

# Artificial Neural Networks (人工神經網路)

Reference :

1. Artificial Neural Networks - Dr. Tun-Wen Pai
2. [Neural Networks Pt. 1: Inside the Black Box](#)

## 概述

人工神經網路使用了一種曲線，能夠近乎完美的符合資料集。

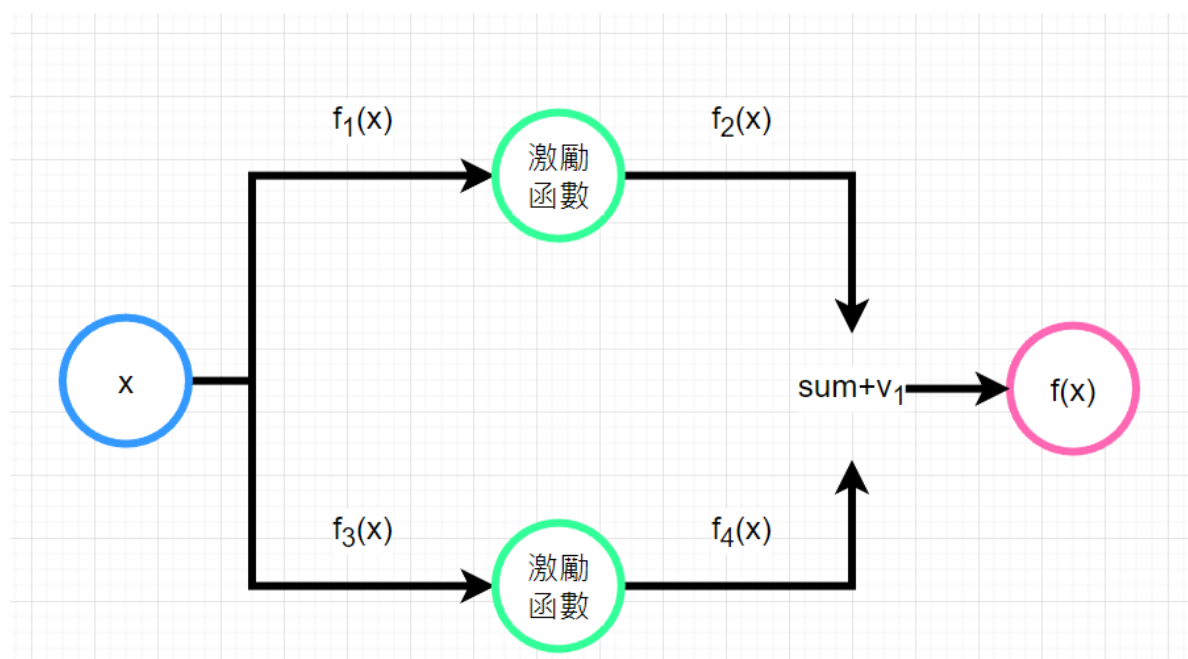
使用的曲線為激勵函數，利用參數、權重等等來製作，藉由神經元來構造曲線，進而符合資料集。

我們可以把他想成將一個參數放入 input 神經元後。

這個 input 神經元會隨著箭頭進入到 Hidden layer 的神經元，通常是一個激勵函數。

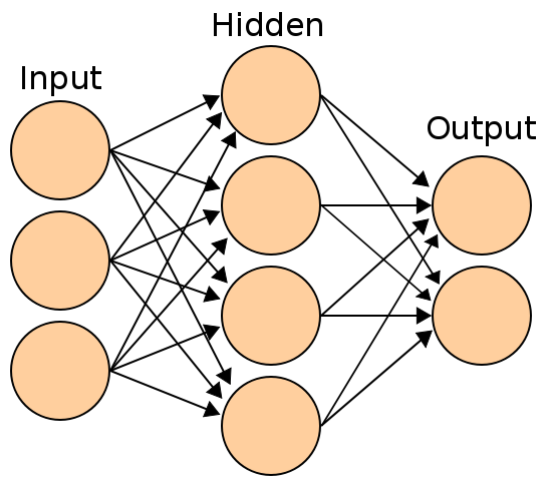
箭頭會逐漸塑造激勵函數，直到 output 神經元將曲線輸出。

下圖是一個簡單的人工神經網路，我們將藍色圈圈稱為 input，綠色圈圈稱為 hidden，粉紅色圈圈稱為 output



而實際上可能會這麼複雜：

Artificial neural network.svg - 維基百科，自由嘅百科全書



## 激勵函數

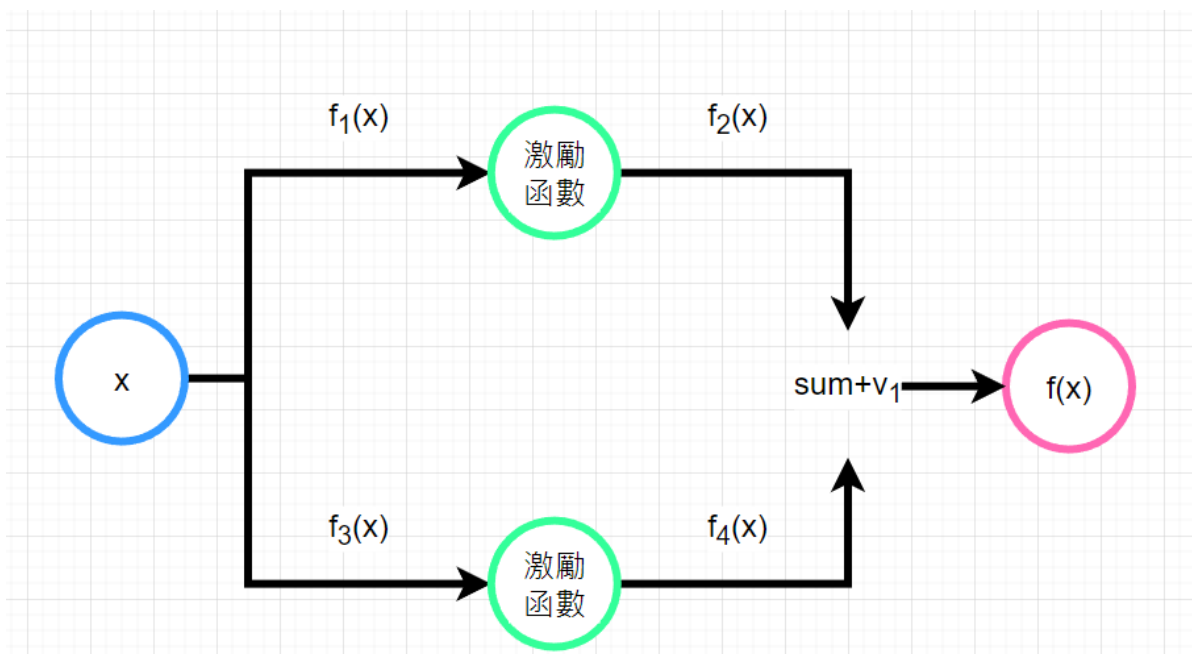
激勵函數在塑造曲線的時候扮演了重要的角色，主要分成四種：

1. Tanh :  $f(x) = \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
2. Sigmoid / Logistic :  $f(x) = \frac{1}{1 + e^{-x}}$
3. ReLu :  $f(x) = x^+ = \max(0, x)$
4. Softplus :  $f(x) = \ln(1 + e^x)$

所謂的激勵函數本質上就是函數，可以想像成把參數放入激勵函數後，可以使激勵函數最後塑造出我們想要的曲線。

## 建構人工神經網路

我們以簡單的例子來說明，以下圖為例。



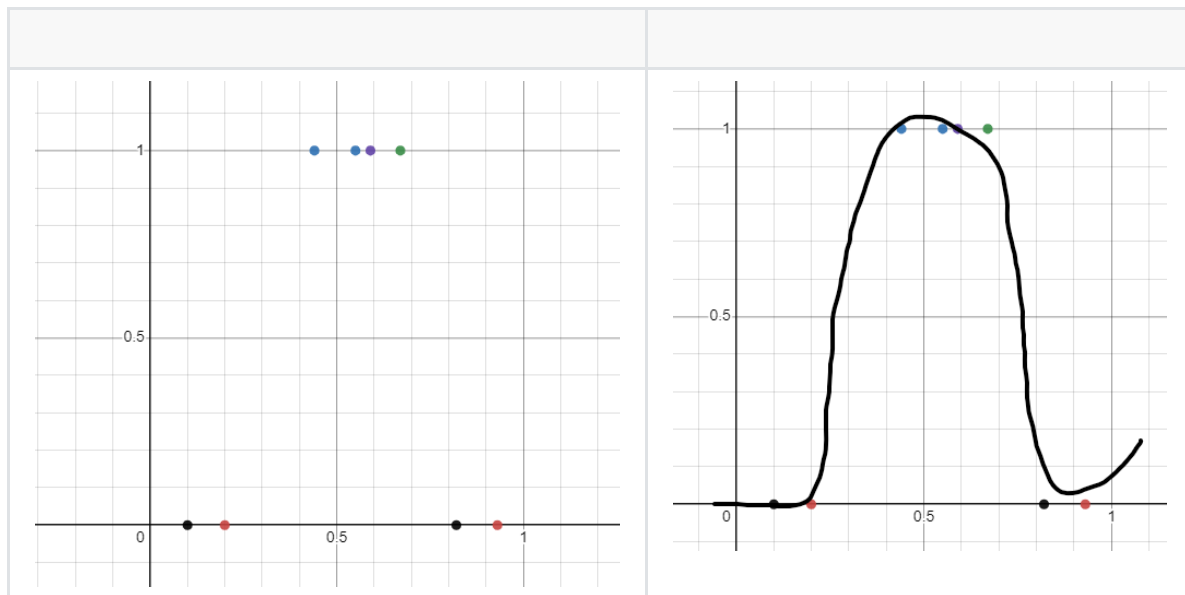
我們有一個簡單的資料集，分成三類，值域界於 0 到 1：

1. 服用少數量 ntut-xuan 筆記的人 → 考不好 (0)
2. 服用中等數量 ntut-xuan 筆記的人 → 考得好 (1)
3. 服用多數量 ntut-xuan 筆記的人 → 考不好 (0)

可以得到左圖。

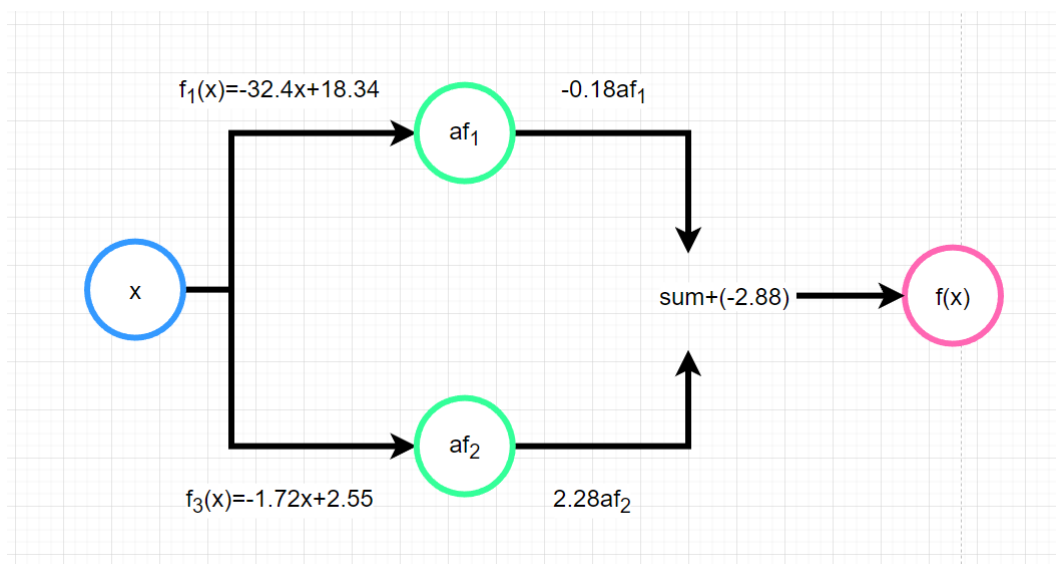
此時我們可能會想要用一條直線來分割這些資料，但這條直線可能不存在，因為不管怎麼畫都沒有辦法概括完全的資料。

如果這時候有一條神奇的函數來讓這些資料 match 就好，就像右圖。

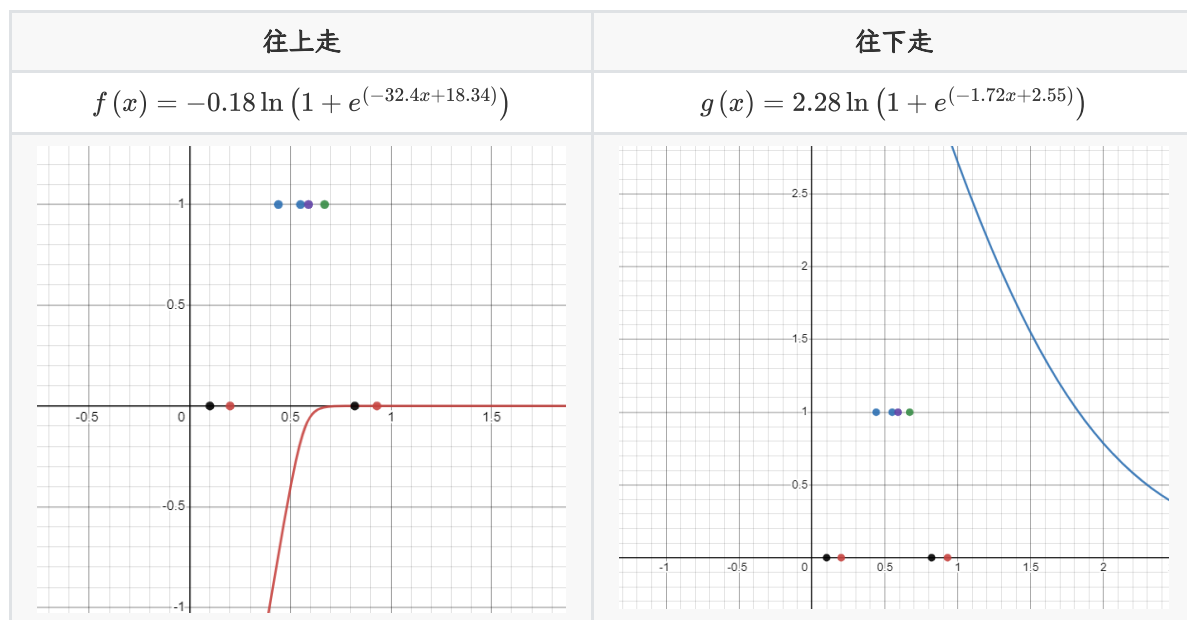


我們假設已經知道了類神經網路的  $f_1, f_2, f_3, f_4, v_1$  參數，我們可以這樣建構我們的類神經網路。

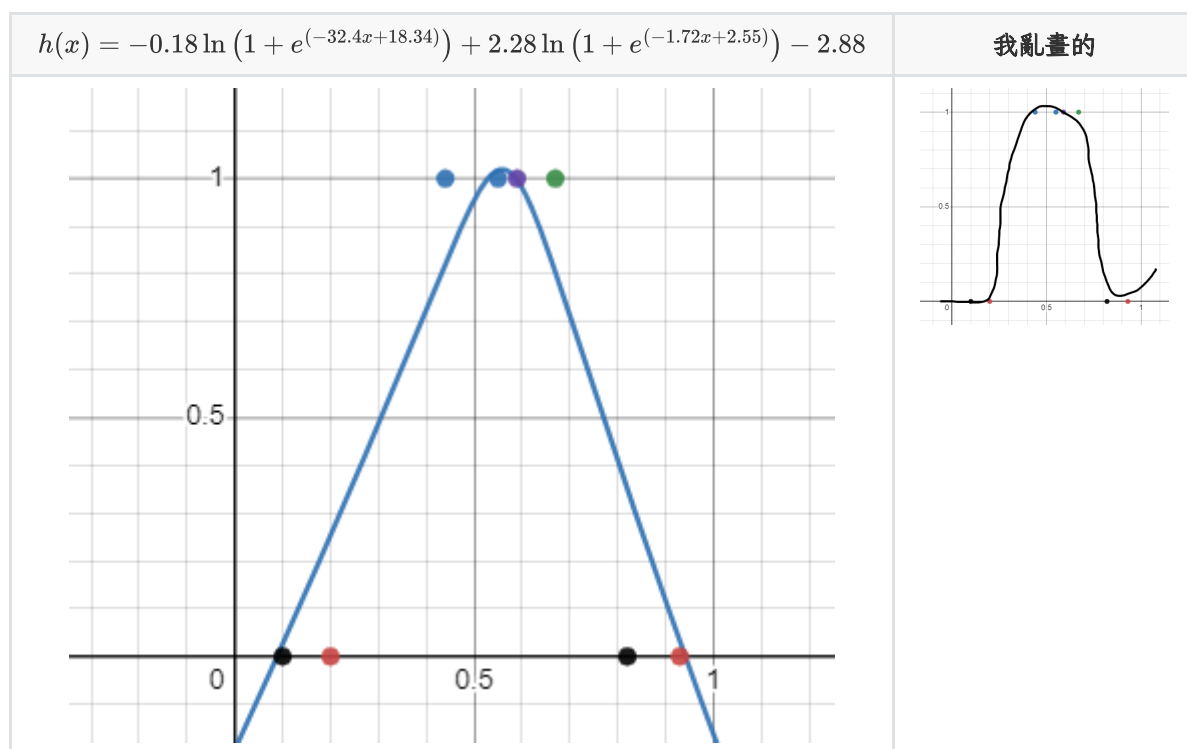
假設我們使用的激勵函數  $af_1, af_2$  為Softplus:  $f(x) = \ln(1 + e^x)$



我們可以由我們建構的類神經網路，往上走建構出一條曲線，往下走建構出另一條曲線，如下圖：



最後將兩個曲線加起來，並減去 2.88，得到以下的曲線，就能夠得到我們幾乎亂畫出來的曲線了！



這時候我們就可以用這條曲線來判別我們特定資料集所出現的結果，所以人工神經網路理論上能夠成功分類所有的資料。

問題在於如何找出參數，來建構我們想要的曲線。