

深度學習

指導教授 李俊宏 教授

主講人 蘇禎佑

神經元

神經元

- Neural Network 是參考生物神經系統的結構，神經元(Neuron)之間互相連結，由外部神經元接收信號，再層層傳導至其他神經元，最後作出反應的過程。

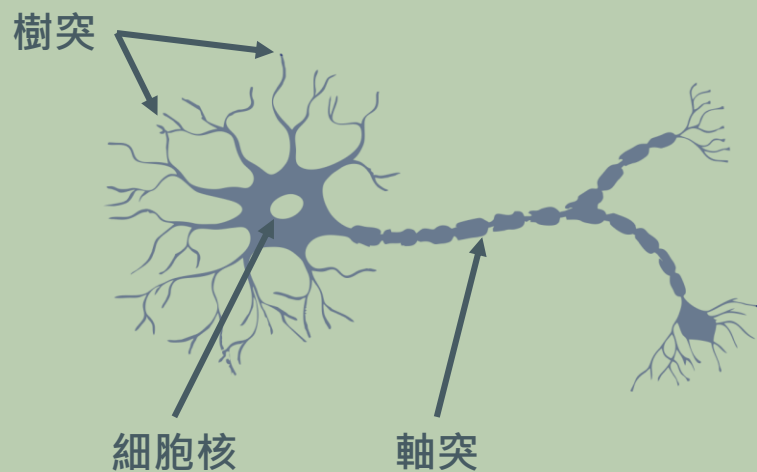


圖. 神經元構造

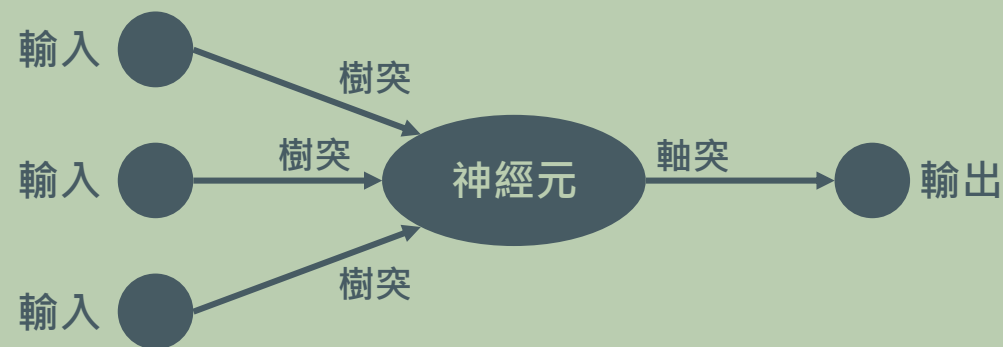
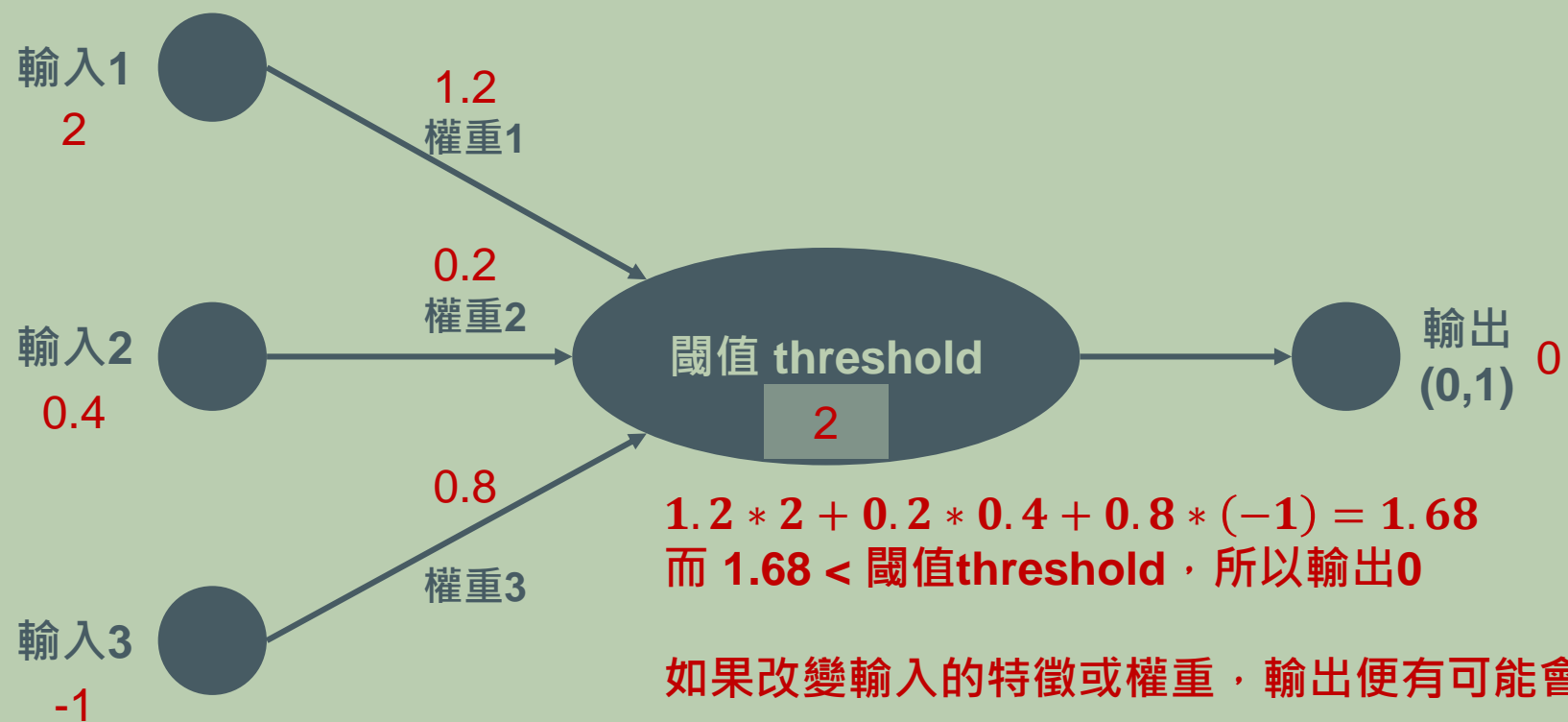


圖. 進一步抽象化神經元的構造

人工神經元



感知器 Perceptron

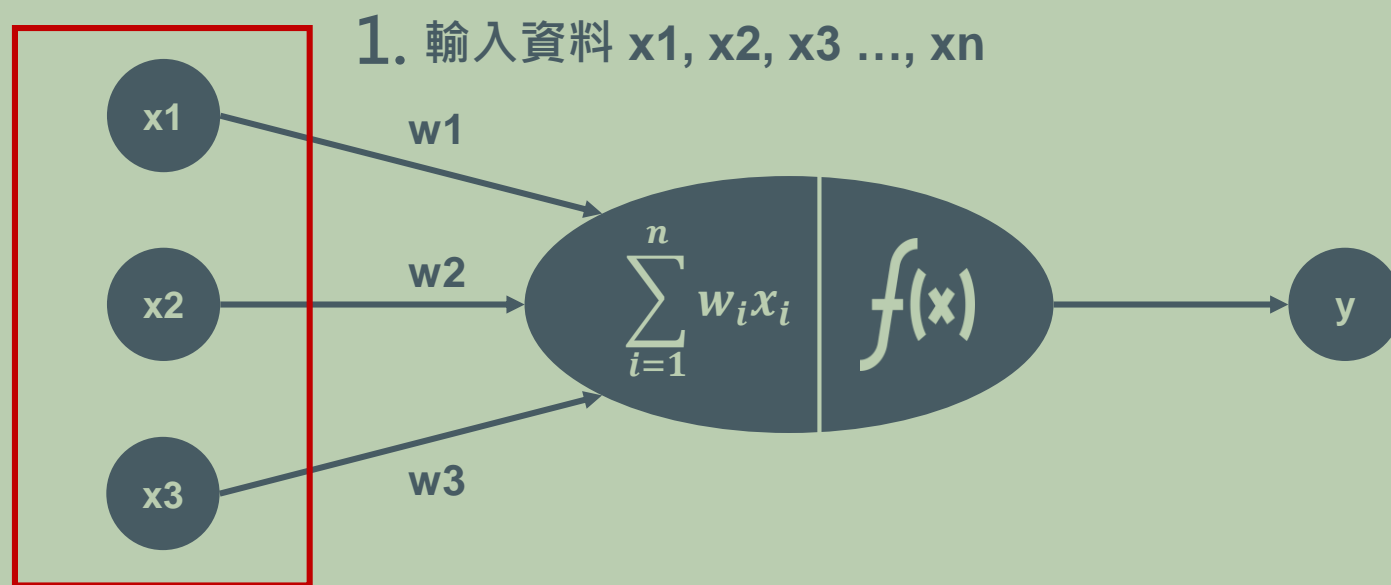
感知器

輸入資料

進行加總

加上偏差值

判斷後輸出



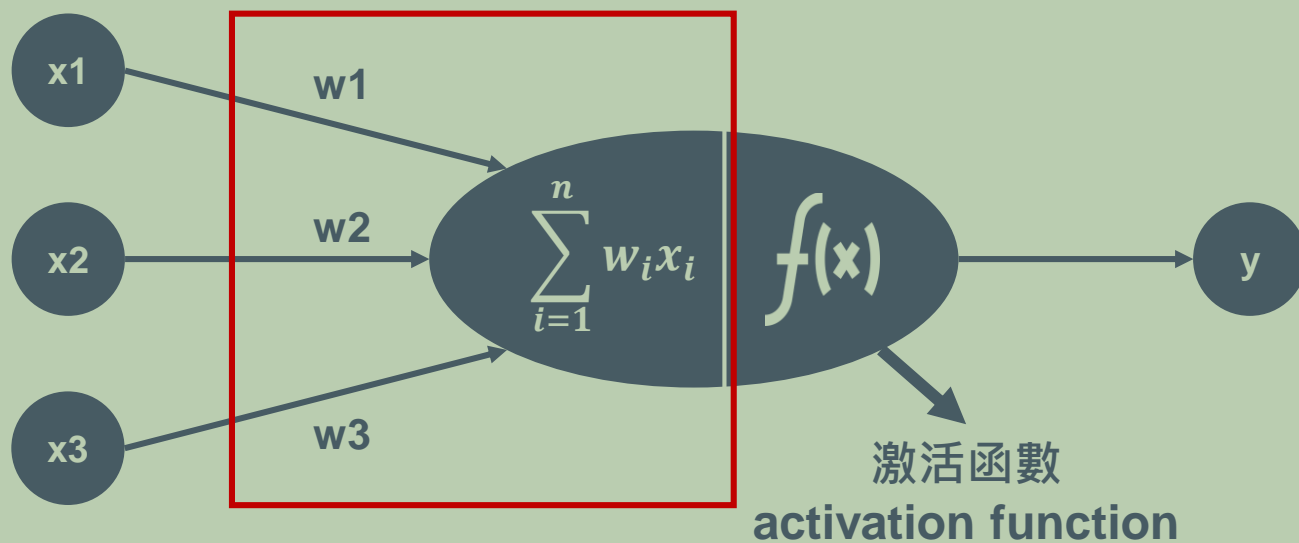
感知器

輸入資料

進行加總

加上偏差值

判斷後輸出



2. 將輸入的資料 ($x_1 \dots$)
乘上相對應的權重 ($w_1 \dots$)
然後將它們相加

$$\sum_{i=1}^n w_i x_i$$

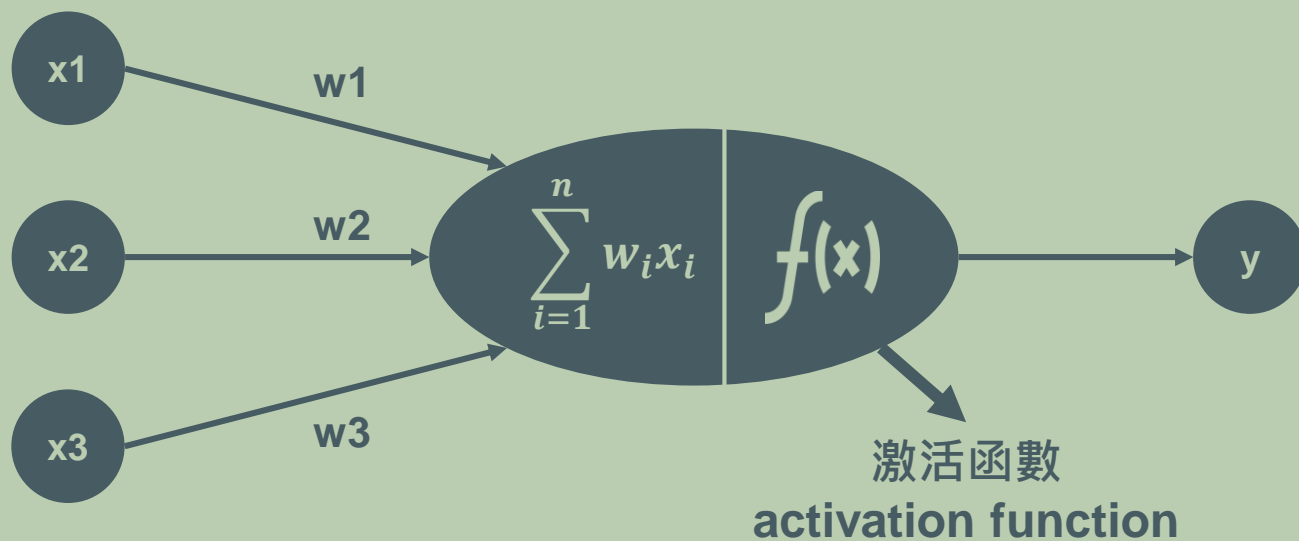
感知器

輸入資料

進行加總

加上偏差值

判斷後輸出



3. 將加總後的值，加上一個偏差值 b_{ias}

$$z = f\left(\sum_{i=1}^n w_i x_i\right) + b_{ias}$$

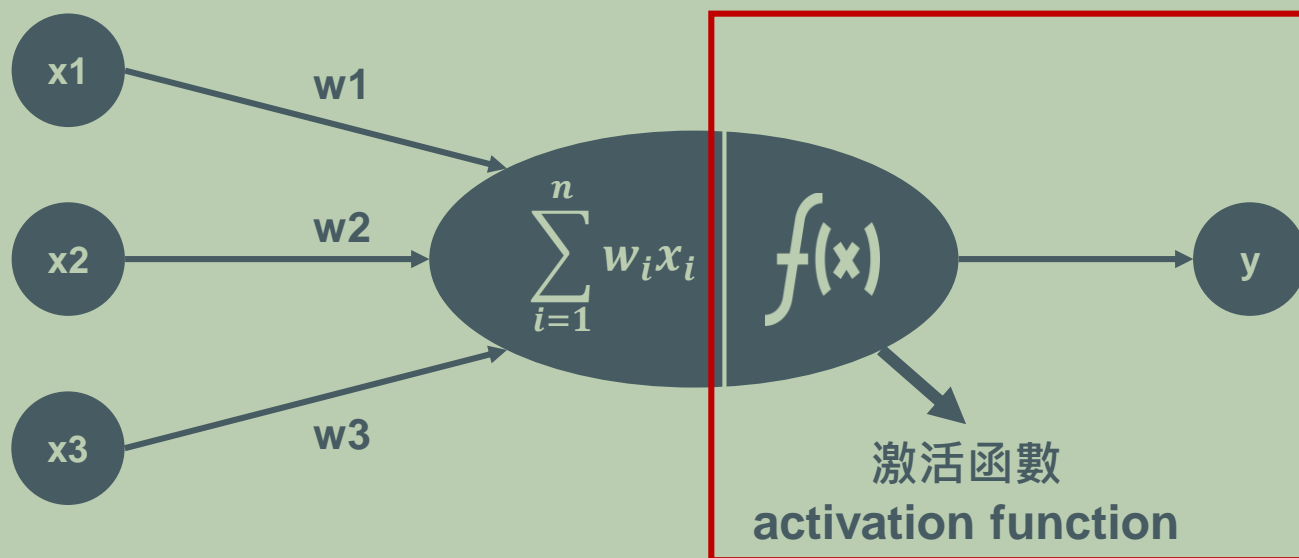
感知器

輸入資料

進行加總

加上偏差值

判斷後輸出



4. 將輸出的值，作為激活函數的輸入
從而判斷輸入資料的種類

這裡是採用階梯函數 **step function**

$$f(z) \begin{cases} 1, & \text{if } z > s \\ 0, & \text{else} \end{cases}$$

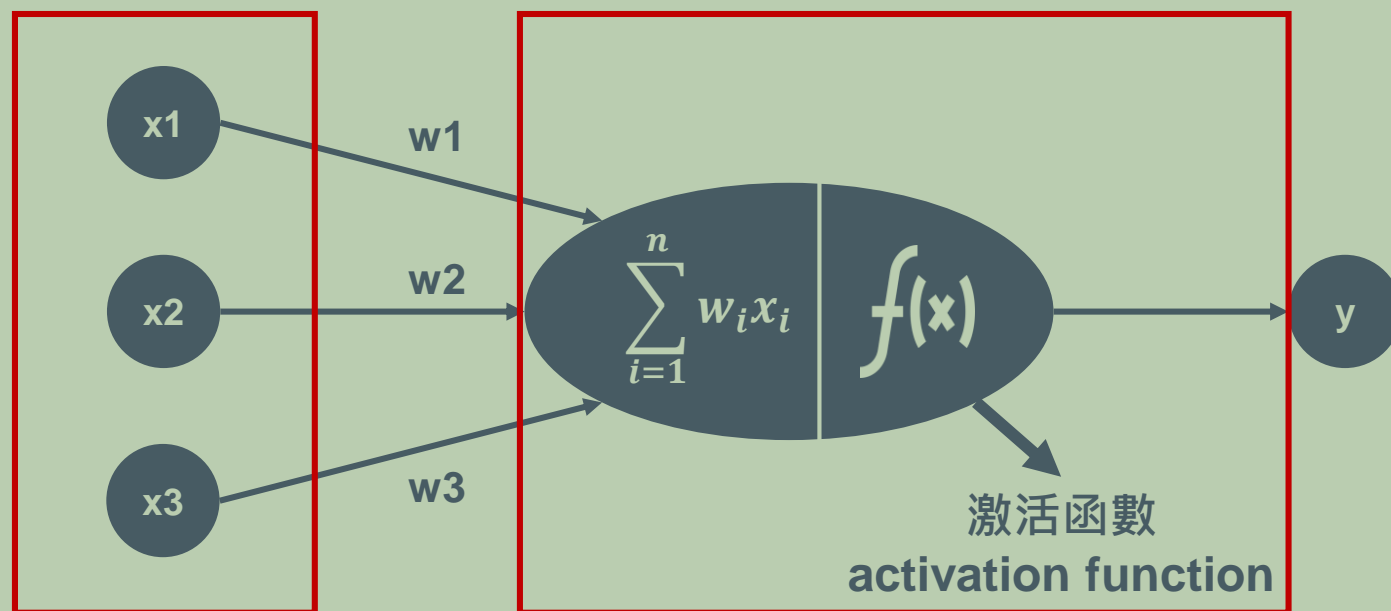
感知器

1. 資料輸入的部份為輸入層，也就是把資料拿給機器看，讓他學習，或者稱為”試著預測”

輸入層

輸出層

2. 資料輸出的部份，則為看過資料後，嘗試預測的判斷依據與結果。



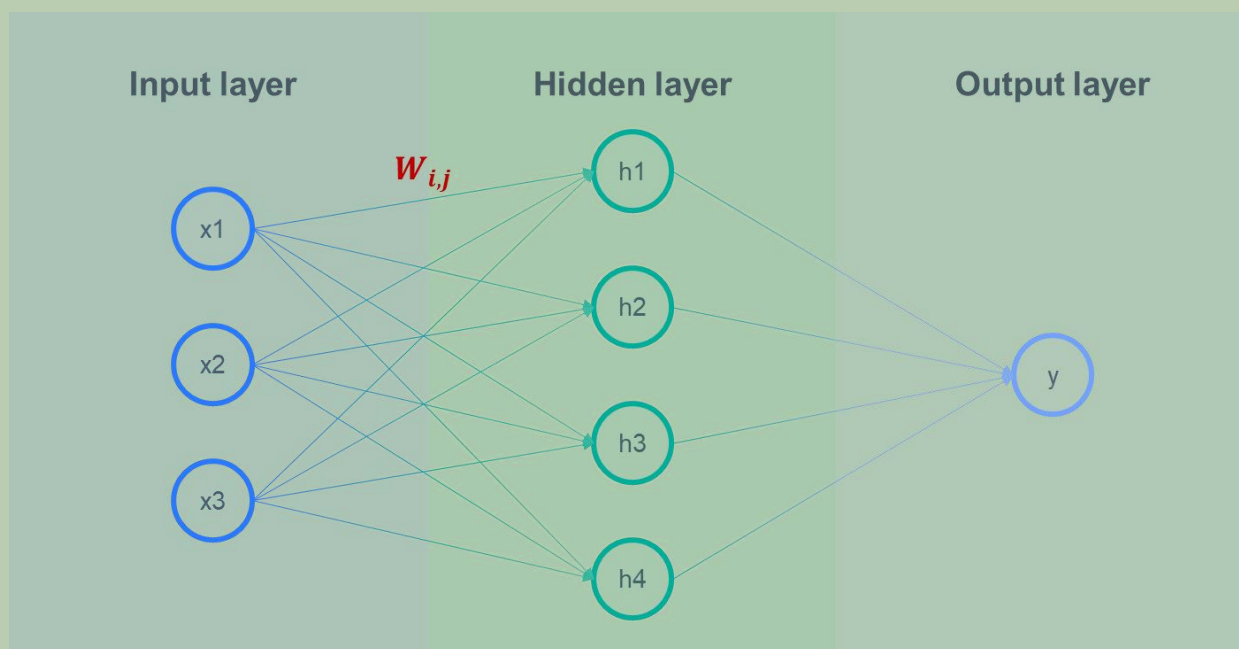
神經網路 **neural network**

神經網路 Neural Network

輸入層

隱藏層

輸出層



需要將資料轉變成輸入特徵

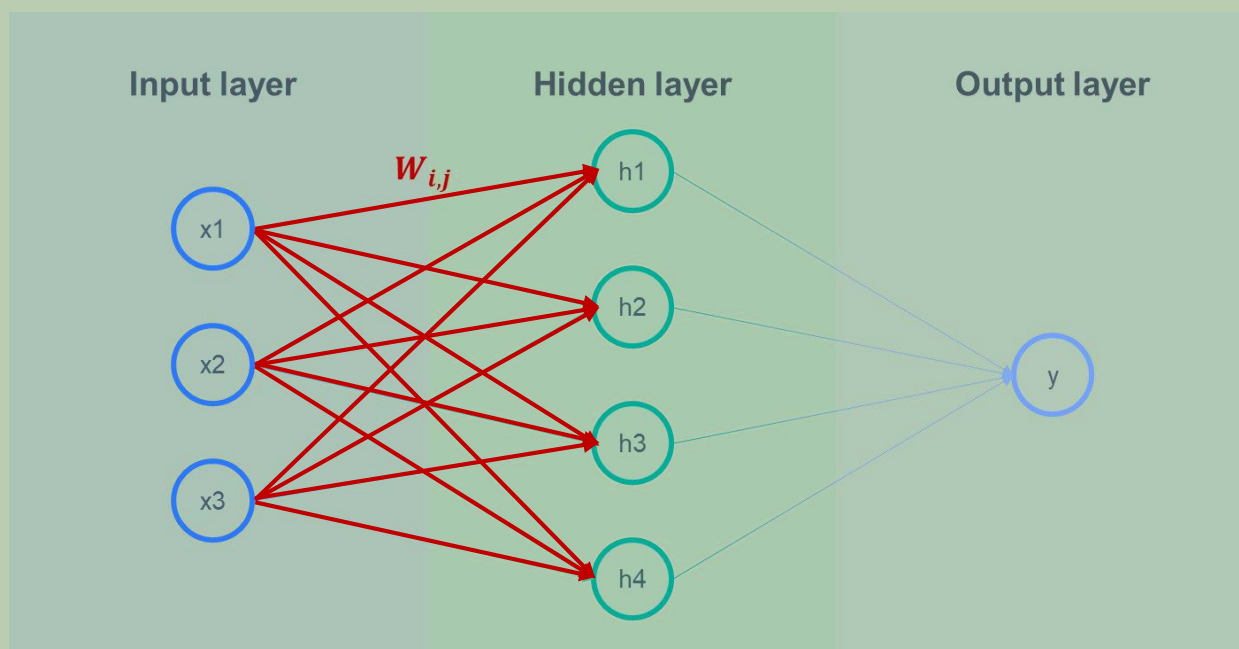
如左圖所示
輸入特徵為 x_1 , x_2 , x_3

神經網路 Neural Network

輸入層

隱藏層

輸出層



輸入特徵經過神經元進行運算

如左圖所示

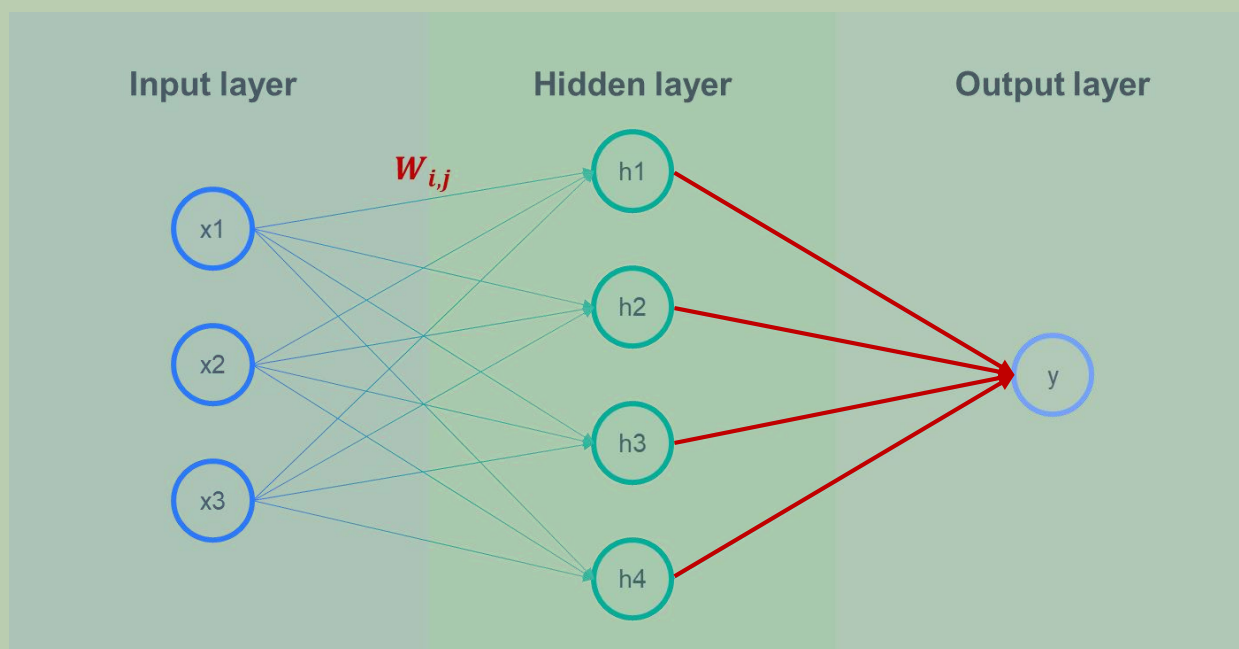
$$\begin{aligned} h_1 &= x_1 W_{11} + x_2 W_{21} + x_3 W_{31} \\ &\vdots \\ h_4 &= x_1 W_{14} + x_2 W_{24} + x_3 W_{34} \end{aligned}$$

神經網路 Neural Network

輸入層

隱藏層

輸出層



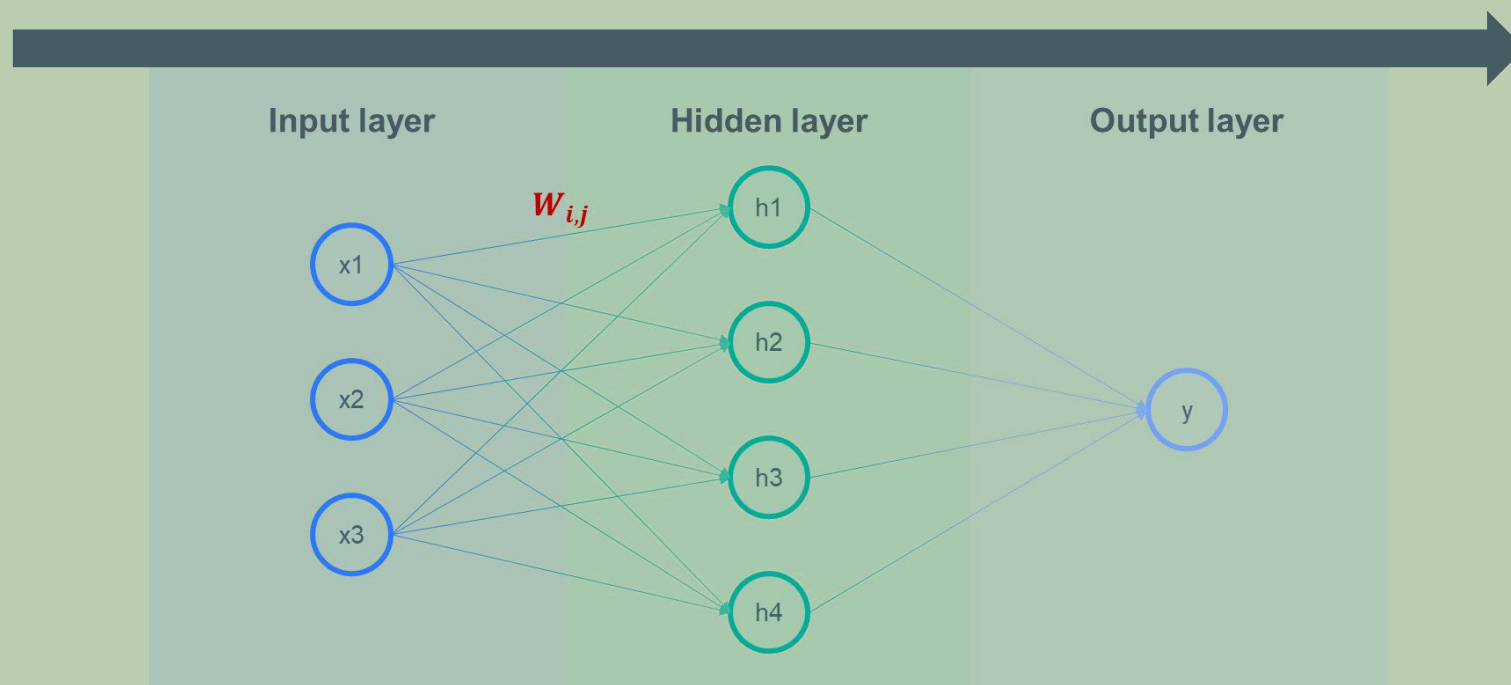
把隱藏層神經元的輸出，進行一個彙整

如左圖所示

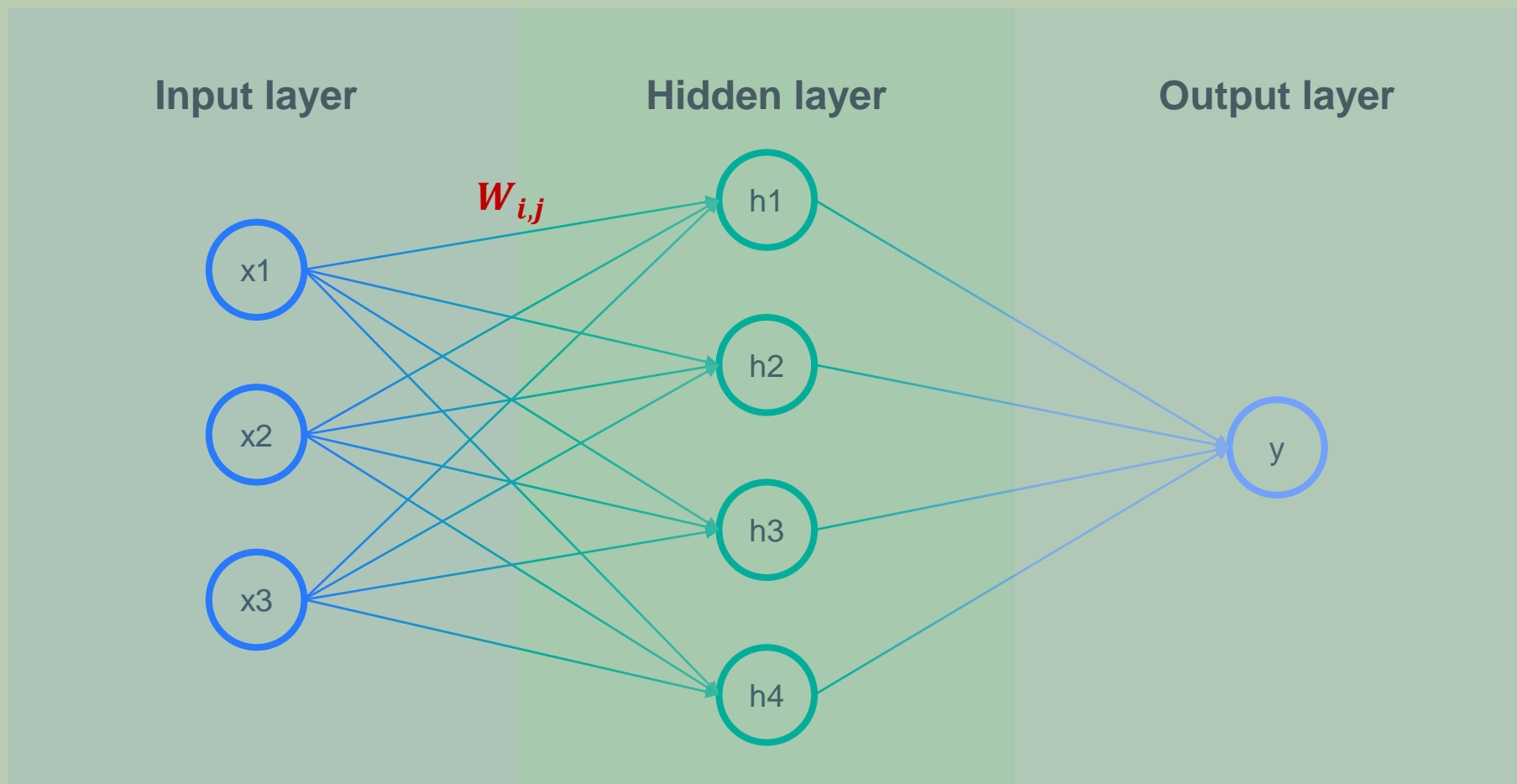
$$y = h1 + h2 + h3 + h4$$

神經網路 Neural Network

- 由前往後，一層層傳遞輸入值，並經過權重計算，產生輸出值。
- 學術上，稱之為「前向傳播法」(forward-propagation)

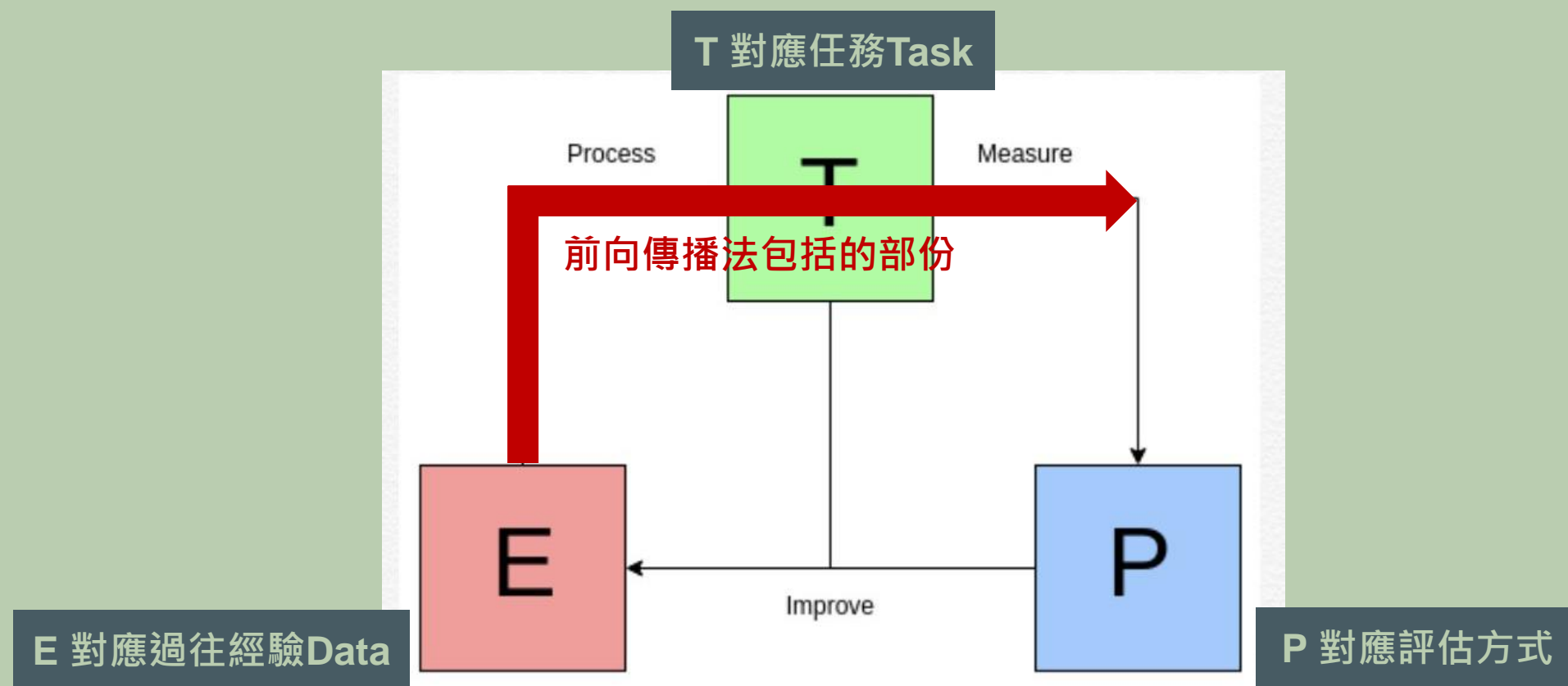


神經網路 Neural Network



訓練迴圈

訓練迴圈



資料來源: Kirk Borne's Twiter

<https://twitter.com/kirkdborne/status/1079062765778669571?lang=zh-Hant>

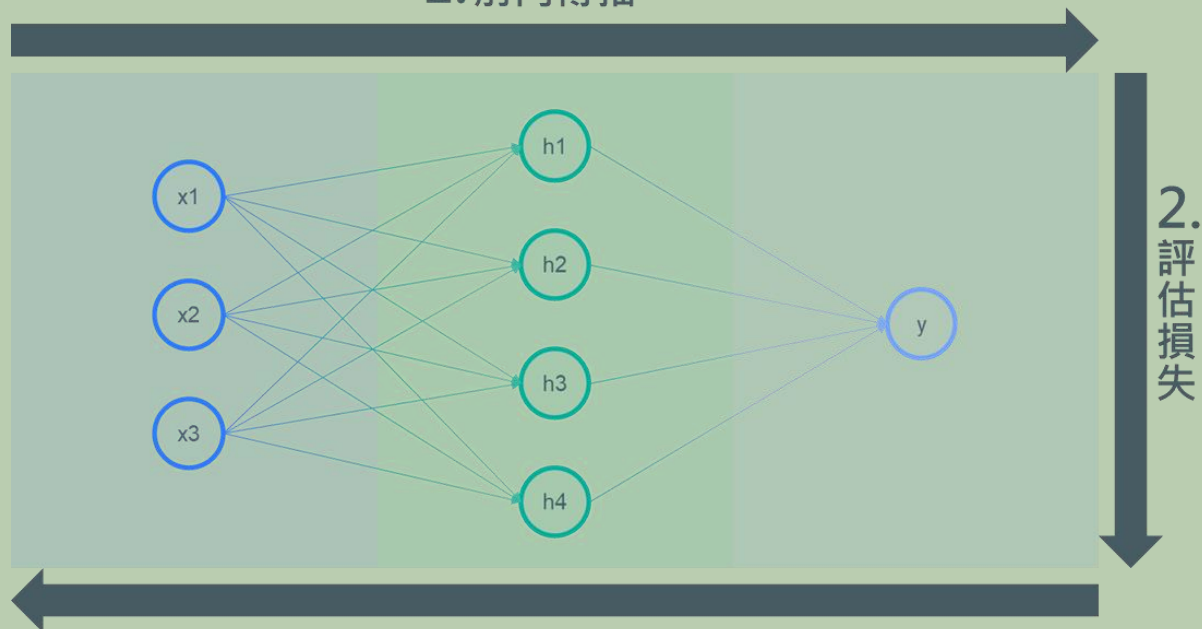
訓練迴圈

前向傳播

評估損失

反向傳播

1. 前向傳播



2. 評估損失

描述過程

模型透過前向傳播法，來對輸入的資料特徵做運算，從而得到它的預測值。

3. 反向傳播

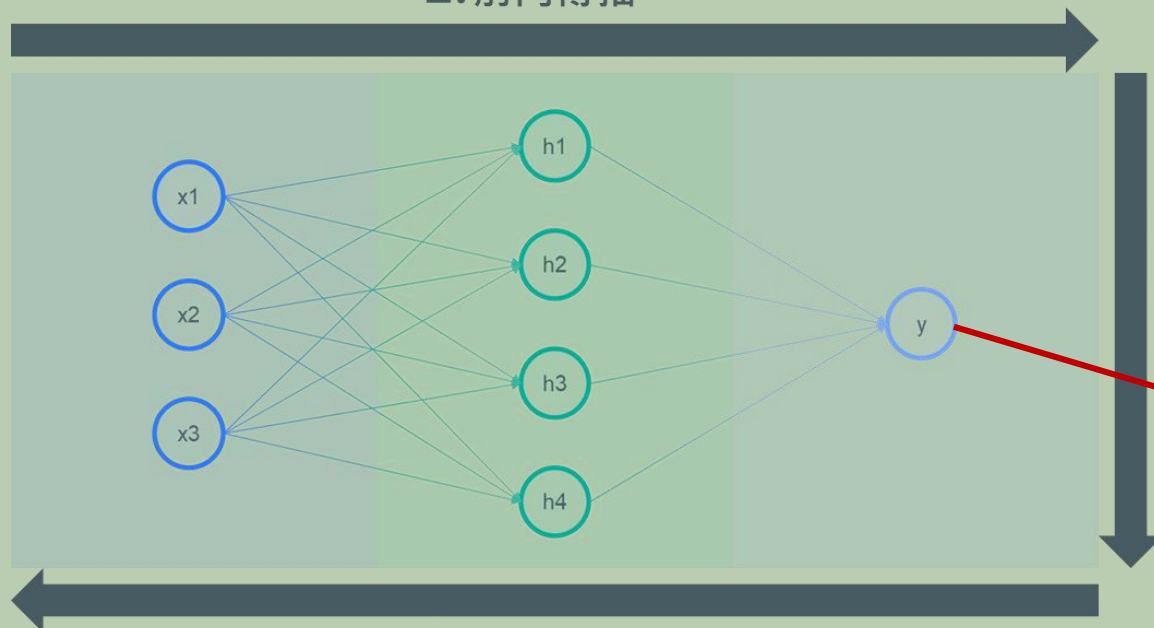
訓練迴圈

前向傳播

評估損失

反向傳播

1. 前向傳播



描述過程

將得到的預測結果 y ，拿去與真實目標值做比較，從而計算出兩者之間的差距(損失)

y

損失函數

兩者間的差距

訓練迴圈

前向傳播

評估損失

反向傳播

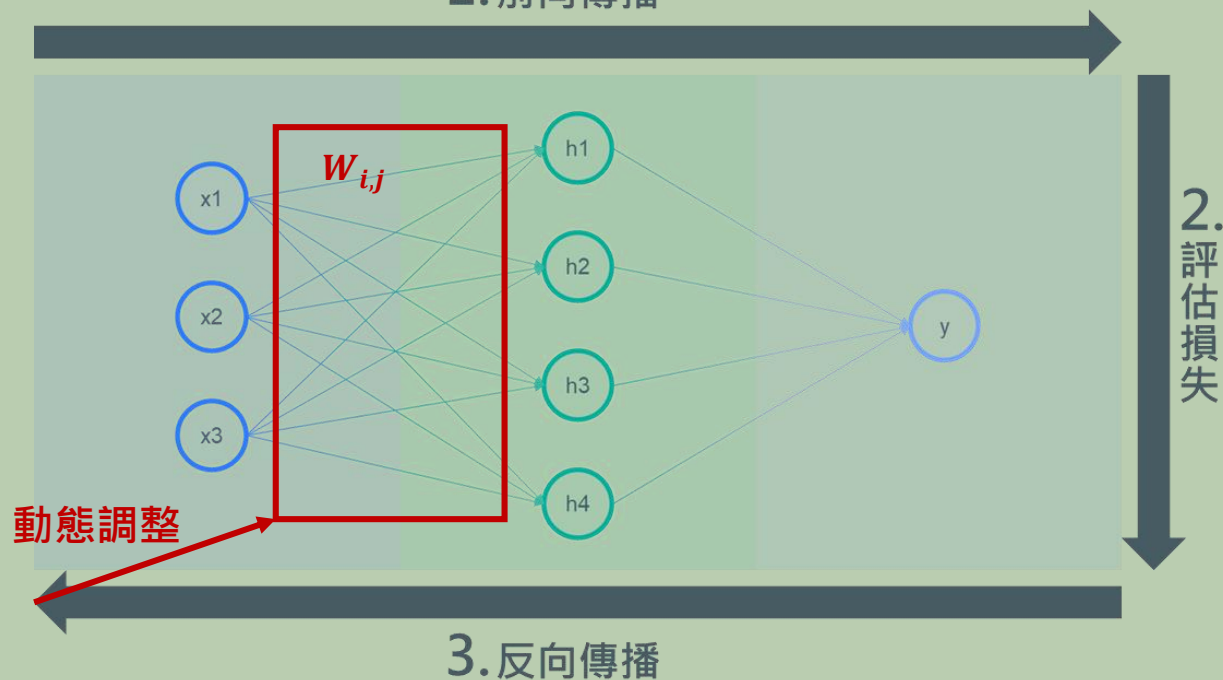
1. 前向傳播

2. 評估損失

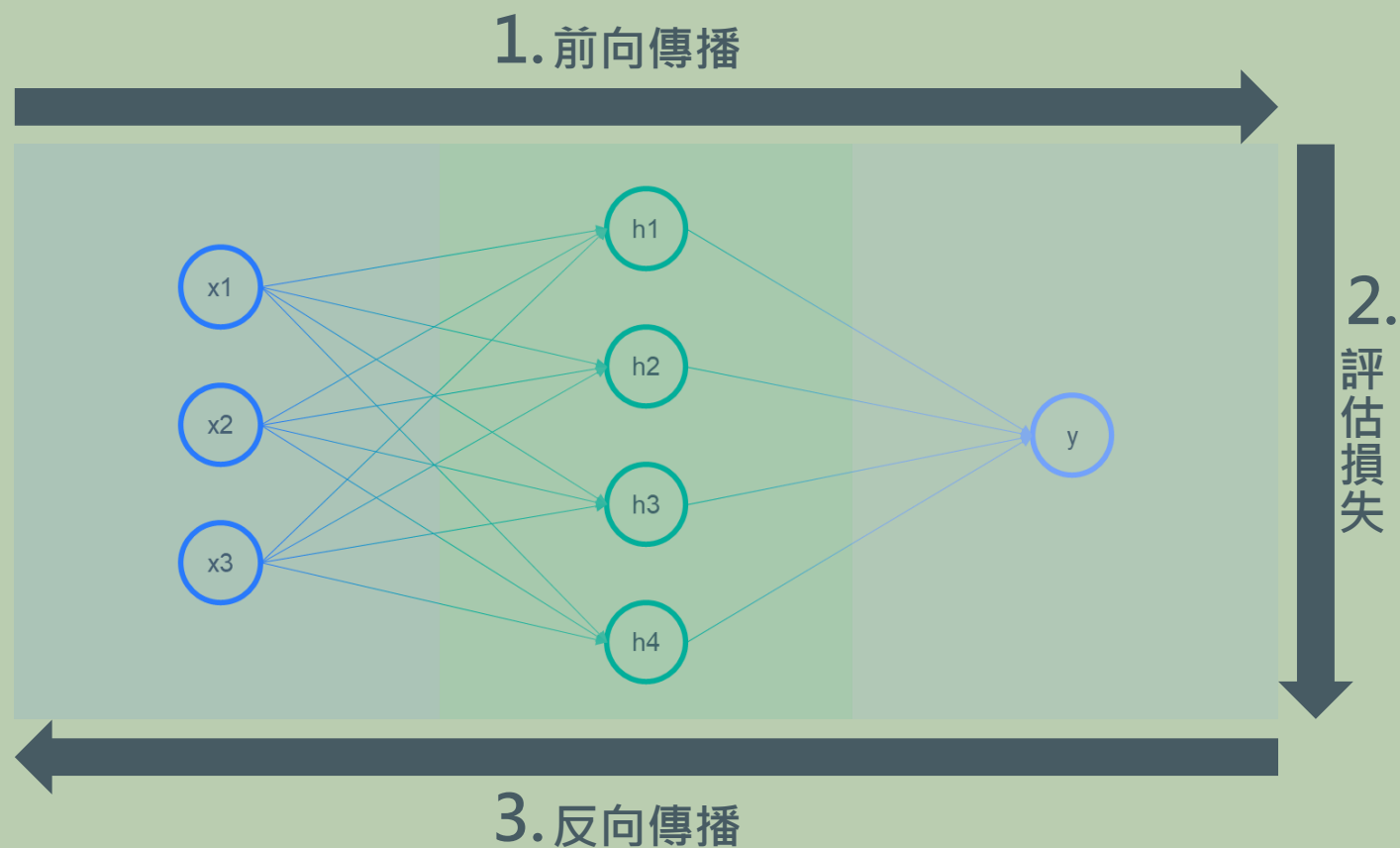
3. 反向傳播

描述過程

根據損失函數計算出的值，來判斷模型有沒有學到東西，從而動態調整模型內部的特徵權重。



訓練迴圈



CNN 卷積神經網路

卷積神經網路

- Convolutional Neural Network，簡稱 CNN。
- CNN 基本上是在深度學習裡必教的一個模型，主要用在視覺領域。
- 它會模仿人類大腦的認知方式
- 舉例來說，辨識一個圖像時，會先注意到顏色鮮明的特徵
- 而後將它們組裝成眼睛、鼻子等形狀
- 這個抽象化的過程，就是 CNN 演算法建立模型的方式。

卷積神經網路

- 在正式介紹 CNN 前
- 要先簡單的介紹，我們是如何將影像變成機器能夠學習的狀態。
- 可以分成以下階段
- 1. 影像 -> 影像特徵 (讓電腦看的懂)
- 2. 卷積 (把影像特徵變小)
- 3. 池化 (挑選合適的特徵，一般越大越好)
- 4. flatten後，接續一個隱藏層和輸出層 (跟神經網路一樣)

影像前處理

- 要先將影像處理成電腦看的懂的形式。

20	225	171	34
87	71	224	75
130	95	165	81
60	40	187	0

原始圖片



20	225	201	34
87	71	174	75
30	95	36	21
60	40	187	0

處理後的矩陣(圖像特徵)

卷積運算

輸入資料

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

圖像特徵

滑動窗口

進行運算

產生新矩陣

1	1	1
1	1	1
1	1	1

卷積核

卷積運算

輸入資料

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

圖像特徵

滑動窗口

進行運算

產生新矩陣

1	1	1
1	1	1
1	1	1

卷積核

卷積運算

輸入資料

滑動窗口

進行運算

產生新矩陣

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

圖像特徵

1	1	1
1	1	1
1	1	1

卷積核

$$\begin{aligned} &1 * 1 + 2 * 1 + 3 * 1 \\ &+ \\ &5 * 1 + 6 * 1 + 7 * 1 \\ &+ \\ &9 * 1 + 10 * 1 + 11 * 1 = 54 \end{aligned}$$



54

卷積結果

卷積運算

輸入資料

滑動窗口

進行運算

產生新矩陣

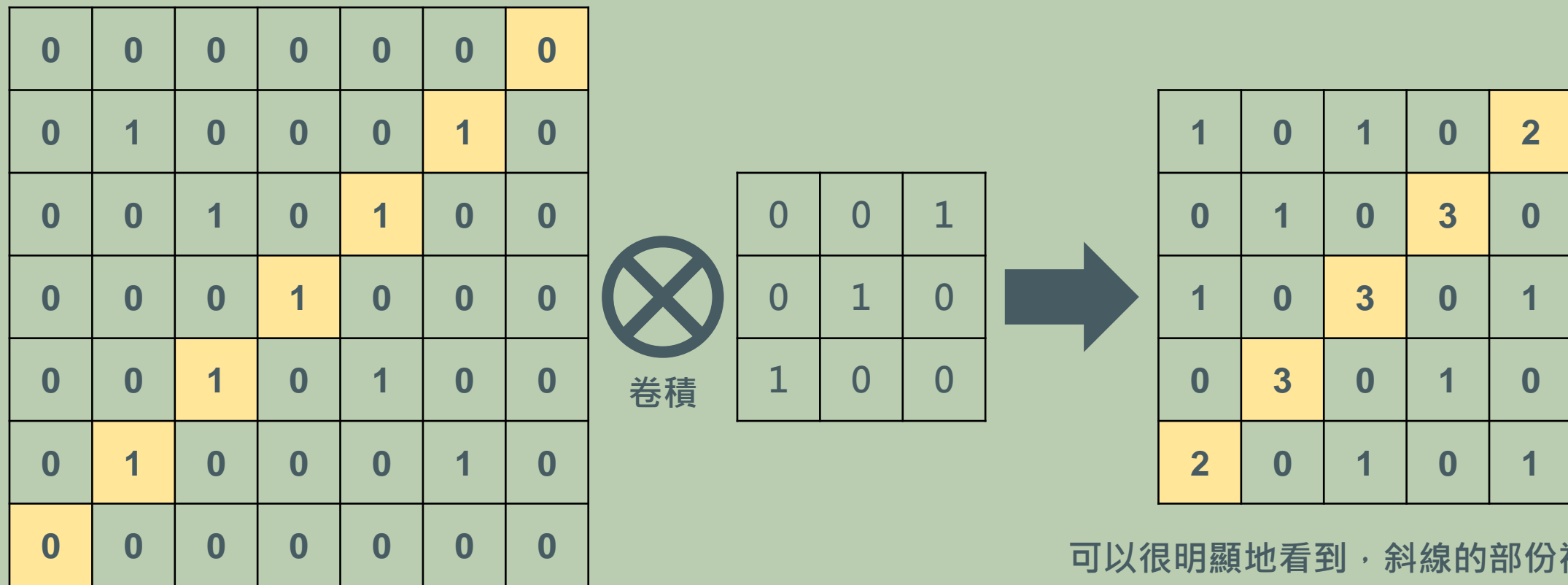
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

圖像特徵



新矩陣

卷積核的作用



池化運算

- 假設這個地方採用2X2的池化運算。

1	0	1	0	2
0	1	0	3	0
1	0	3	0	1
0	3	0	1	0
2	0	1	0	1

特徵圖

池化

用顏色表示
數字越小顏色越深

1	3	2
3	3	1
2	1	1

圖被壓縮，且保留了更強烈的斜線樣式。

平坦層

- 將資料進行攤平 Flatten

1	3	2
3	3	1
2	1	1

經過池化層後的圖片特徵



全連接層

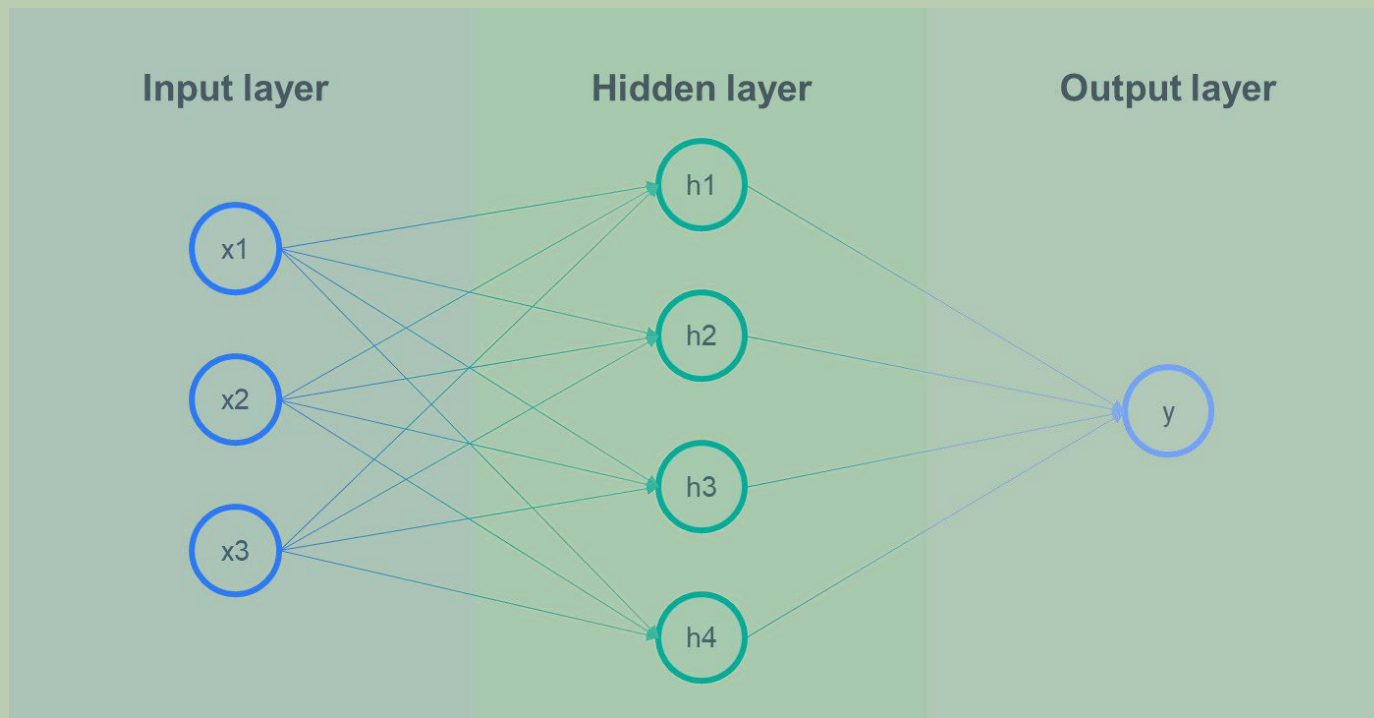
- Full Connection / 全連接層

1
3
2
3
3
1
2
1
1

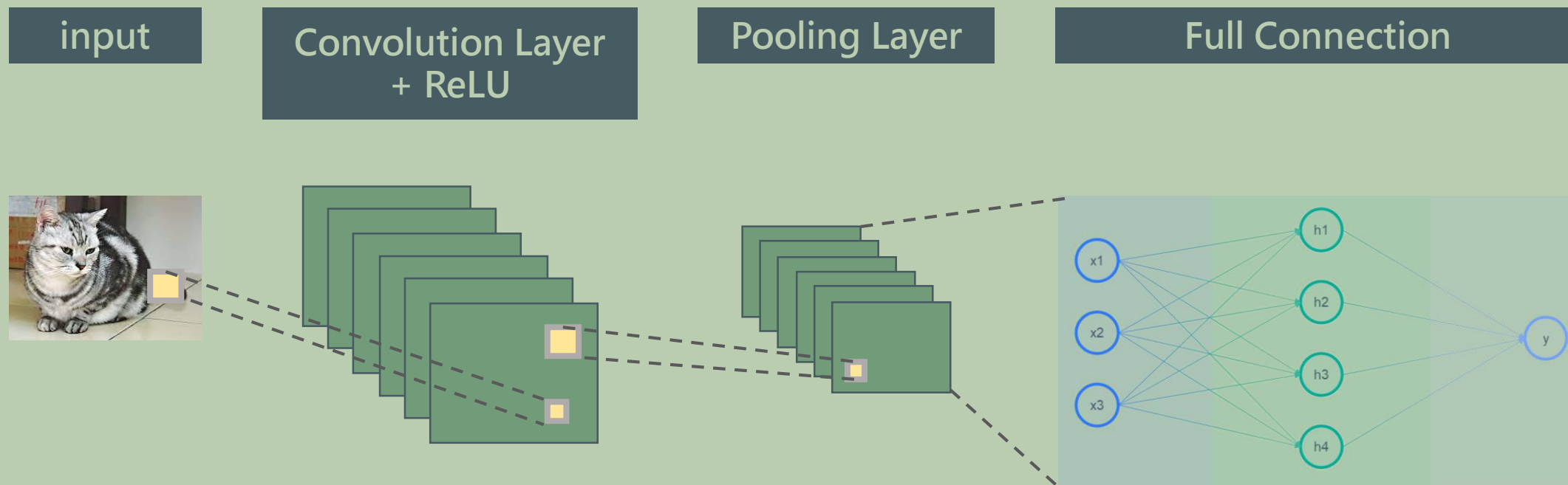
經過平坦層後的
圖像特徵



輸入至神經網路
(全連接層)



完整的模型架構



一維卷積

- 因為金融的資料多為一維特徵。

5	4	3	1	8	7	5
---	---	---	---	---	---	---

圖像特徵

1	1	1
---	---	---

卷積核

--	--	--	--	--

新矩陣

Thanks for your listening