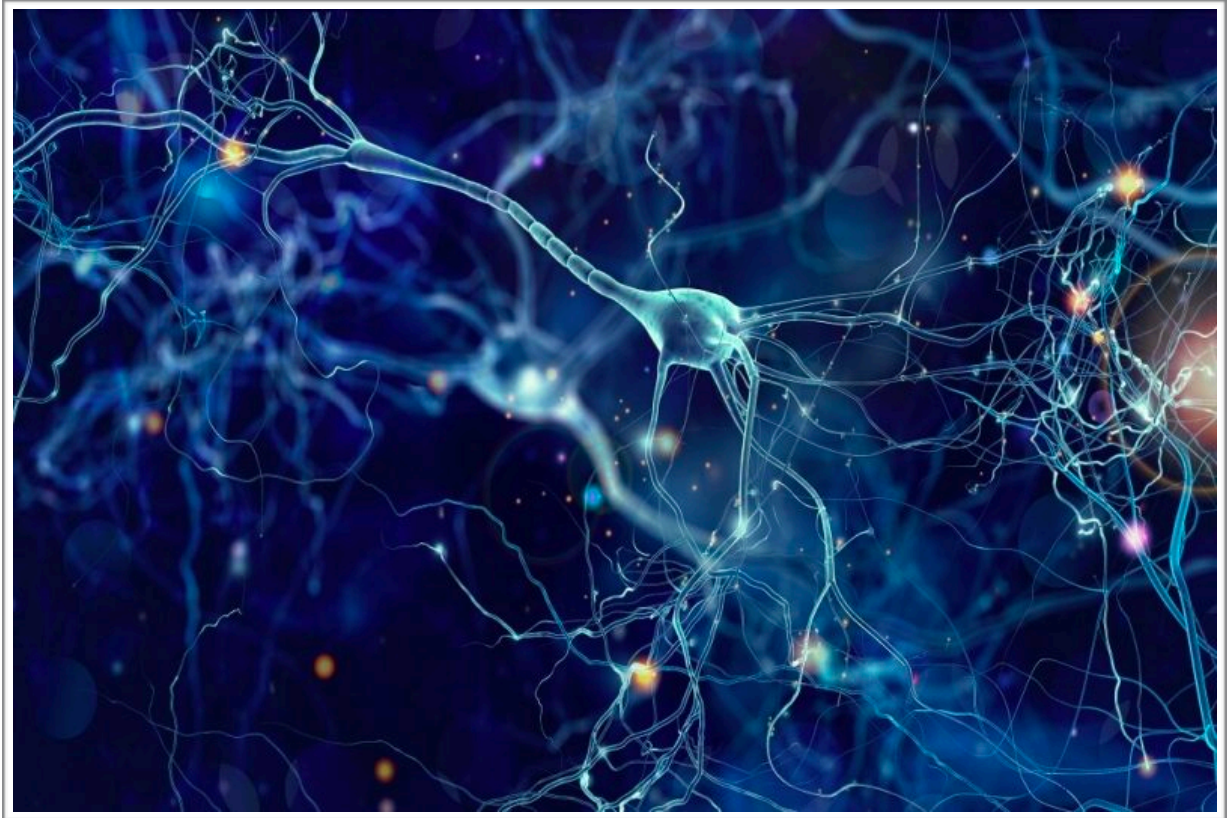


工業4.0框架下的機器學習-神經網路

Neural network of machine learning



陳品洋 資管三 408530014

12/29/21

工業4.0與法律 簡立仁教授 廖蕙玟教授（依筆畫）

專題大綱

abstract

前言	introduction	3
神經網路	neural network	4
神經網路實作過程	neural network implementation	5
參考	reference	8
附錄	appendix	8

前言

introduction

製作動機motivation

為了更強化自己對於科技發展趨勢的掌握，深夜中細細思考，若工業4.0帶給世界的功效之一是能夠使許多產業達到智能製造，那是否我們也應該了解機器是怎麼去思考、怎麼去因應現實世界中各種不同的複雜環境並從中獲取經驗的？若智慧生產中的某些產品無法符合人類需要，那這些程式碼智慧該如何加以學習人類的需要並改善所生產的產品？

智慧，某種程度上需要一定的自省能力，這也是類神經網路與機器學習所能做到的功能。而本報告我將不會只侷限在理論的闡述，我將會實實在在、盡我所能提供給讀者我是如何實現一個簡單的神經網路，並在後面會附上我的github連結，裡頭的程式碼我很歡迎老師或同學細細品味。



神經網路

Neural network

什麼是機器學習？

機器學習首先著重在兩個重點：

- 1.機器學習這門學問主要研究對象的是人工智慧，具體來說，即如何透過學習經驗改善自身演算法。
- 2.此門科學是用資料或從前收集的經驗來最佳化電腦程式的效能。

機器學習涉及的層面包含許多古典學科，包含統計學、機率論、微積分、逼近論等等。透過一連串的演算法從一堆資料中得知其「**規律**」，並針對未來作出改善(fitness)與規劃。

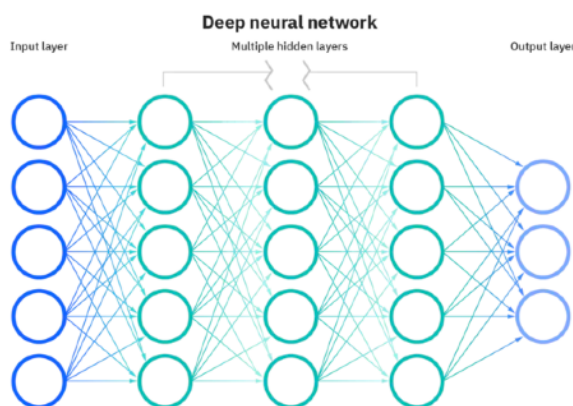
什麼是類神經網路？

又稱人工神經網路(Artificial neural network,ANN)，簡言之就是模仿生物神經網路的組織與功用製造出的演算法或數學模型。

ANN通常由一個節點層（layer）組成，其中節點層包含一個輸出層（output layer），一個輸入層(input layer)及一個以上的隱藏層(hidden layer)。

每一個層級裡的節點（或稱神經元neural）會連接至其它個節點，並

且每一節點都設有**閾值**(threshold)及加權值。若任何一個神經元所輸出的值高於指定的閾值，則開啟此閥門讓資料能夠進到下一個節點，反之則否。所以隨著時間的推移，系統能夠自行調整閾值讓能夠通過的資料變得更精確，並且使通過閾值的物件或資料（基因genes）加以繁衍，即達成進化論中的「**演化**」。



神經網路製作過程

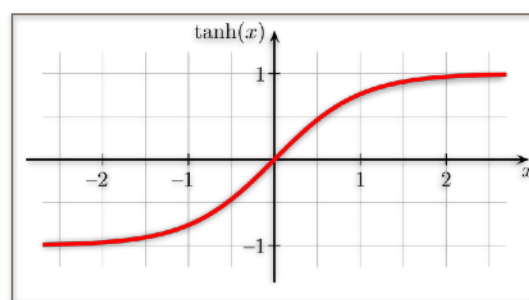
neural network implementation

Neat-python

在Neat-python的實現中，維護了一組個體基因組。每個基因組包含兩組描述如何構建人工神經網路的基因：

- 1.節點基因，每個基因指定一個神經元。
- 2.連接基因，每個基因指定神經元之間的單個連接。

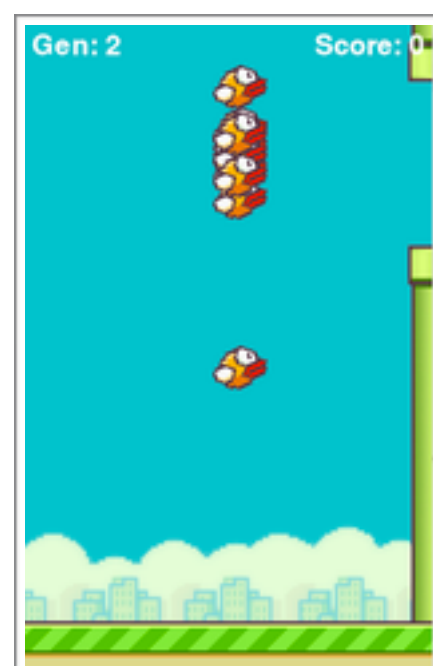
為了演化出一個問題的解決方案，用戶必須提供一個適應度函數（在此專題為tanh函數，因為此函數在平面座標上為1至-1），它計算一個單一的實數，表明單個基因組的質量：解決問題的能力越好意味著得分越高。該算法通過用戶指定的代數進行，每一代都是通過繁殖（有性或無性）和上一代最適合個體的突變產生的。



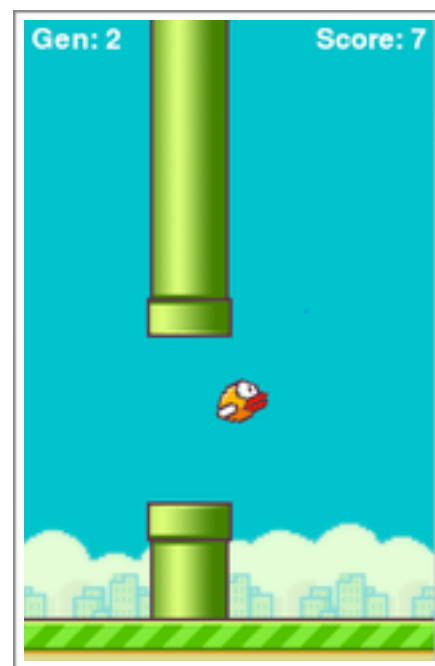
實現構思

這次將先撰寫簡單的Flappy Bird遊戲，一般狀態下此遊戲的玩法是玩家利用點擊控制螢幕中的鳥，點擊一次此鳥就會拍動一次翅膀並往上飛，目標是盡可能在碰到上下水管之前獲取最高分數。所以我們必須撰寫出一個這樣的小遊戲。

再來就是實現神經網路的構思，我會先在遊戲開始前多生成幾隻鳥（第一代）並且這些鳥都能夠記錄數據（包含個體的自適應力等等），讓他們隨意飛，所以會有許多鳥因為過不去水管而死掉，但我們要的是那些活過去的。



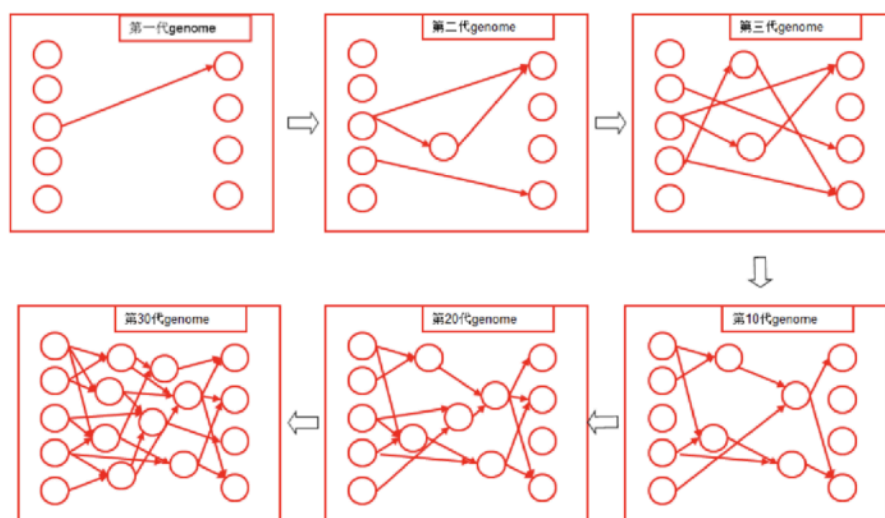
最後，在同一個世代（generation）會剩下一隻適應力最好的鳥，待其死後，取其數據並以此為中心，依據我們設定的變異值來生產下一代更優良適應力的鳥。以此來實現機器針對經驗學習的目標。



這次所用的NEAT核心

Neuro Evolution of Augmenting Topologies (拓撲學) (NEAT),

1. Input neural (輸入神經元) : Bird.y(鳥的y軸), Top pipe (上水管的高度), bottom pipe (下水管的高度)
2. Output neural (輸出神經元) : Jump (跳)
3. Activation (激活函數) : tanh (輸入神經元之值若大於0則跳，用來判斷鳥與水管的距離為多少時該跳)
4. population size(一個世代的數量)
5. fitness function (取其最遠的)
6. max generation (若學習次數多於我們設置的最高次數則表示此系統效率不如預期要加以調整)



圖示

```
pygame 2.0.1 (SDL 2.0.14, Python 3.8.3)
Hello from the pygame community. https://www.pygame.org/contribute.html

***** Running generation 0 *****

Population's average fitness: 4.31200 stdev: 3.28266
Best fitness: 22.60000 - size: (1, 3) - species 1 - id 6
Average adjusted fitness: 0.099
Mean genetic distance 1.690, standard deviation 0.531
Population of 50 members in 2 species:
  ID  age  size  fitness  adj fit  stag
  ===  ==  ===  =====  =====  =====
    1   0   29    22.6    0.102    0
    2   0   21     7.5    0.095    0
Total extinctions: 0
Generation time: 6.913 sec

***** Running generation 1 *****
```

上圖是在console畫面我們需要印出的基本資料（此圖為第0代的資料），包含初始適應值、第幾代、系統自行調整的適用值、標準差、這一代平均跑的距離等等

```
[[NEAT]
fitness_criterion      = max
fitness_threshold      = 100
pop_size               = 50
reset_on_extinction    = False
```

上圖為我們所設定的適應值，包含閾值、一代的鳥數、若全死是否滅絕（否，這樣才能繼續讓下一代學習啊！）

```
# node bias options
bias_init_mean         = 0.0
bias_init_stdev         = 1.0
bias_max_value          = 30.0
bias_min_value          = -30.0
bias_mutate_power       = 0.5
bias_mutate_rate        = 0.7
bias_replace_rate       = 0.1
```

這裡是設定鳥類必須依照上一代最佳為中心，然後依照我們所設定的變異量產生50隻不同運動的鳥。

參考reference:

1. (維基百科) <https://zh.wikipedia.org/wiki/%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0> page4.
- 2.(IBM/2020-8-17) <https://www.ibm.com/tw-zh/cloud/learn/neural-networks> page4.
- 3.(Tech with Tim/2019/8/10)https://www.youtube.com/watch?v=MMxFDaIOHsE&list=PLzMcbGfZo4-lwGZWXz5Qgta_YNX3_vLS2&ab_channel=TechWithTim page5,6

附錄appendix:

我的github: <https://github.com/aStudentLoningToGrow/AI-flappy>

順帶一提，github已於8月份開始停用帳號密碼上傳資料了，因為要加強防止駭客入侵的能力，github改用權杖加密(security token)上傳方法加強防護。

	aStudentLoningToGrow first edition	ed69e13 3 days ago	🕒 1 commit
📁	imgs	first edition	3 days ago
📄	AI-flappy.py	first edition	3 days ago
📄	[NEAT]	first edition	3 days ago
📄	config-feedforward.txt	first edition	3 days ago