

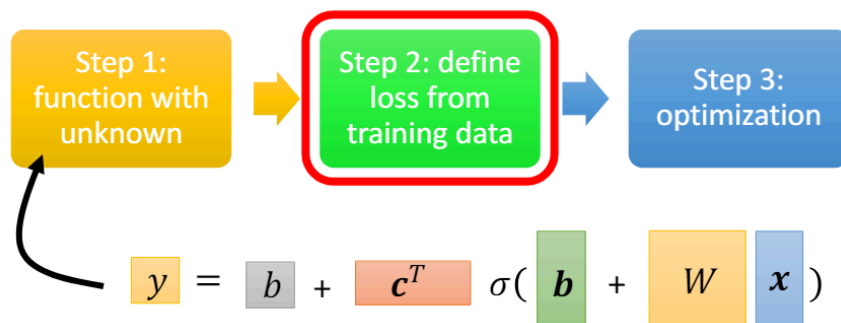
## 【機器學習2021】預測本頻道觀看人數 (下) - 深度學習基本概念簡介-筆記

來源: 台灣大學教授 Hung-yi Lee

▶ 【機器學習2021】預測本頻道觀看人數 (下) - 深度學習基本概念簡介

### 本章節要點:

- 重新定義了Model式子



### Linear Model問題: 面臨Model bias

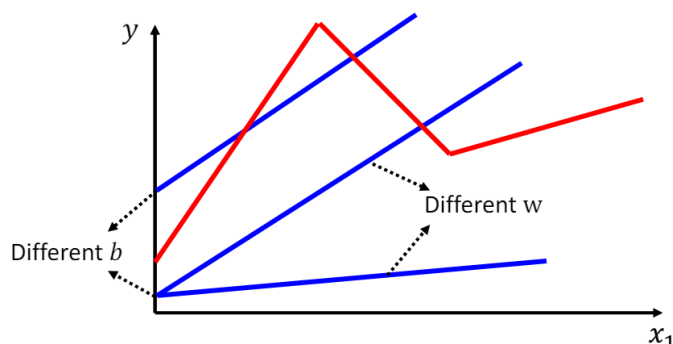
**Model bias:** 來自Model限制的問題

原因: 無論參數 $w$ 、 $b$ 怎麼運作, 都匯市

input愈大, output愈大

EX: 前一天觀看人數愈多, 隔天一定愈多

(來自老師上堂課的例子)



### 怎麼產生紅色那條線呢?

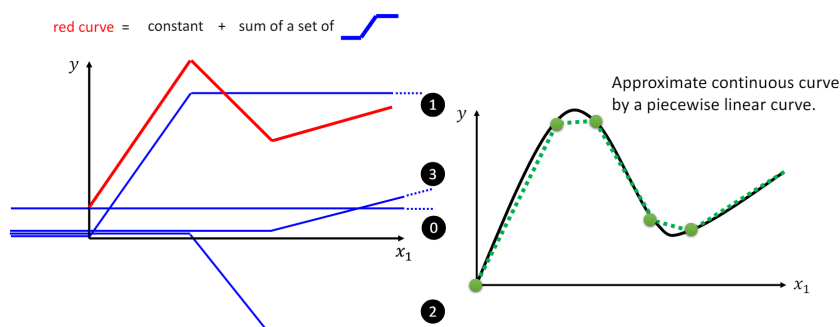
由非常多條的藍線形成

紅線 =  $0 + (1 + 3 + 2)$

\*每個轉彎處都可以由再+一條藍線產生

\*就算是連續的線段, 也可以像是

Piecewise linear curve一樣



### 這條藍線又是怎麼產生的呢?

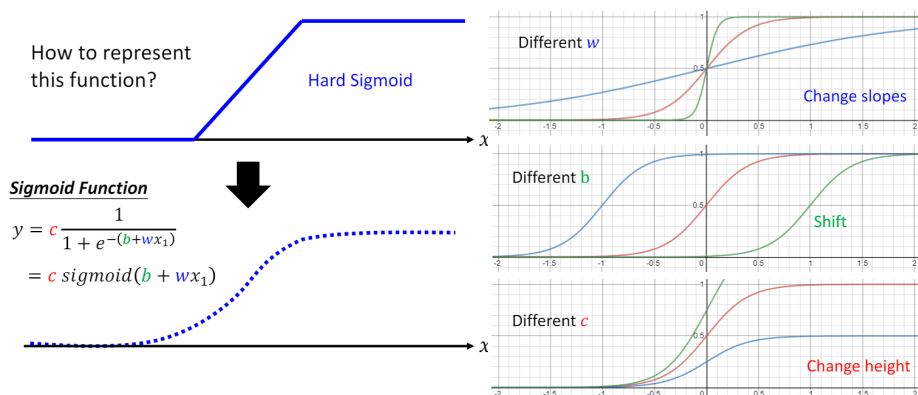
由sigmoid function逼近

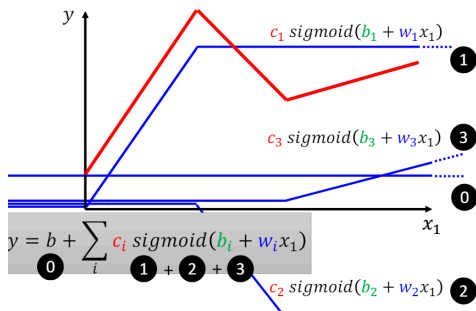
Function解釋:

當 $x_1$ 趨近無窮大時,  $e$ 消失, output =  $c$

當 $x_1$ 趨近負很大時,  $e$ 很大, output = 0

不同參數對線的型態改變





最終紅線的結果

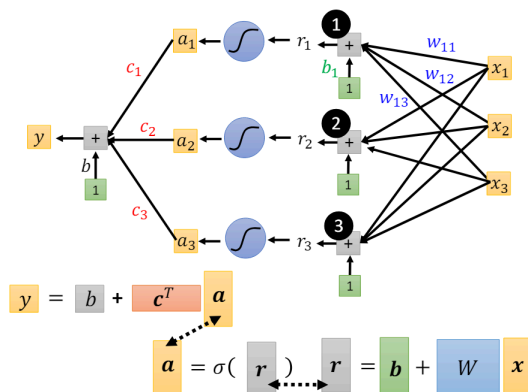
$$y = b + wx_1$$

$$y = b + \sum_i c_i \text{sigmoid}(b_i + w_i x_1)$$

$$y = b + \sum_j w_j x_j$$

$$y = b + \sum_i c_i \text{sigmoid}\left(b_i + \sum_j w_{ij} x_j\right)$$

最終新Model式子推導



<-最終式子簡化理解結果

$$y = b + c^T \sigma(b + Wx)$$

feature  $x$

Unknown parameters

Rows of  $W$

$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \end{bmatrix}$

重新定義未知參數們->為theta

### optimization步驟:

選一個點當初始點 ->  
第一遍計算所有參數的gradient ->  
第一次更新所有參數 ->

$$\theta^* = \arg \min_{\theta} L$$

➤ (Randomly) Pick initial values  $\theta^0$

➤ Compute gradient  $g = \nabla L(\theta^0)$

$$\theta^1 \leftarrow \theta^0 - \eta g$$

➤ Compute gradient  $g = \nabla L(\theta^1)$

$$\theta^2 \leftarrow \theta^1 - \eta g$$

➤ Compute gradient  $g = \nabla L(\theta^2)$

$$\theta^3 \leftarrow \theta^2 - \eta g$$

### 實例解析

把Loss分成很多batch計算，每一次計算都會更新

\*update: 每一次更新

\*epoch: 把所有batch看過一遍

$$\theta^* = \arg \min_{\theta} L$$

➤ (Randomly) Pick initial values  $\theta^0$

➤ Compute gradient  $g = \nabla L^1(\theta^0)$

$$\text{update } \theta^1 \leftarrow \theta^0 - \eta g$$

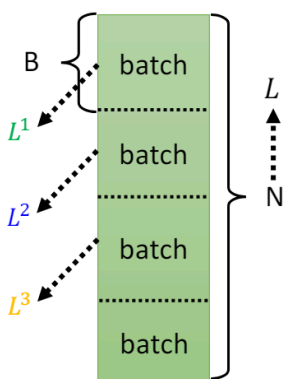
➤ Compute gradient  $g = \nabla L^2(\theta^1)$

$$\text{update } \theta^2 \leftarrow \theta^1 - \eta g$$

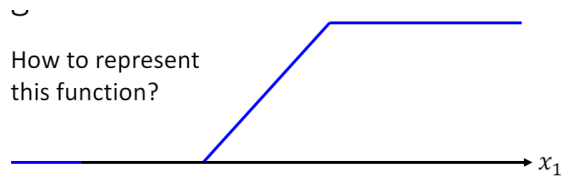
➤ Compute gradient  $g = \nabla L^3(\theta^2)$

$$\text{update } \theta^3 \leftarrow \theta^2 - \eta g$$

1 epoch = see all the batches once



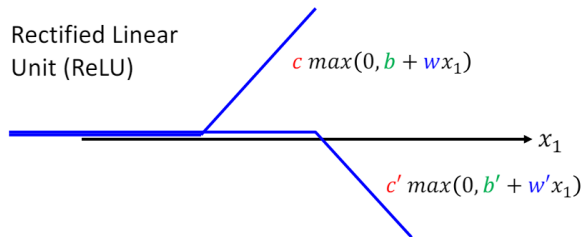
Sigmoid ->



ReLU ->

只有sigmoid的一半

他們都叫做: Activation function



一層neuron (ex: sigmoid、ReLU) = 一層hidden layer

很多layers = deep learning

**Overfitting:** 當訓練的Loss下降，預測的Loss上升

**備註:**

此筆記中的所有參考資料皆來自Hung-yi Lee 教授的上課影片和投影片，非常感謝Hung-yi Lee 教授的精彩課程，讓我重拾對人工智慧的興趣，不再覺得人工智慧像是一團迷霧~

此筆記目的是為了可以讓我更好的記得上課內容，並且做自己的學習紀錄~

若有侵權會停止分享這個筆記!