深度學習簡介

機器學習百日馬拉松

Agenda

- 深度學習的基礎知識知識點
- 深度學習的核心概念
- 一些矩陣與向量的基礎知識介紹
- 深度學習的常見問題 (FAQ)

我們先認識一下
深度學習的基礎知識點

基礎知識點

ANNs
人工神經網路

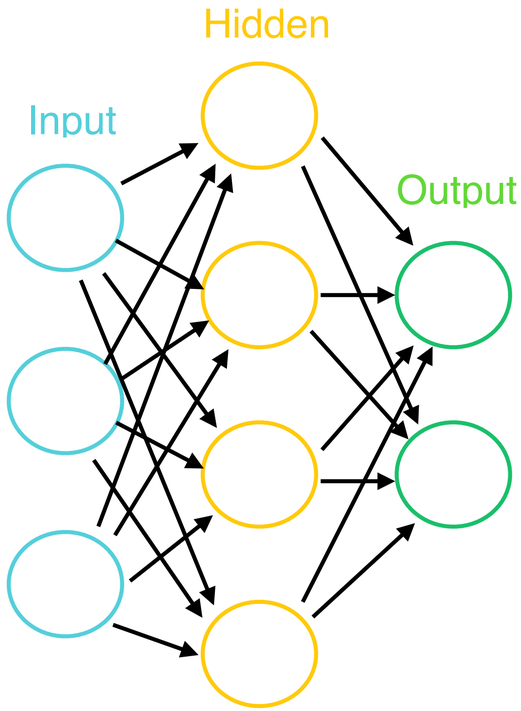
Perception
感知器

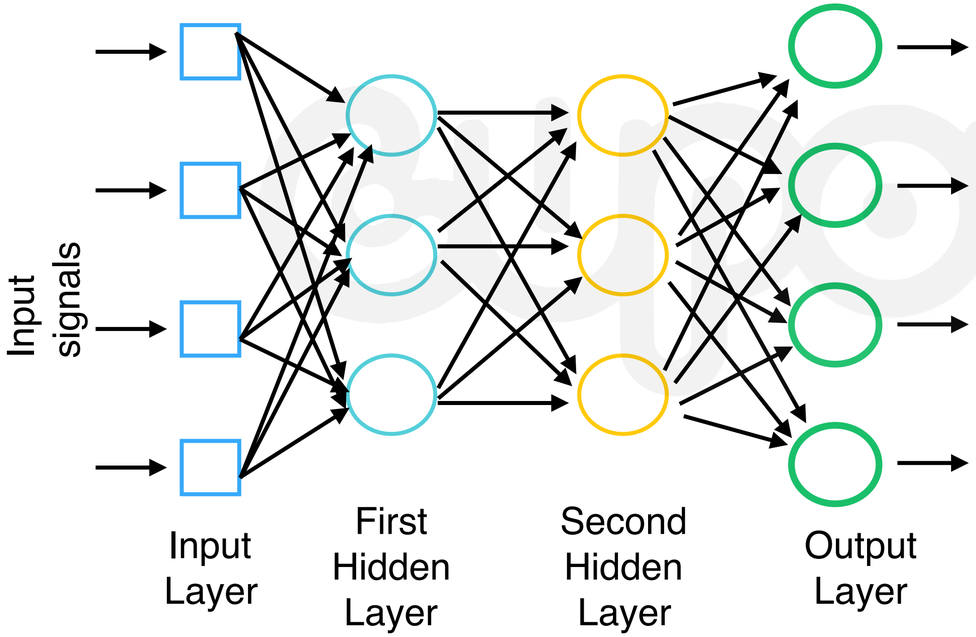
Feedforward
Neural Network
前饋神經網路

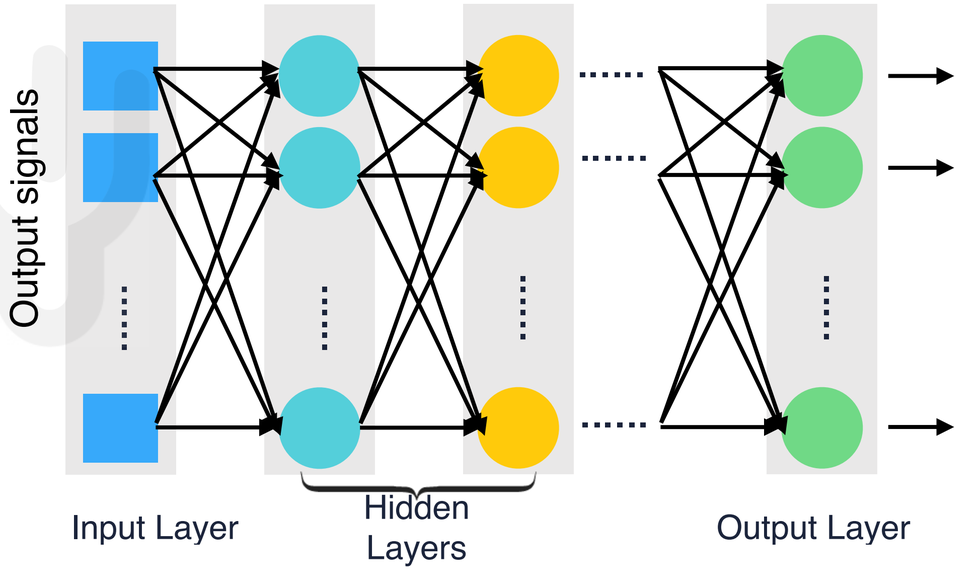
CNN
卷積神經網路

RNN
遞歸神經網路

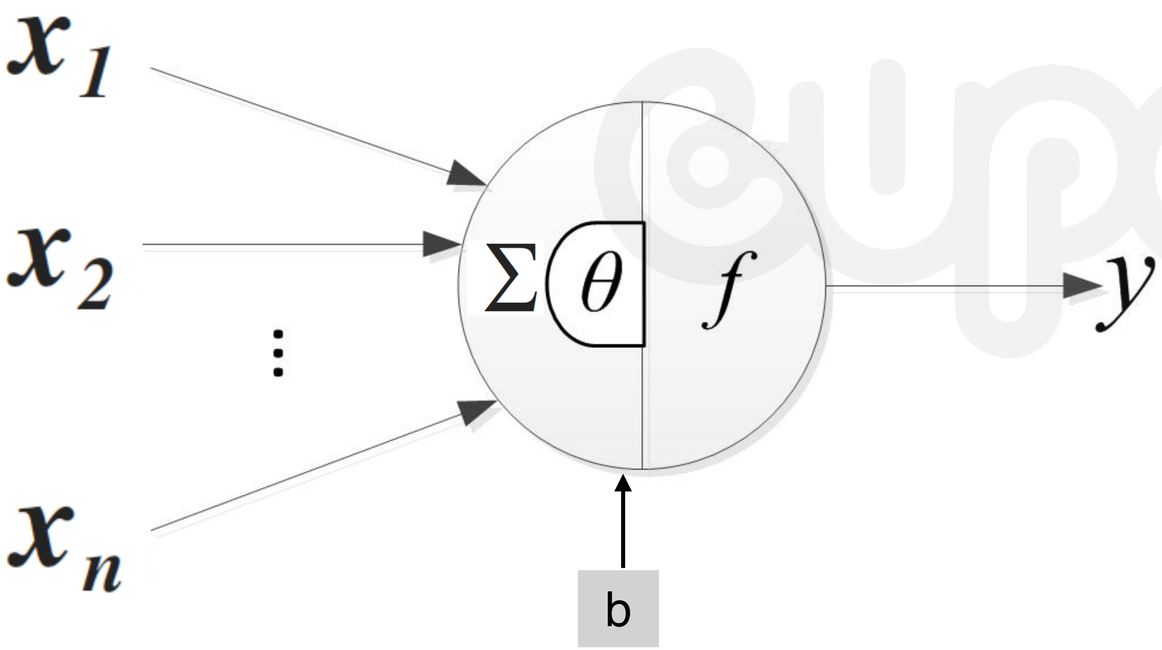
ANNs：人工神經網路


Hidden
Input
Output


Input signals
Input Layer
First Hidden Layer
Second Hidden Layer
Output Layer

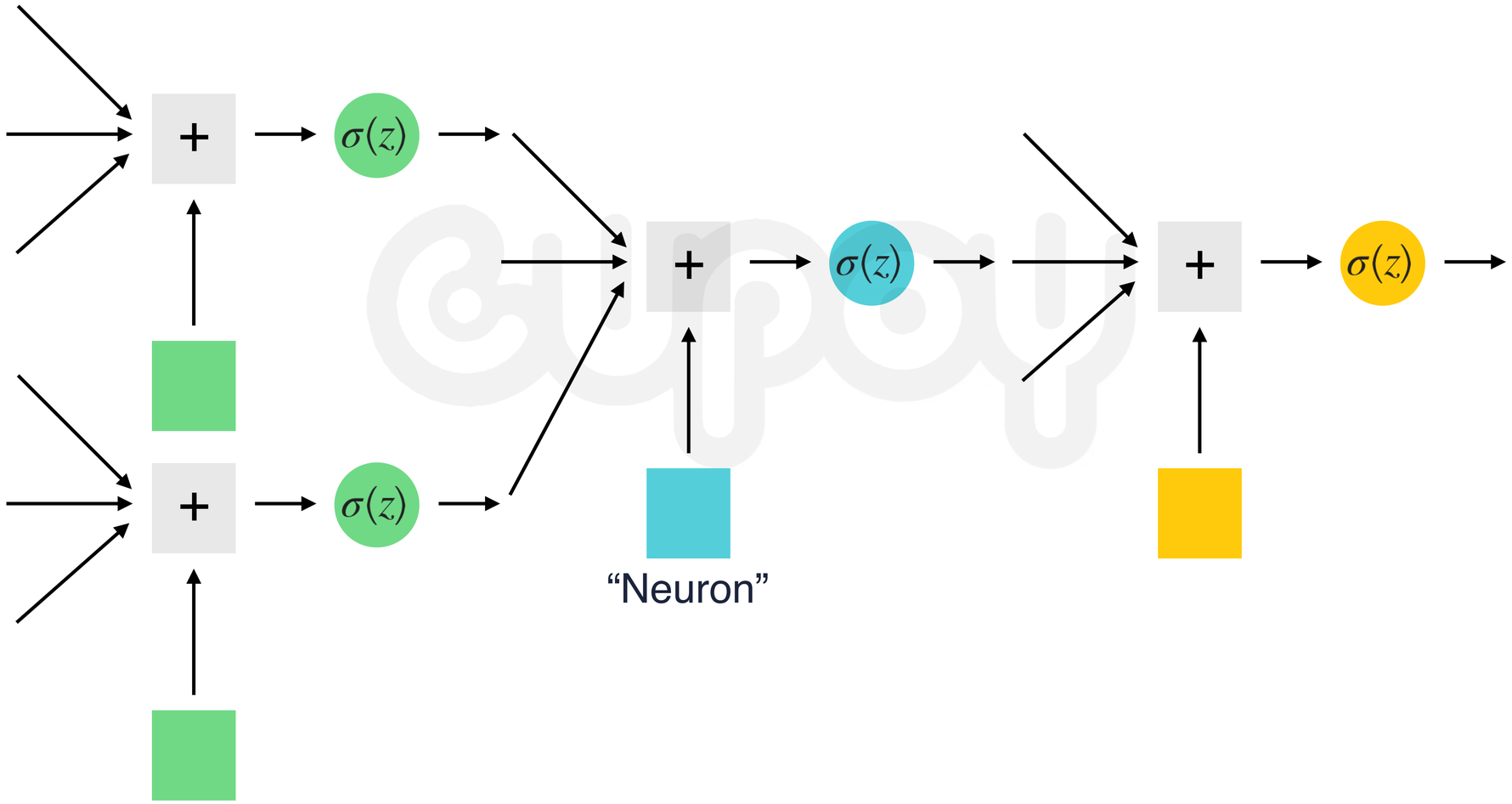

Output signals
Input Layer
Hidden Layers
Output Layer

Perceptron：感知器（Neuron)

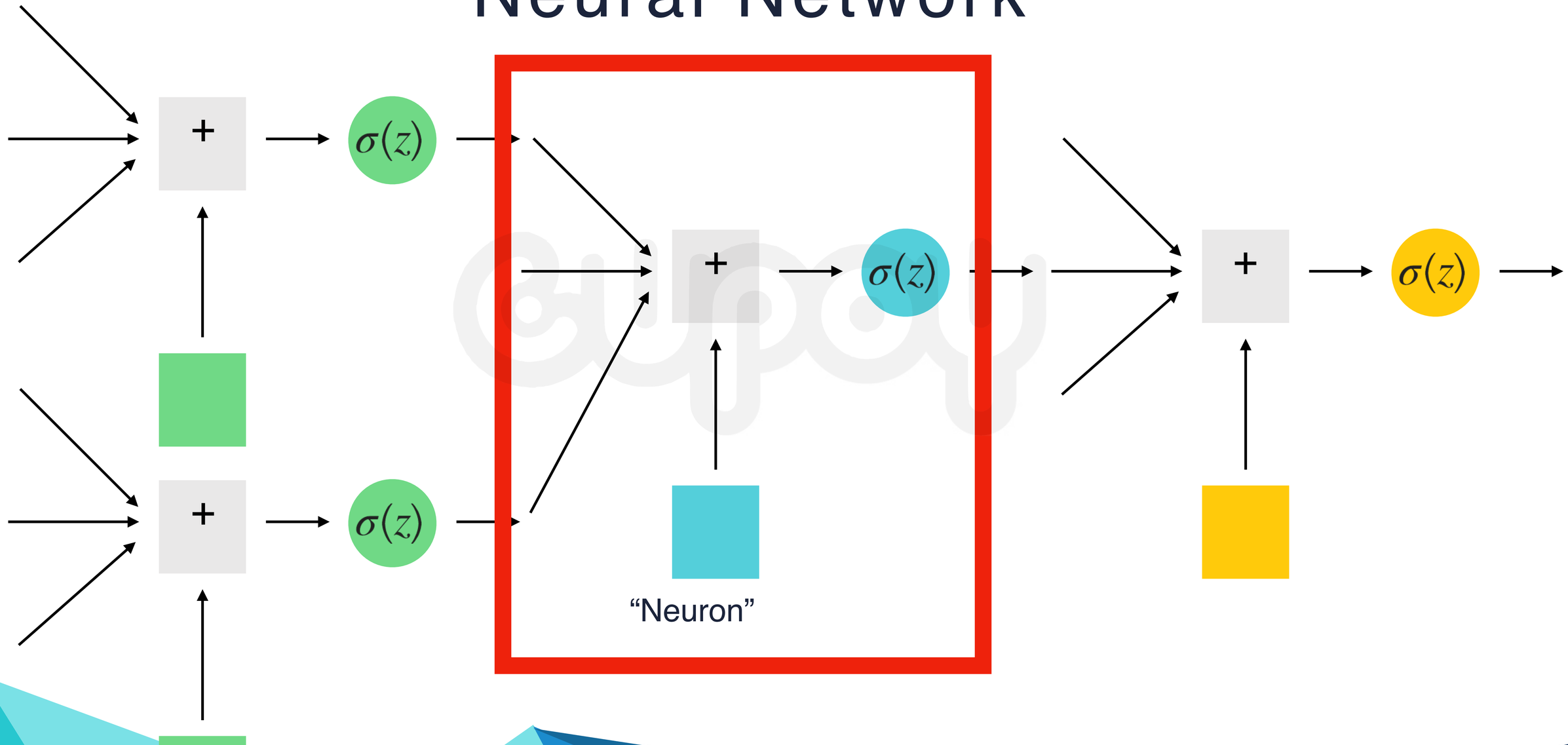


Neuron	作用
輸入層 (x向量)	接收輸入信號
加權和(Σ)	加工處理信號
閾值函數(f)	控制輸出
輸出(y)	輸出結果

Neural Network

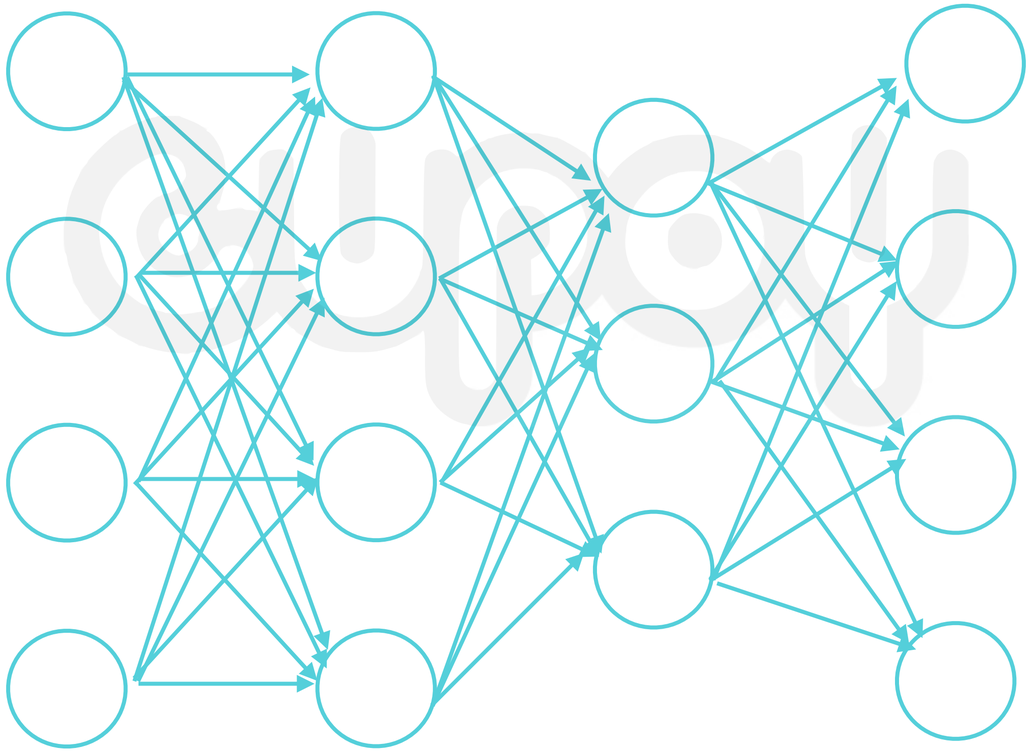


Neural Network

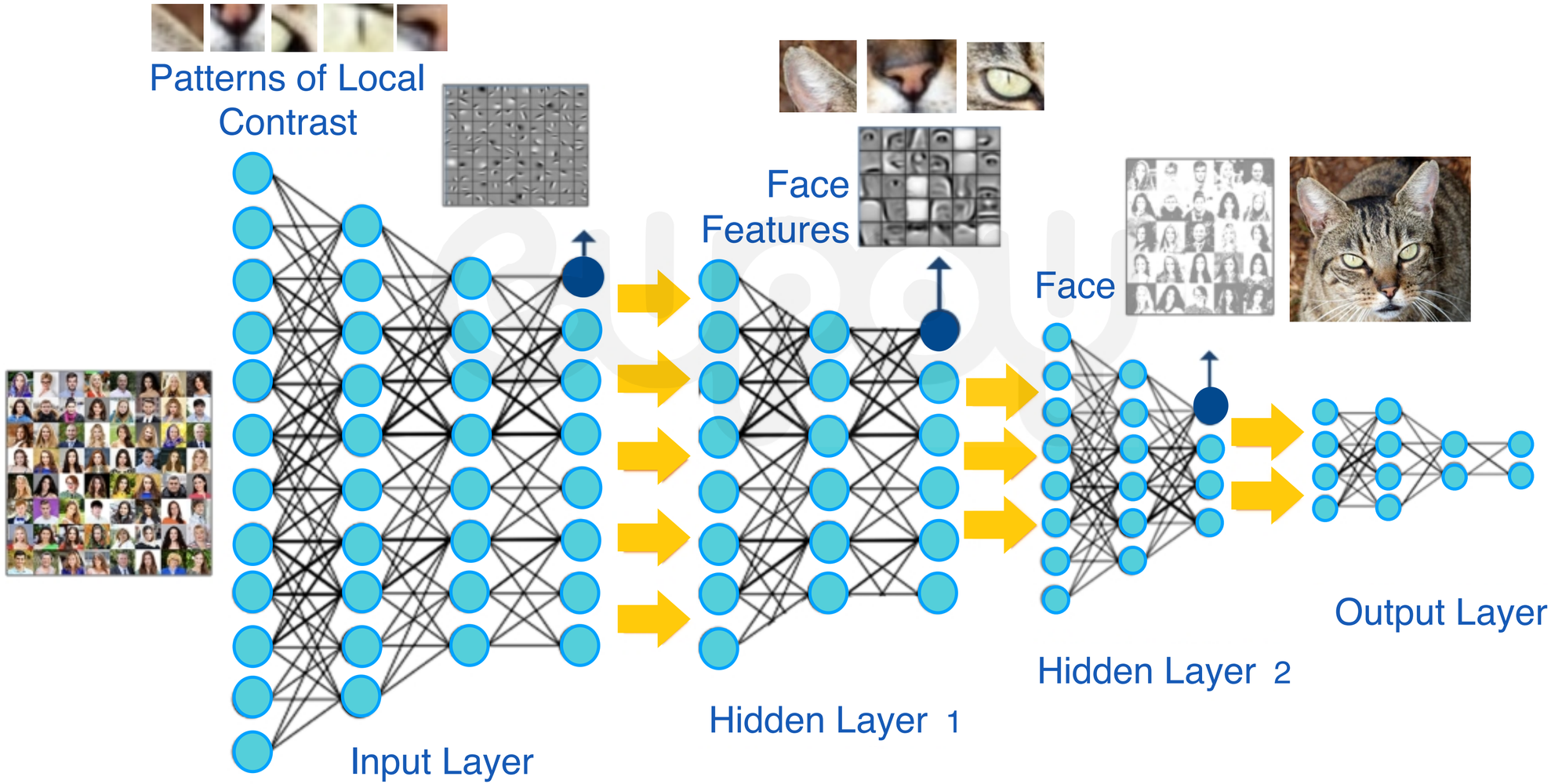


前饋神經網路說明

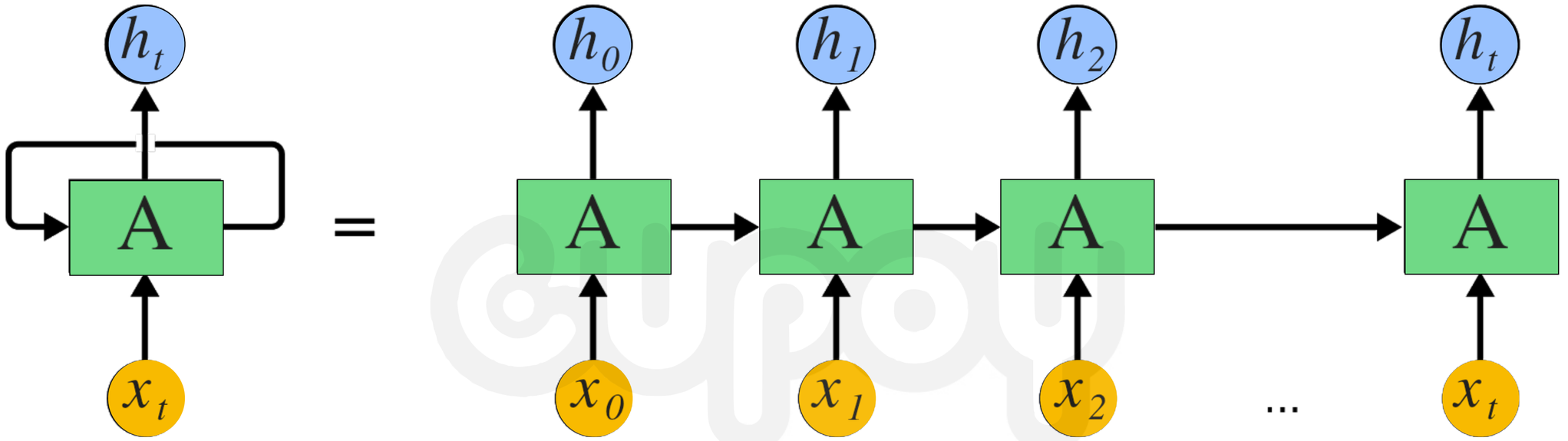
各層資訊固定往前傳遞：稱為前饋神經網路



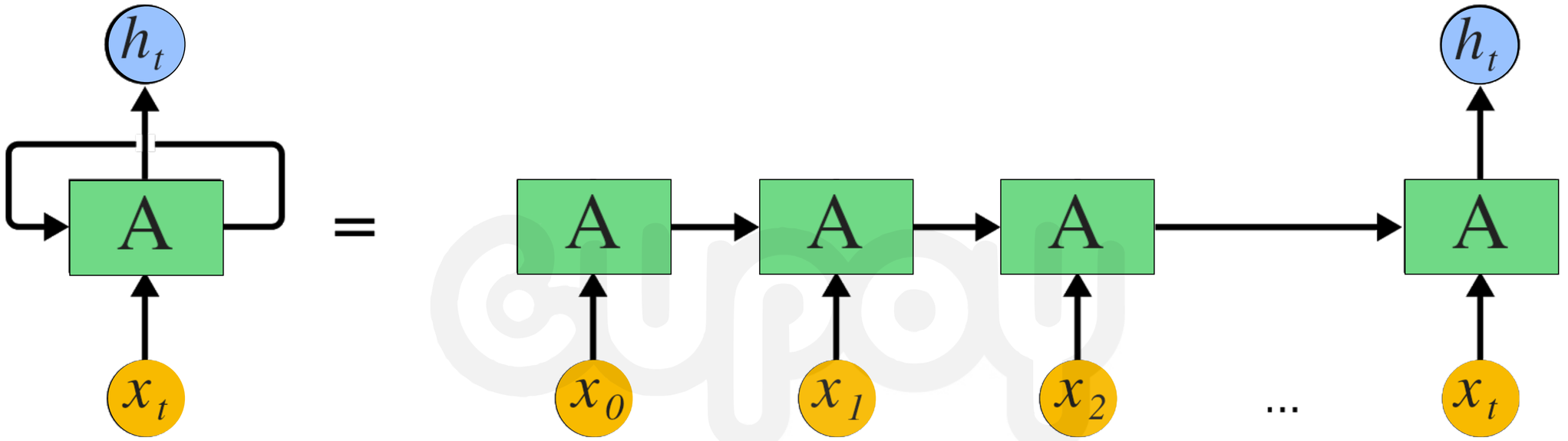
CNN屬於前饋神經網路

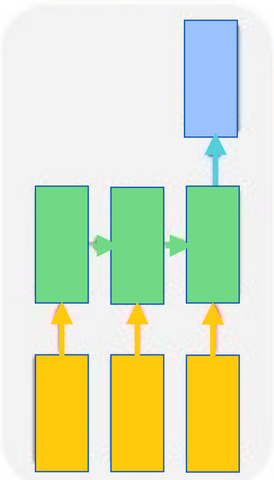


RNN 不是前饋神經網路

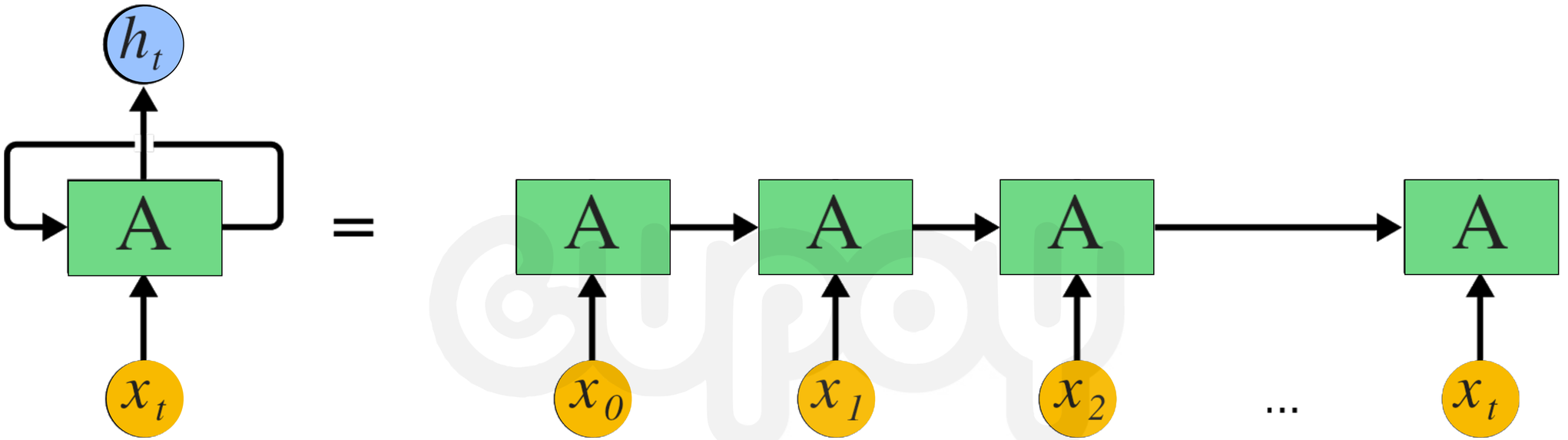


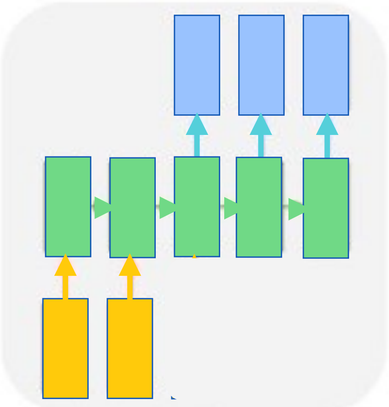
RNN：遞歸神經網路(Case1)



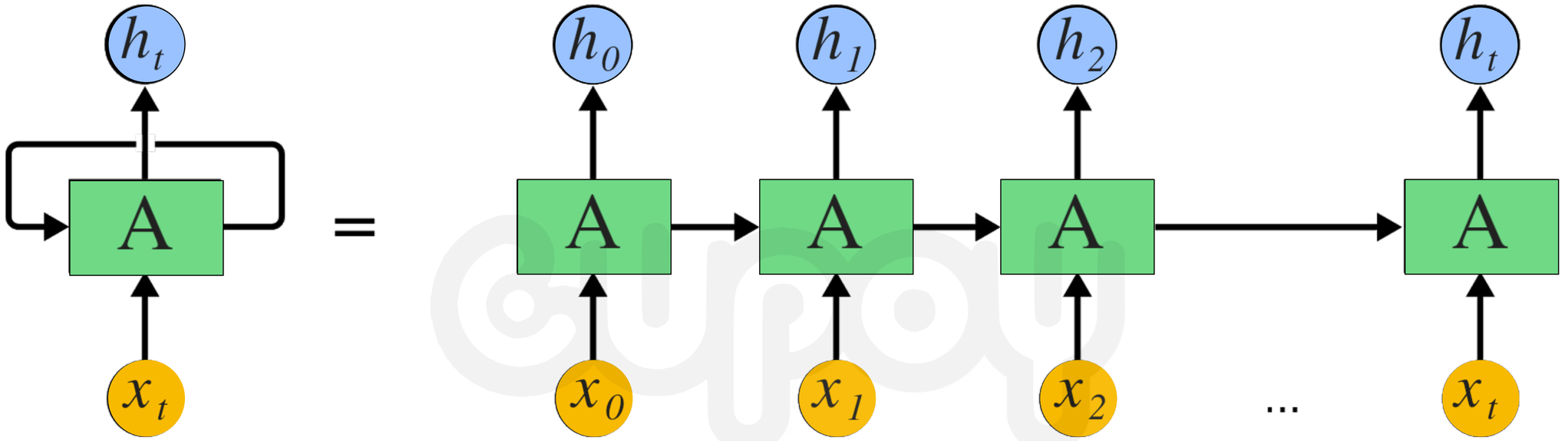


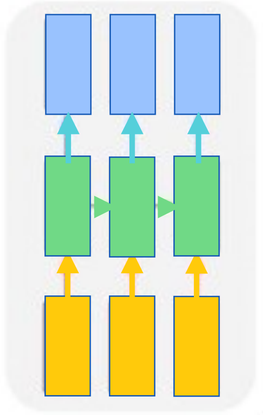
RNN：遞歸神經網路(Case2)





RNN：遞歸神經網路(Case3)





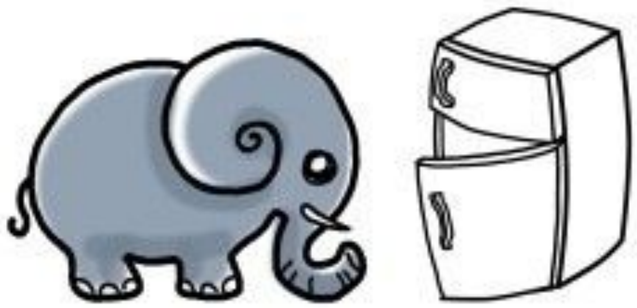
深度學習的核心概念

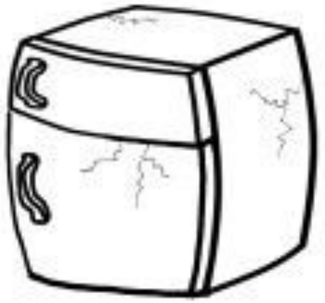
深度學習的三個步驟

01 Define a set of Functions

02 Evaluate the Functions

03 Pick the best Function



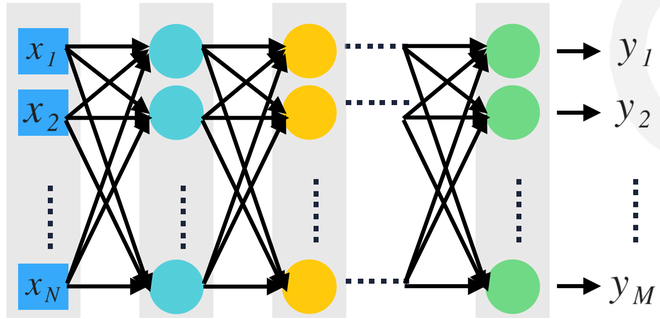


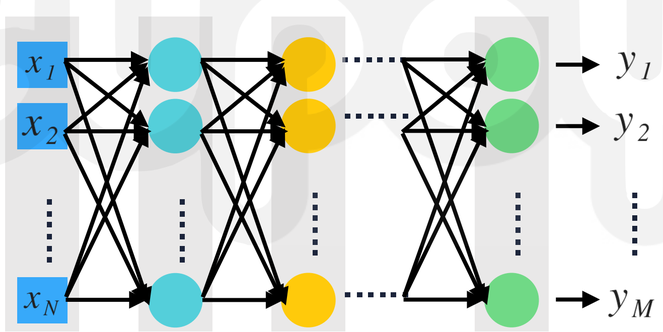
深度學習的三個步驟

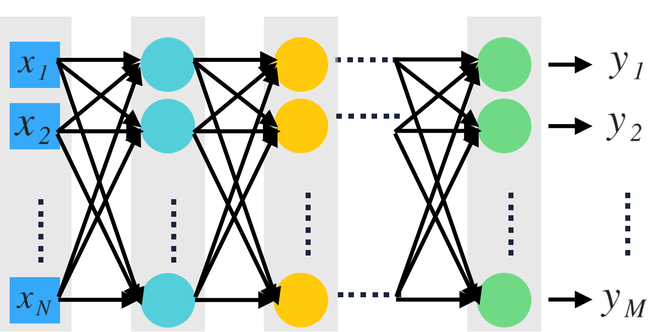
01 Define a set of functions

02 Evaluate functions

03 Pick the best function







$F_1 = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b$

$F_2 = \ldots$

$F_3 = \ldots$

$F_1 = w_1 x_1 + w_2 x_2 + \ldots$

$F_2 = \ldots$

$F_3 = \ldots$

?

0.8

0.3

0.5

Evaluation Function

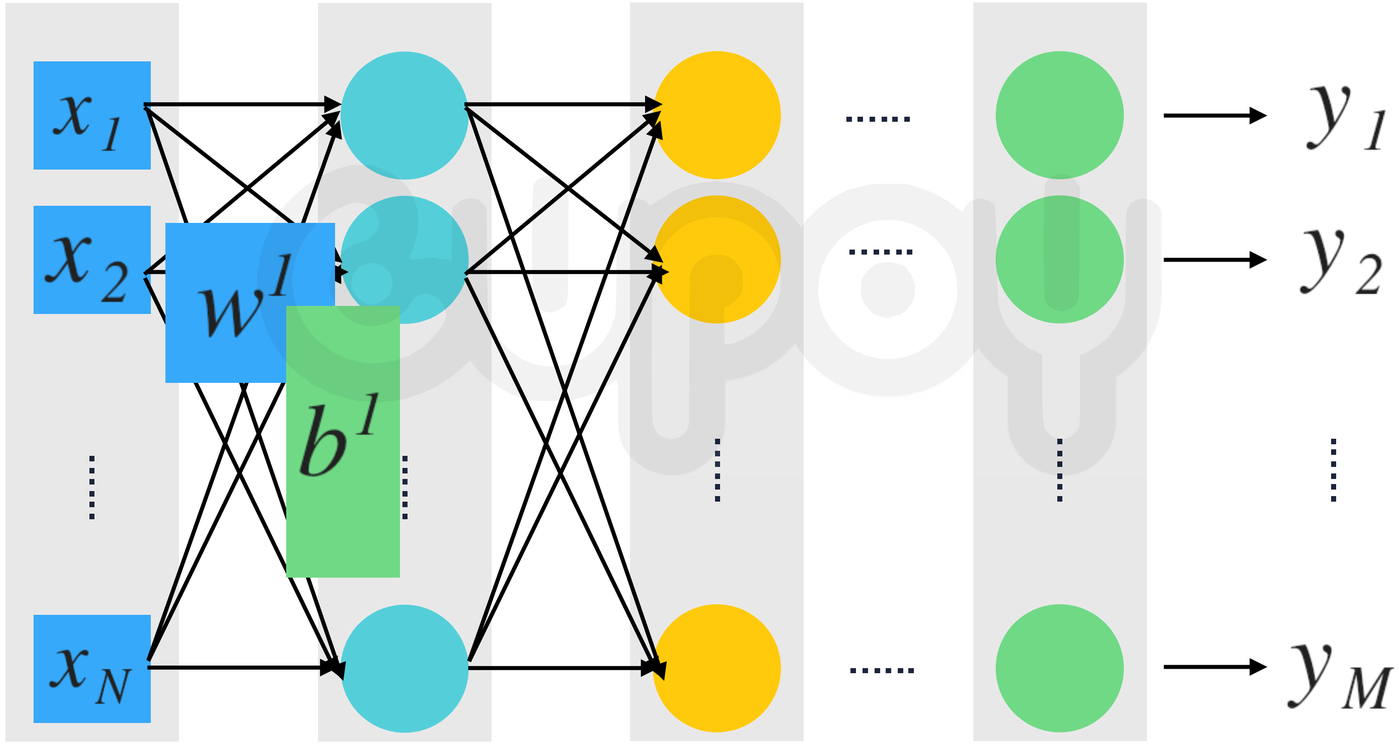
Best Function

$F_1 = w_1 x_1 + w_2 x_2 + \ldots$ → 0.8

$F_2 = \ldots$ → 0.3

$F_3 = \ldots$ → 0.5

Step1 define a set of functions

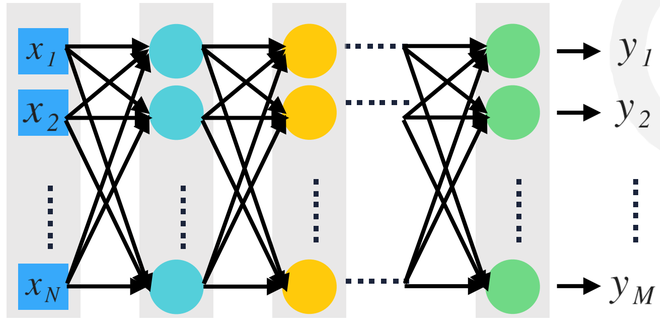


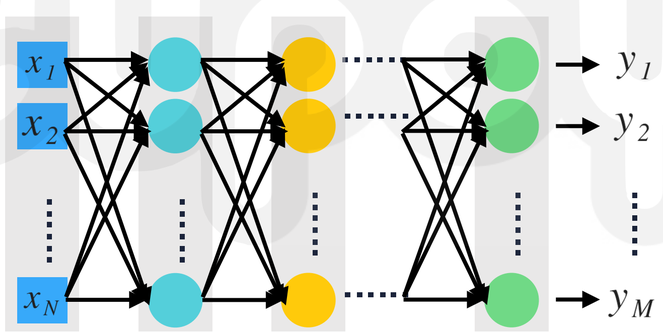
深度學習的三個步驟

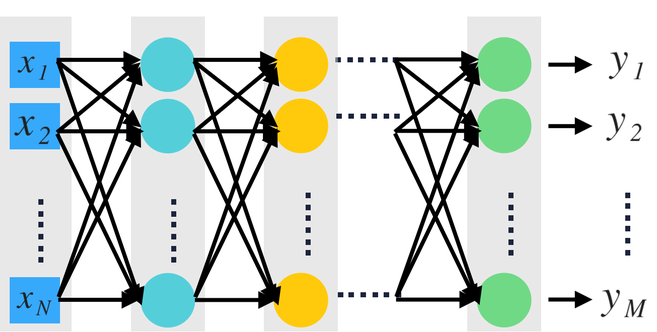
01 Define a set of Functions

02 Evaluate Functions

03 Pick the best Function







$F_1 = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b$

$F_2 = \ldots$

$F_3 = \ldots$

$F_1 = w_1 x_1 + w_2 x_2 + \ldots$

$F_2 = \ldots$

$F_3 = \ldots$

?

0.8

0.3

0.5

Evaluation Function

Best Function

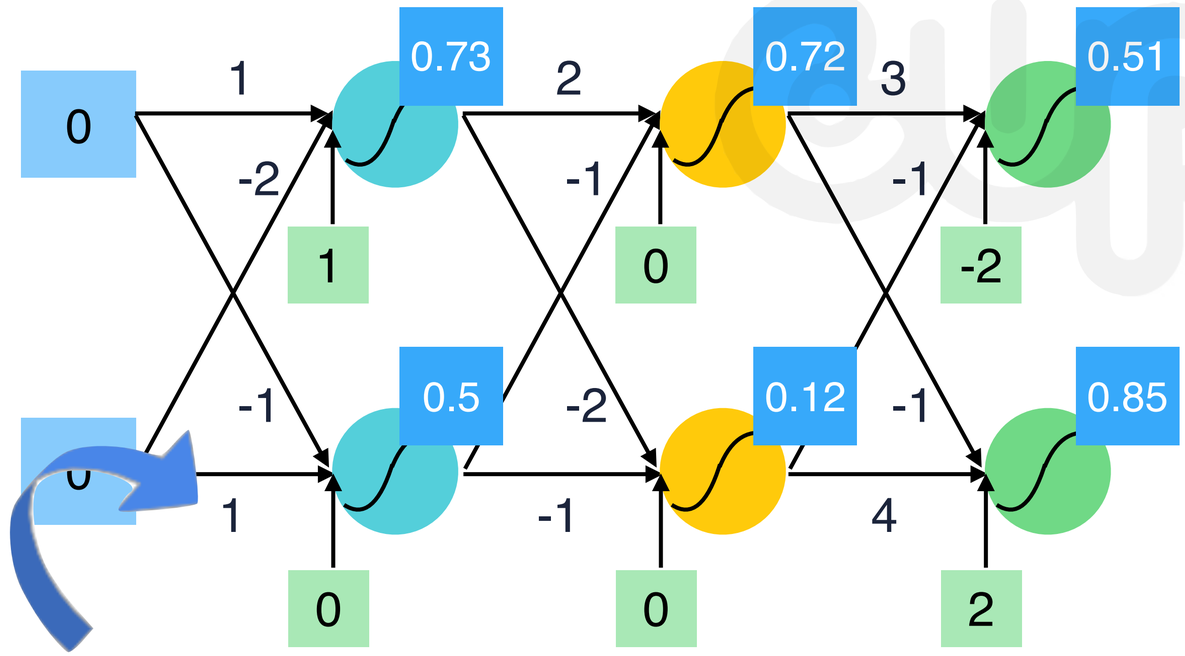
$F_1 = w_1 x_1 + w_2 x_2 + \ldots$ → 0.8

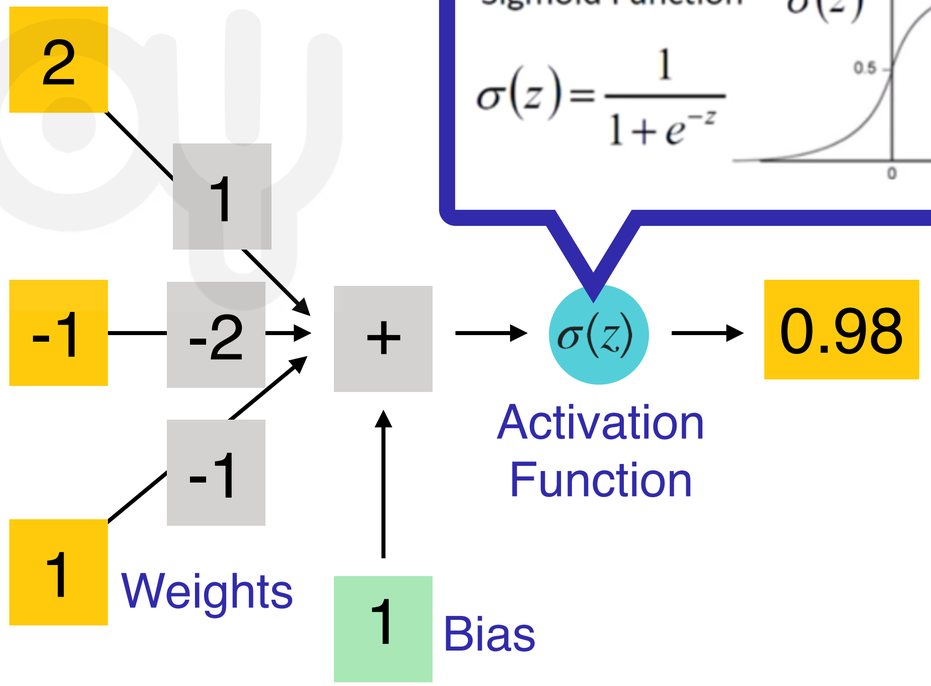
$F_2 = \ldots$ → 0.3

$F_3 = \ldots$ → 0.5

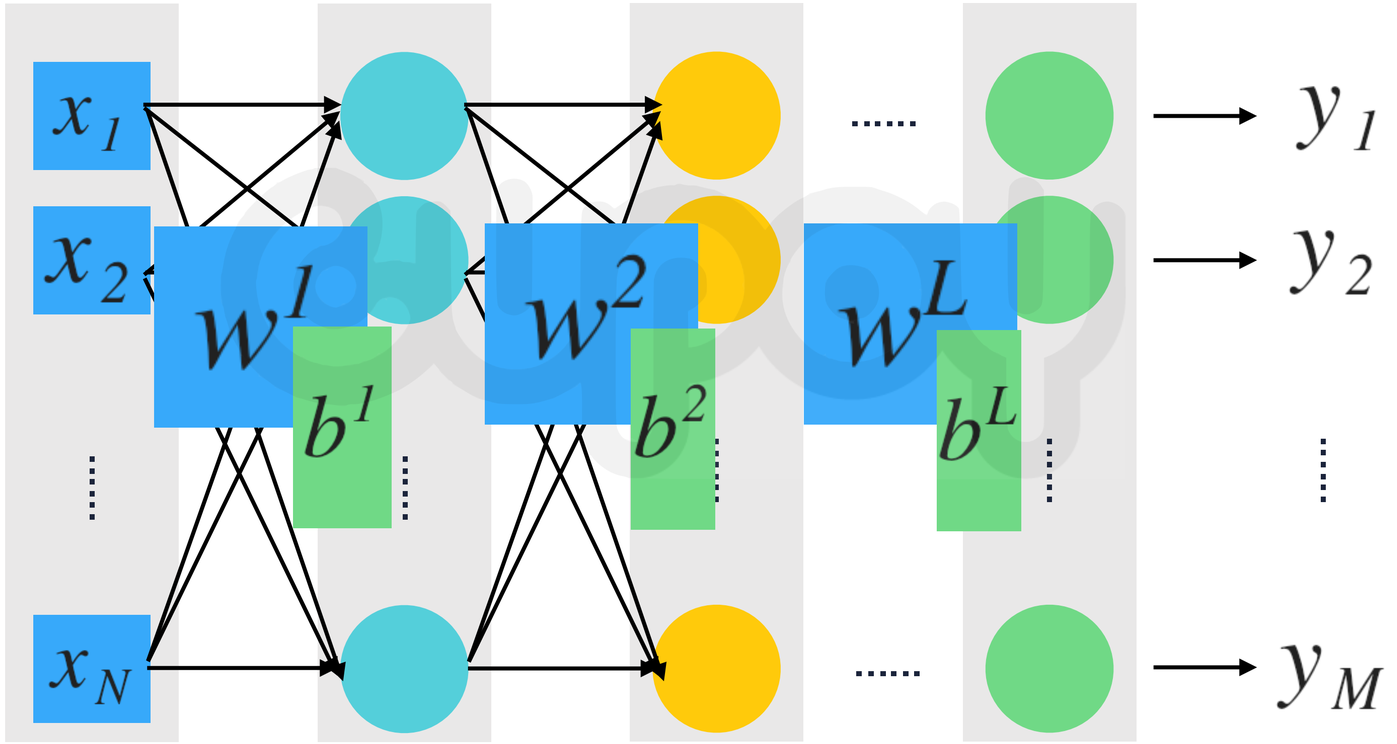
神經元間資料傳遞與運算流程

Fully Connect Feedforward Network

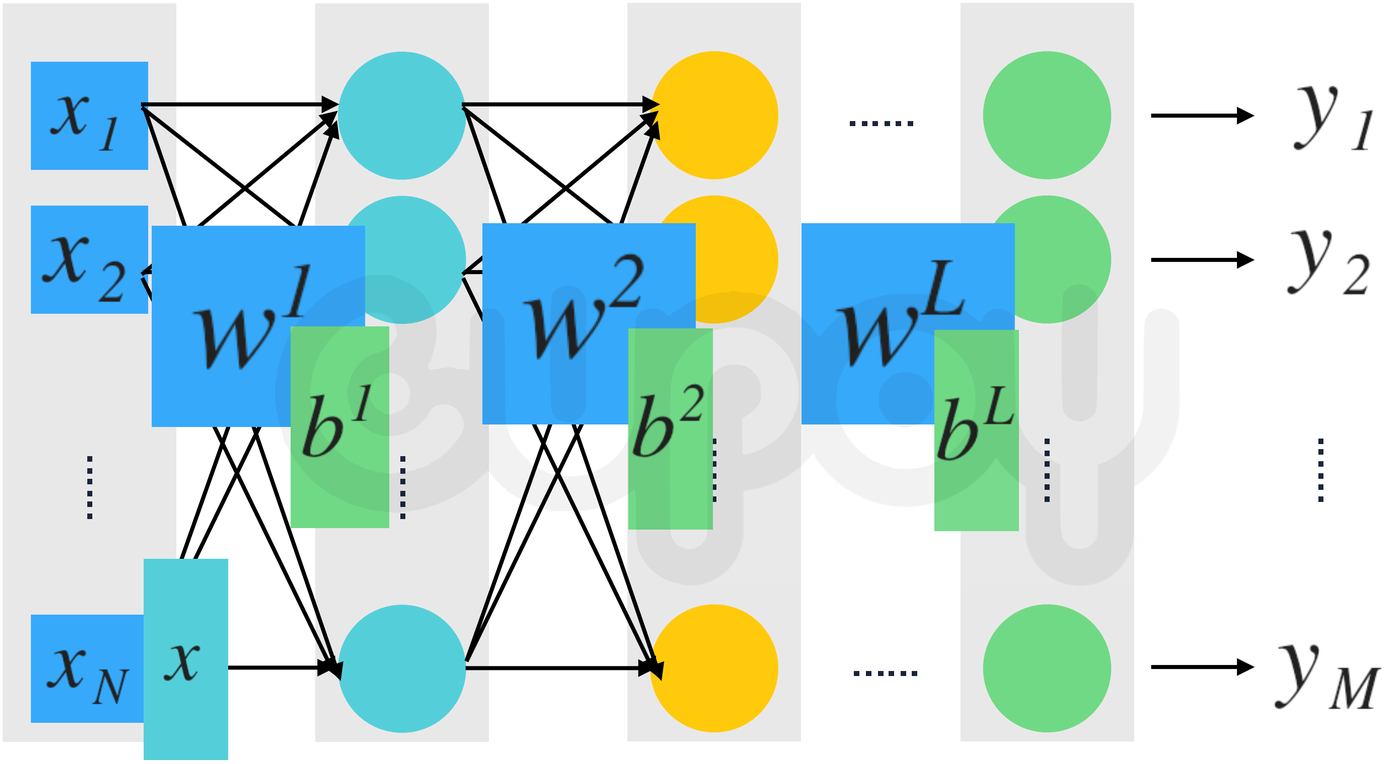




神經元間資料傳遞與運算流程

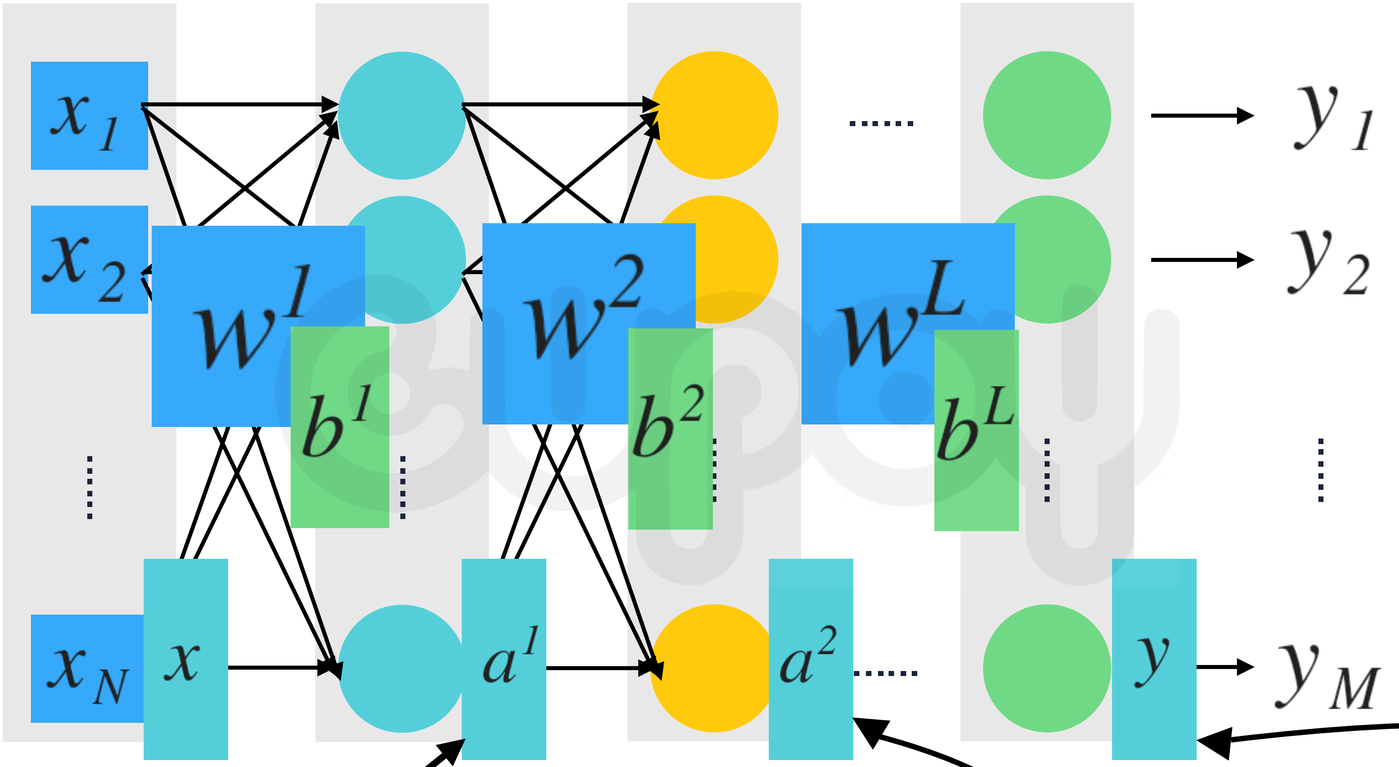


神經元間資料傳遞與運算流程



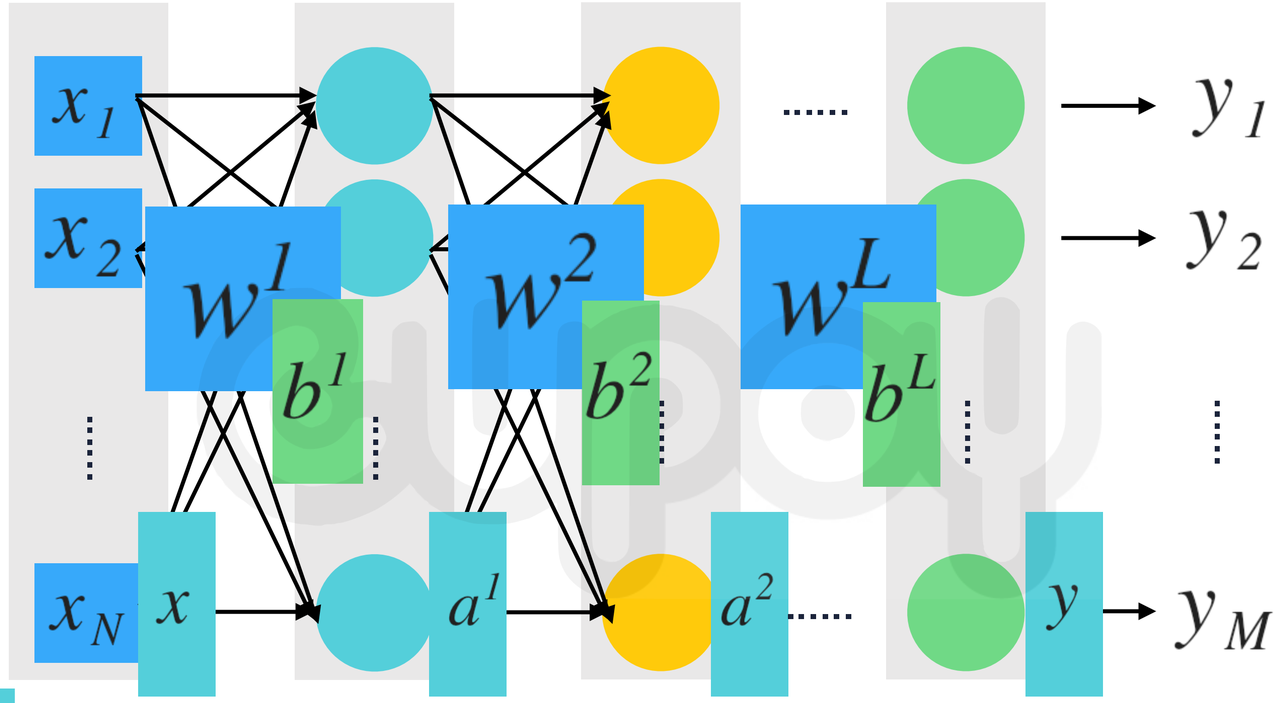
$$\sigma\left(w^1 \; x + b^1 \right)$$

神經元間資料傳遞與運算流程



$$\sigma\left(w^1 x + b^1\right) \quad \sigma\left(w^2 a^1 + b^2\right) \quad \sigma\left(w^L a^{L-1} + b^L\right)$$

神經元間資料傳遞與運算流程



$$y = f(x)$$ Using parallel computing techniques to speed up matrix operation

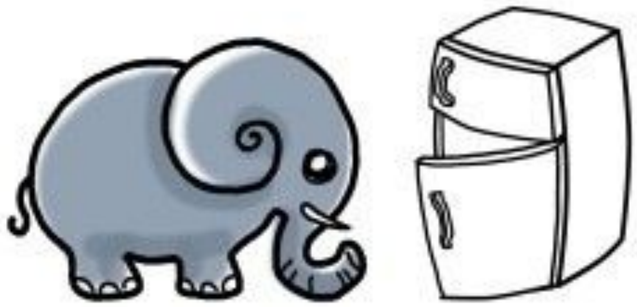
$$= \sigma(W^L \cdots \sigma(W^2 \sigma(W^1 x + b^1) + b^2) \cdots + b^L)$$

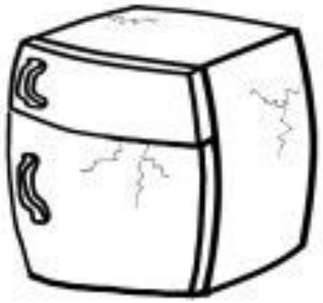
Step2+3 goodness of function and pick the best function

01 Define a set of Functions

02 Evaluate Functions

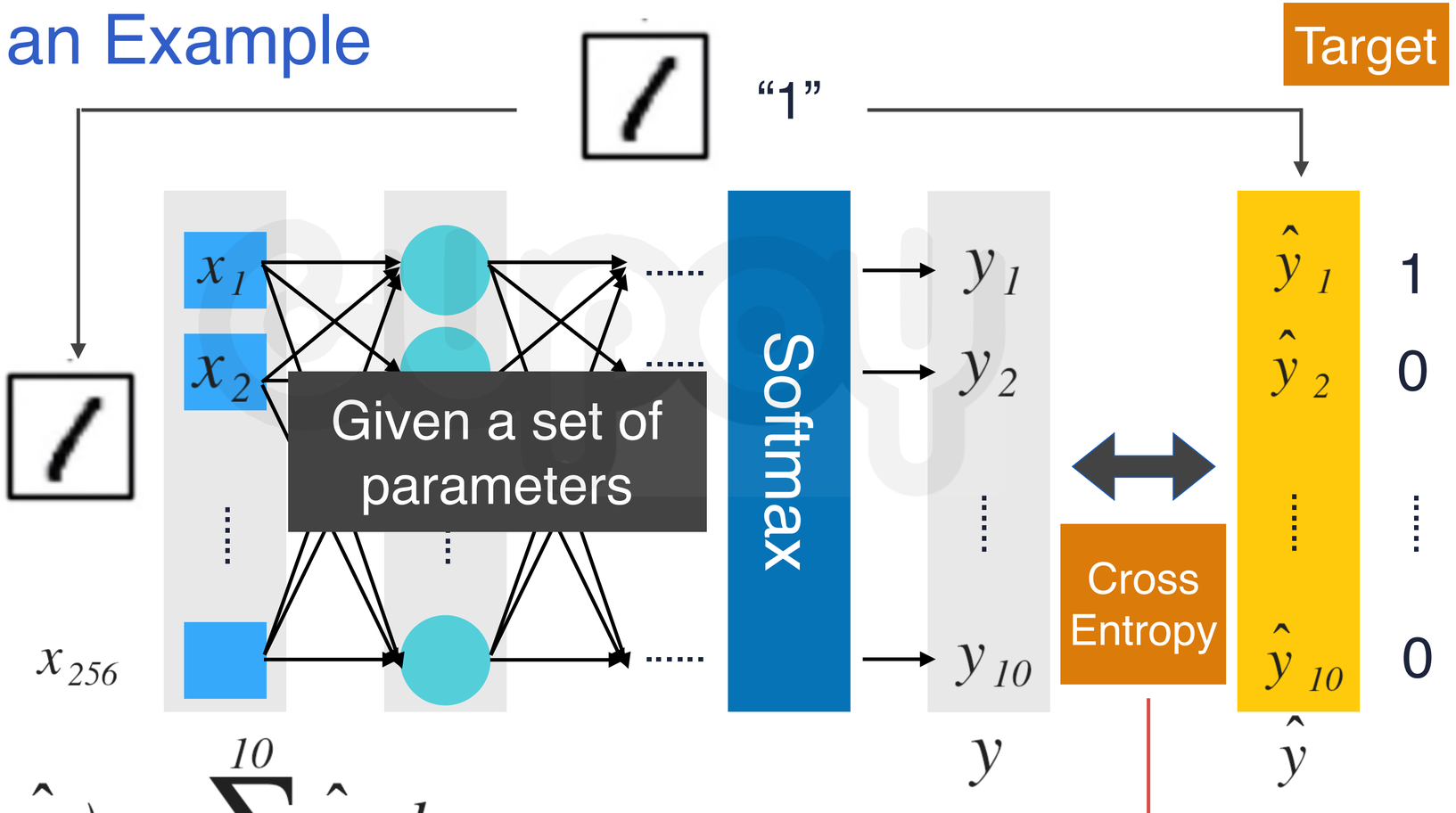
03 Pick the best Function





Step2+3 goodness of function and pick the best function Neural Network

Loss for an Example

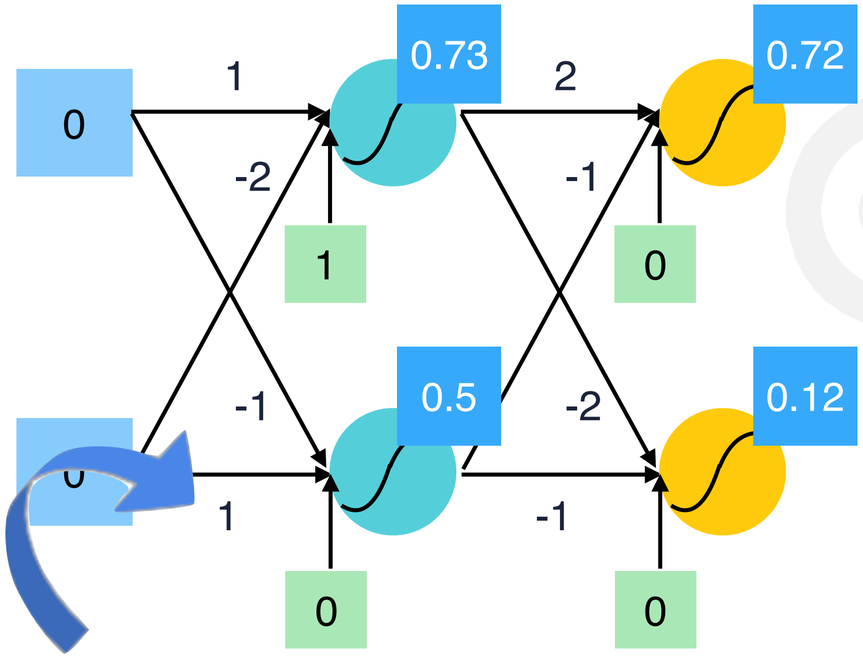


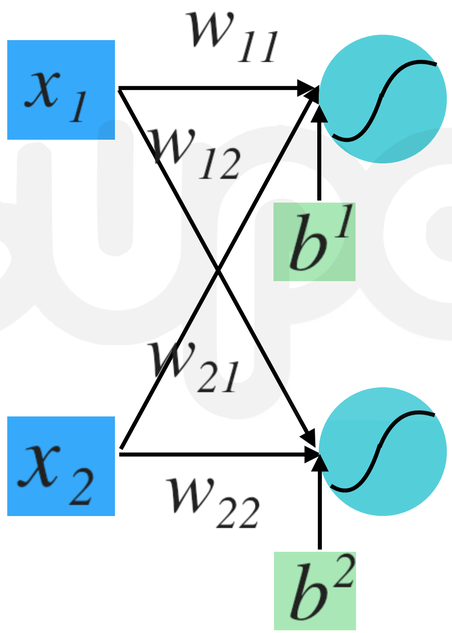
$$c(y, \hat{y}) = -\sum_{i=1}^{10} \hat{y}_i \ln y_i$$

圖片來源：李宏毅老師 - Brief Introduction of Deep Learning

一些矩陣與向量的基礎知識介紹

1、為什麼Input Layer的輸入要轉成向量?





$$a_1 = w_x + b$$

$$= \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -2 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\sigma(a_1) = \begin{bmatrix} \sigma(1) \\ \sigma(0) \end{bmatrix} = \begin{bmatrix} 0.73 \\ 0.5 \end{bmatrix}$$

2、使用Numpy 操作矩陣乘法 (*/dot/multiple) (1/2)

向量內積(numpy.dot)

$W=(w_1,w_2,\cdots,w_K)$

$V=(v_1,v_2,\cdots,v_K)$

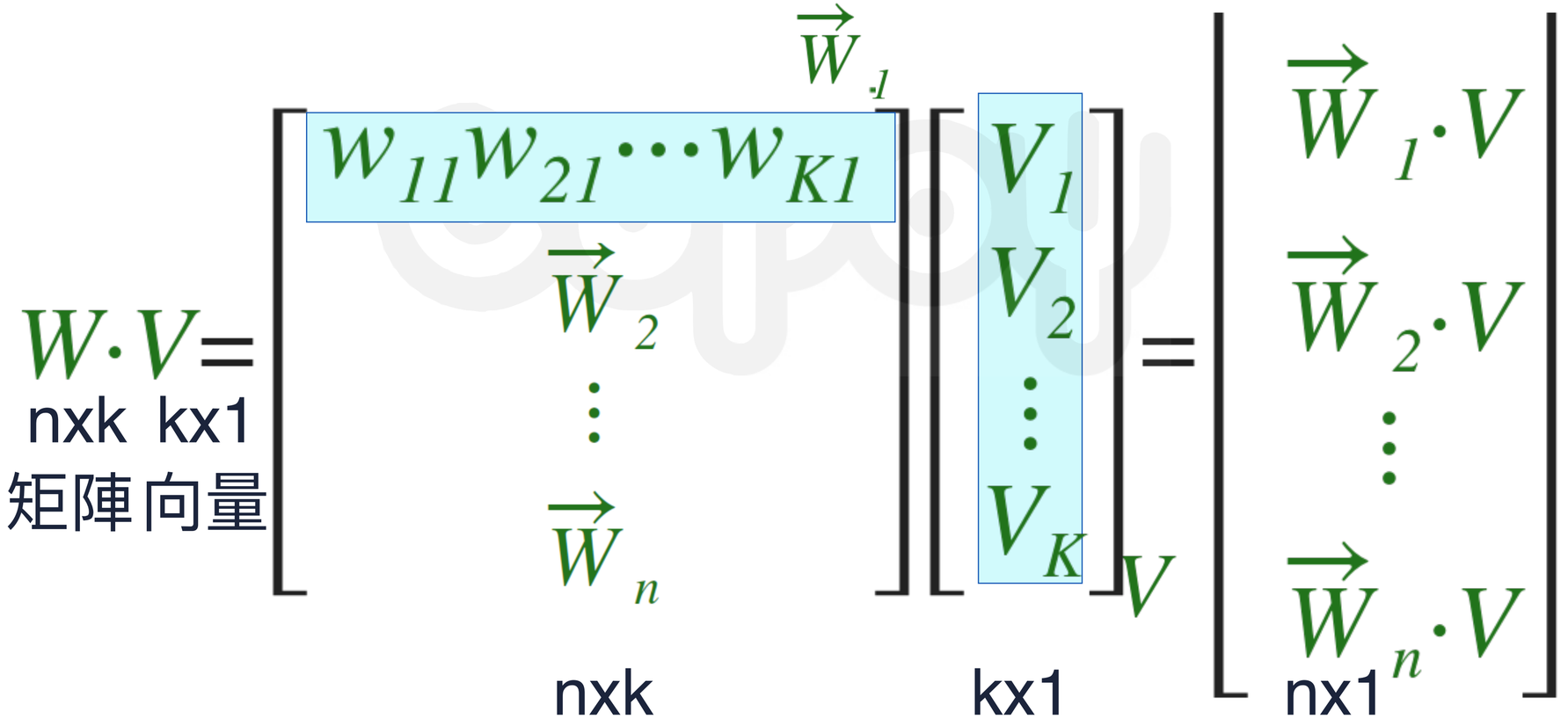
則 $W\cdot V=w_1v_1+w_2v_2+\cdots+w_Kv_K$ 稱為內積

為了與矩陣運算有一致性，會寫成下列形式但內容不變

$$W^TV=[w_1w_2\cdots w_K]\begin{bmatrix}v_1\\v_2\\\vdots\\v_K\end{bmatrix}$$

2、使用Numpy 操作矩陣乘法 (*/dot/multiple) (2/2)

類神經計算時，最常用到的運算是矩陣x向量



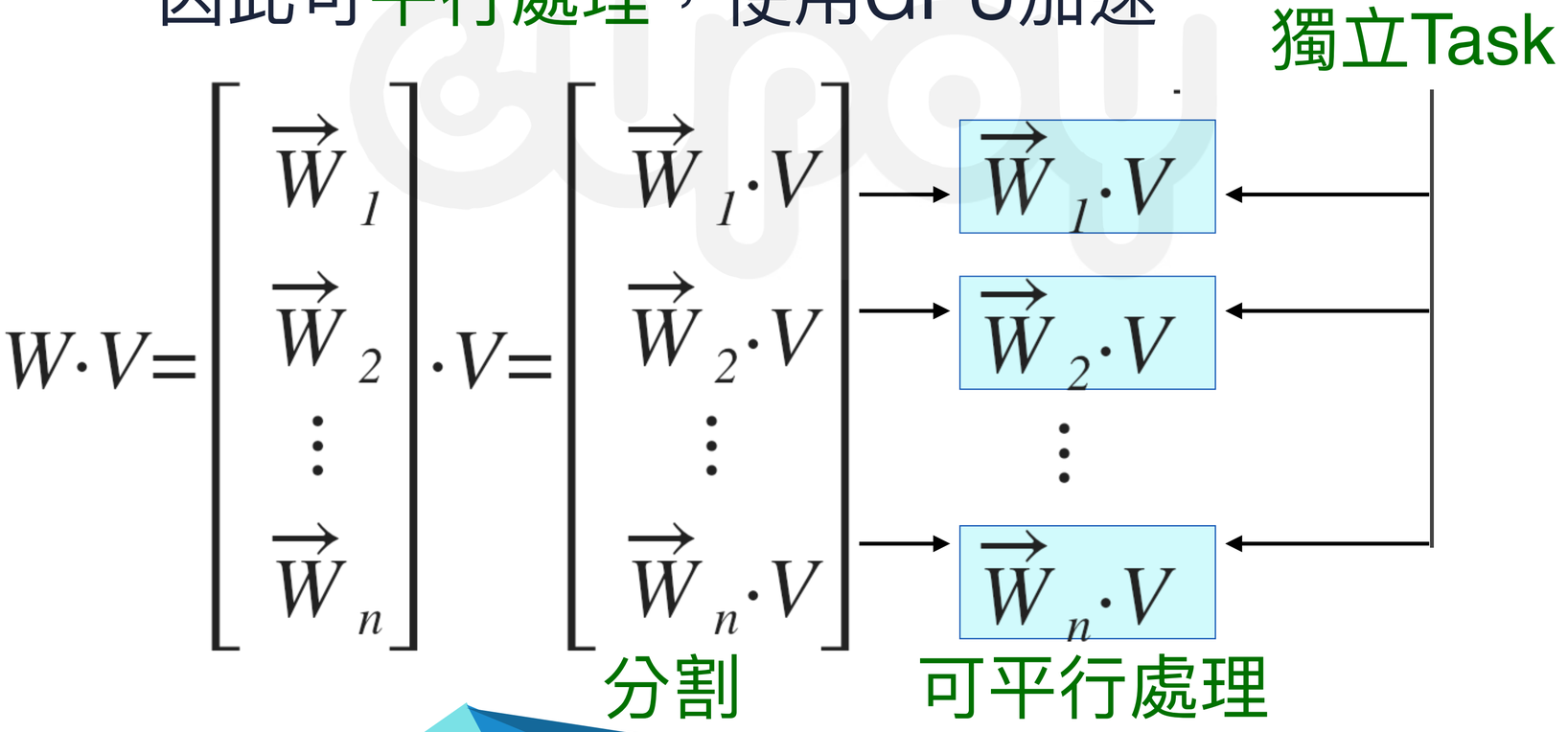
在numpy中，矩陣相乘/向量內積都是dot

3、應用GPU加速矩陣運算的範例

GPU如何加速矩陣運算

由於乘積的每個元素計算獨立

因此可平行處理，使用GPU加速

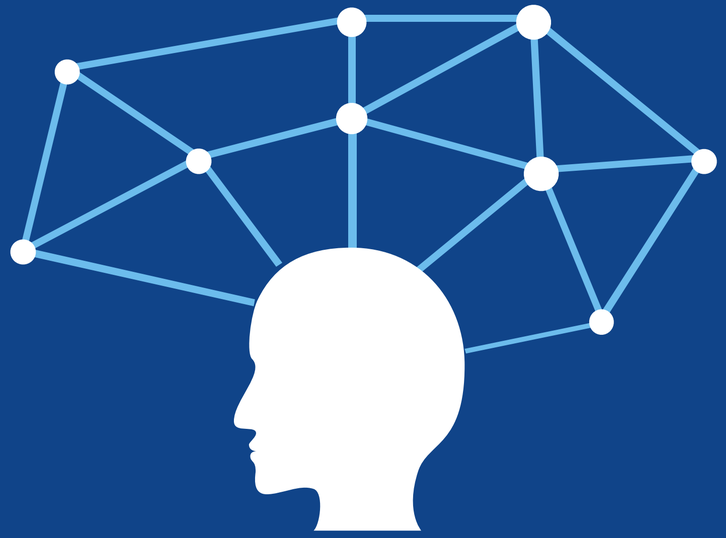


FAQ
常見問題

01

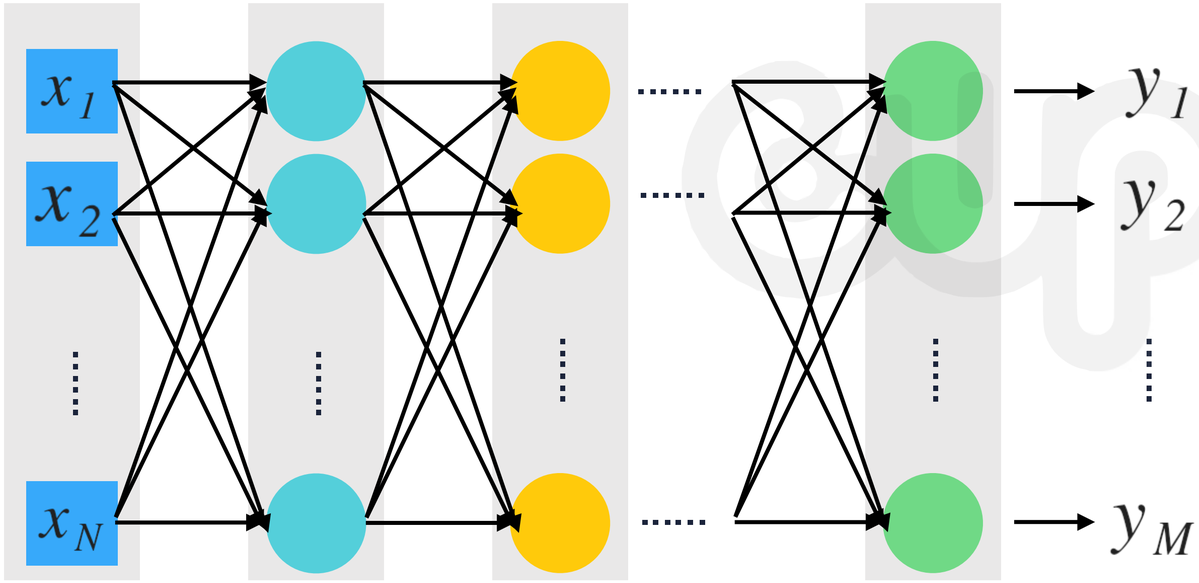
深度學習要有多少Layers？

每個Layer要有多少Neuron？



Ans:

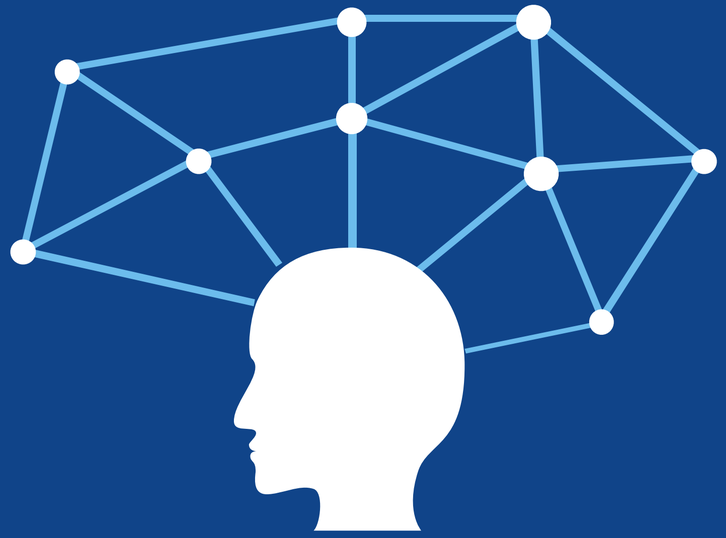
“一般來說是經驗法則，Trial & Error .”



Trail and Error + Intuition

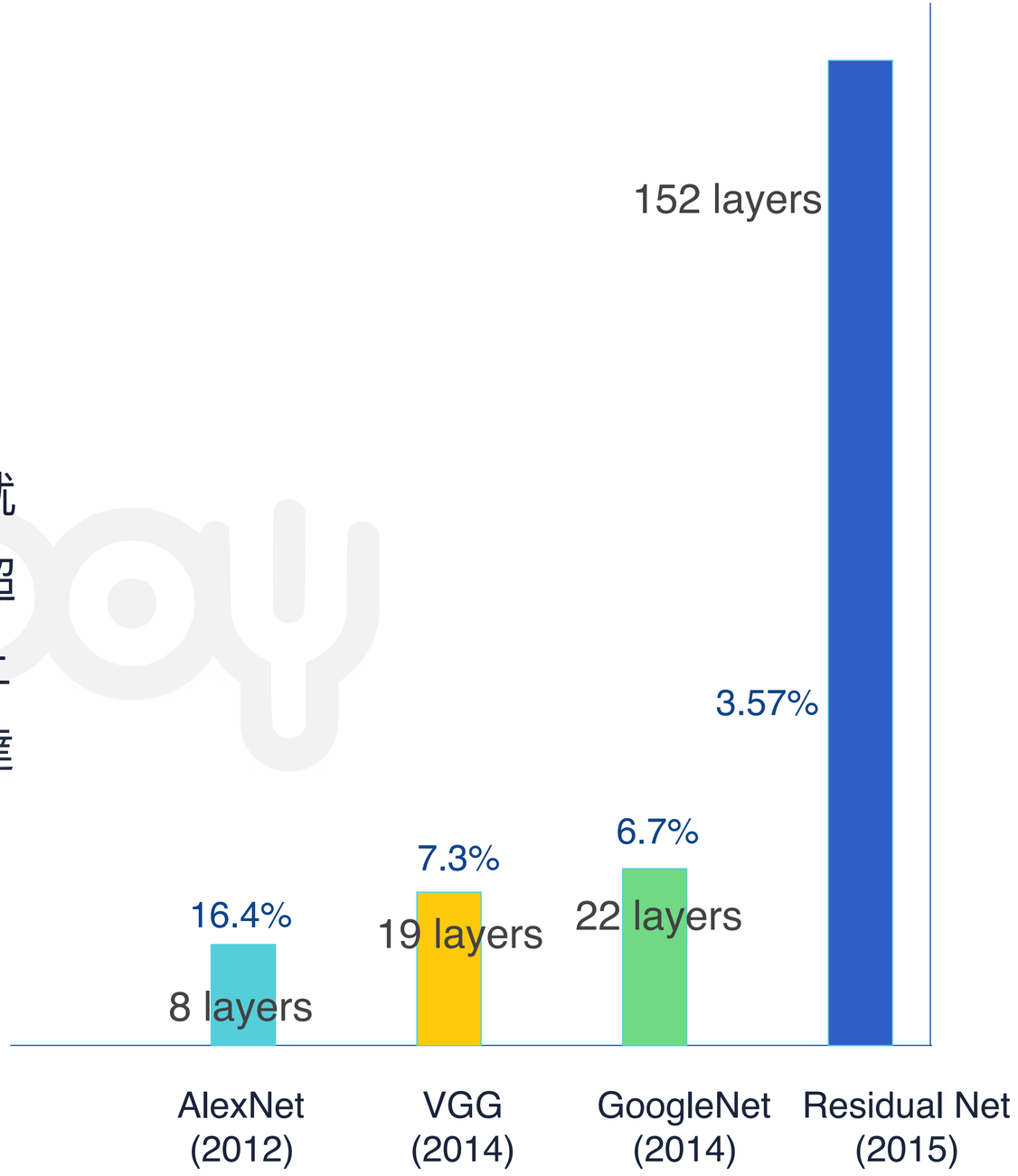
02

要幾層才叫DNN？DNN的Hidden layers 可以有多少層?



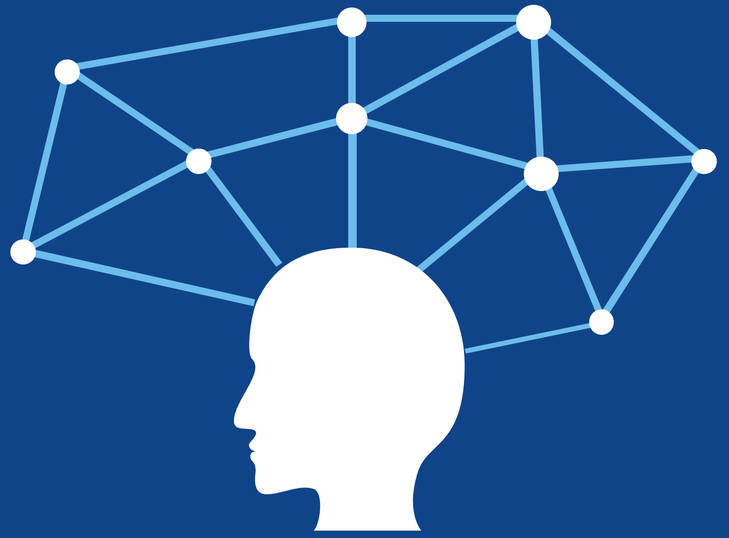
Ans:

一般來說是3層以上才叫DNN (但也有人用類神經就叫深度學習的)。未經過特殊設計的NN, 隱藏層數超過二十幾層就不會更好了。但使用 ResNet, 理論上只要資料足夠, 會越深越好(競賽中ResNet模型到達152層)



03

深度學習層數是越多越好嗎?



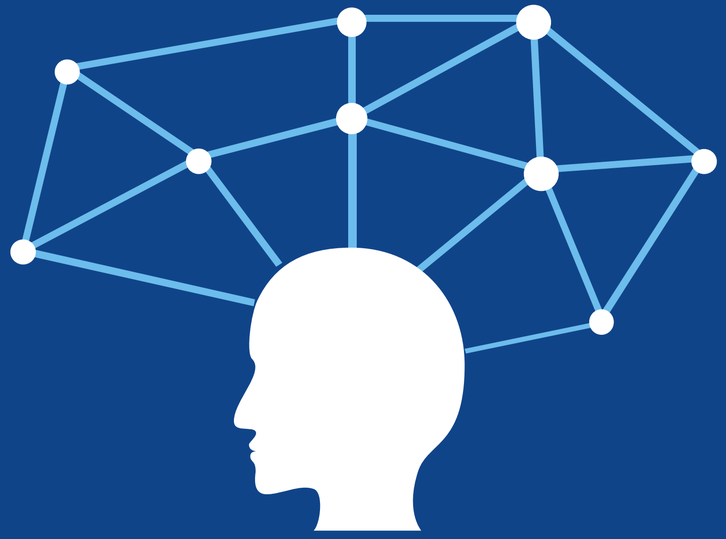
Ans:

未經過特殊設計的NN, 隱藏層數超過二十幾層就不太會更好了

不過有BN(批次標準化)可以更深, 而有了ResNet後, 如果資料量足夠, 越多層有可能越好

(AlphaGo Zero 版就有採用 RL+ResNet 概念, 因為能夠自己持續產生對弈棋譜, 因此能越來越好)

之後還有新的DNN設計　也幾乎都會以ResNet做為參考基準之一

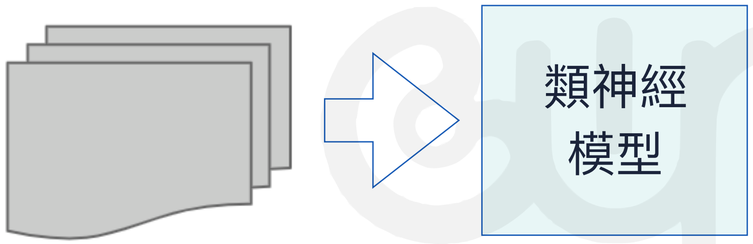


04

傳統的ML，需要先進行Feature Transform / Feature Engineer，但DL不用，那麼DL產生的問題是如何決定Neural Network Structure?

Ans: 深度學習 v.s. 機器學習

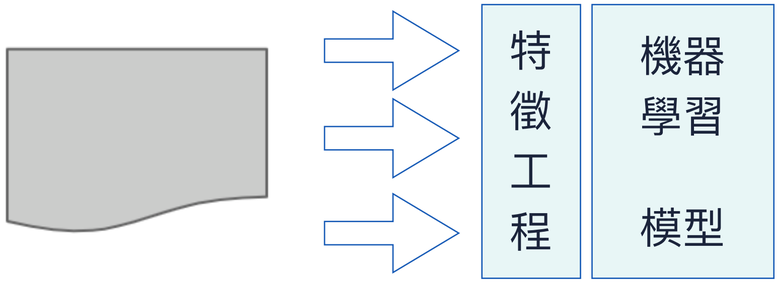
輸入資料　輸入函數　評價方式　輸入最好



類似

GridSearch
RandomSearch

深度學習 輸入大量資料，類神經自行歸納出規則



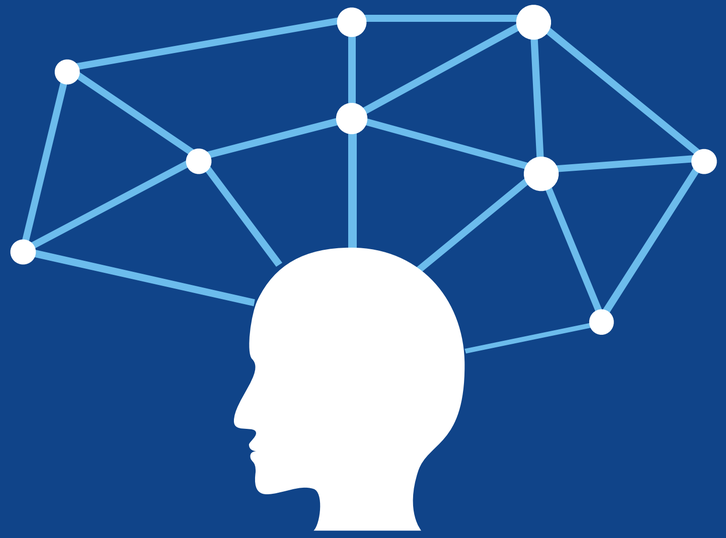
類似

SDG

機器學習 輸入人工刻畫的特徵，搭配機器學習模型預測

05

對於語音 / 影像辨識 / 自然語言等問題, 深度學習是否比機器學習效果還要好?



Ans:

目前來說，這三個領域上都是深度學習更好，所以我們可以知道：如果是人類難以說清楚的特徵，通常深度學習的效果會比較好，因為能自動生成難以用規則制定的特徵。

參考資料

- 李宏毅老師 - Brief Introduction of Deep Learning
- 深度學習-神經元（neuron）與感知機（perceptron）
- 機器學習 - 神經網路
  (多層感知機 Multilayer perceptron, MLP)運作方式

深度學習簡介

機器學習百日馬拉松