深度學習理論概念

Deep Learning Basics

2025/ 06/ 17

PRESENTED BY AI Foundation

CREATED FOR

INNOLUX

目錄

Machine Learning and Deep Learning

機器學習與深度學習

深度學習歷程

人工智慧 Artificial Intelligence
透過電腦程式來呈現人類智慧的技術

機器學習 Machine learning
讓電腦自主學習規律來解決問題

深度學習 Deep Learning
以神經網路架構進行學習

1950s　1980s　2010s

深度學習歷程

人工智慧 Artificial Intelligence

透過電腦程式來呈現人類智慧的技術

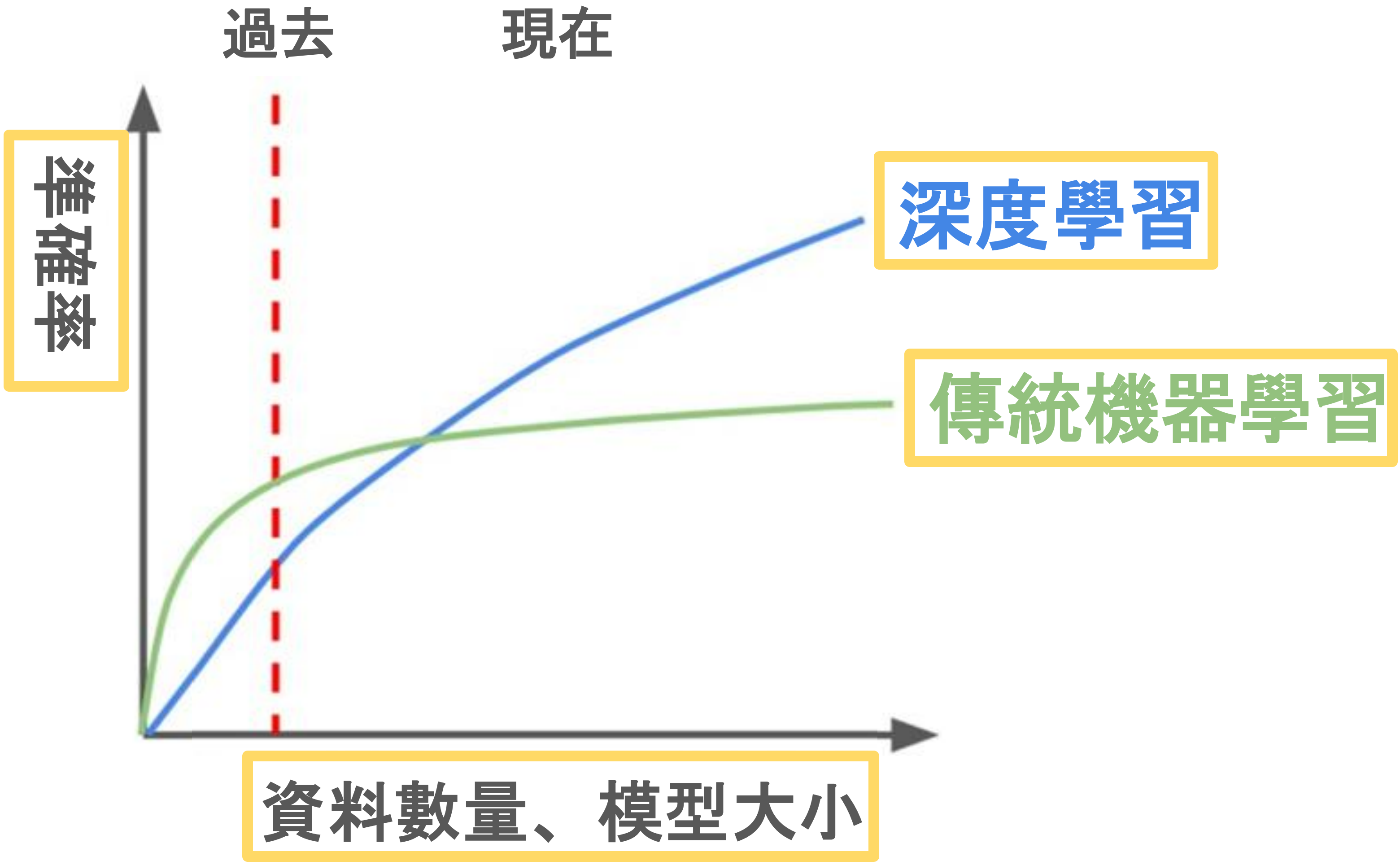
機器學習 Machine learning

讓電腦自主學習規律來解決問題

深度學習 Deep Learning?

以神經網路架構進行學習

為何需要深度學習?



機器學習(Review)

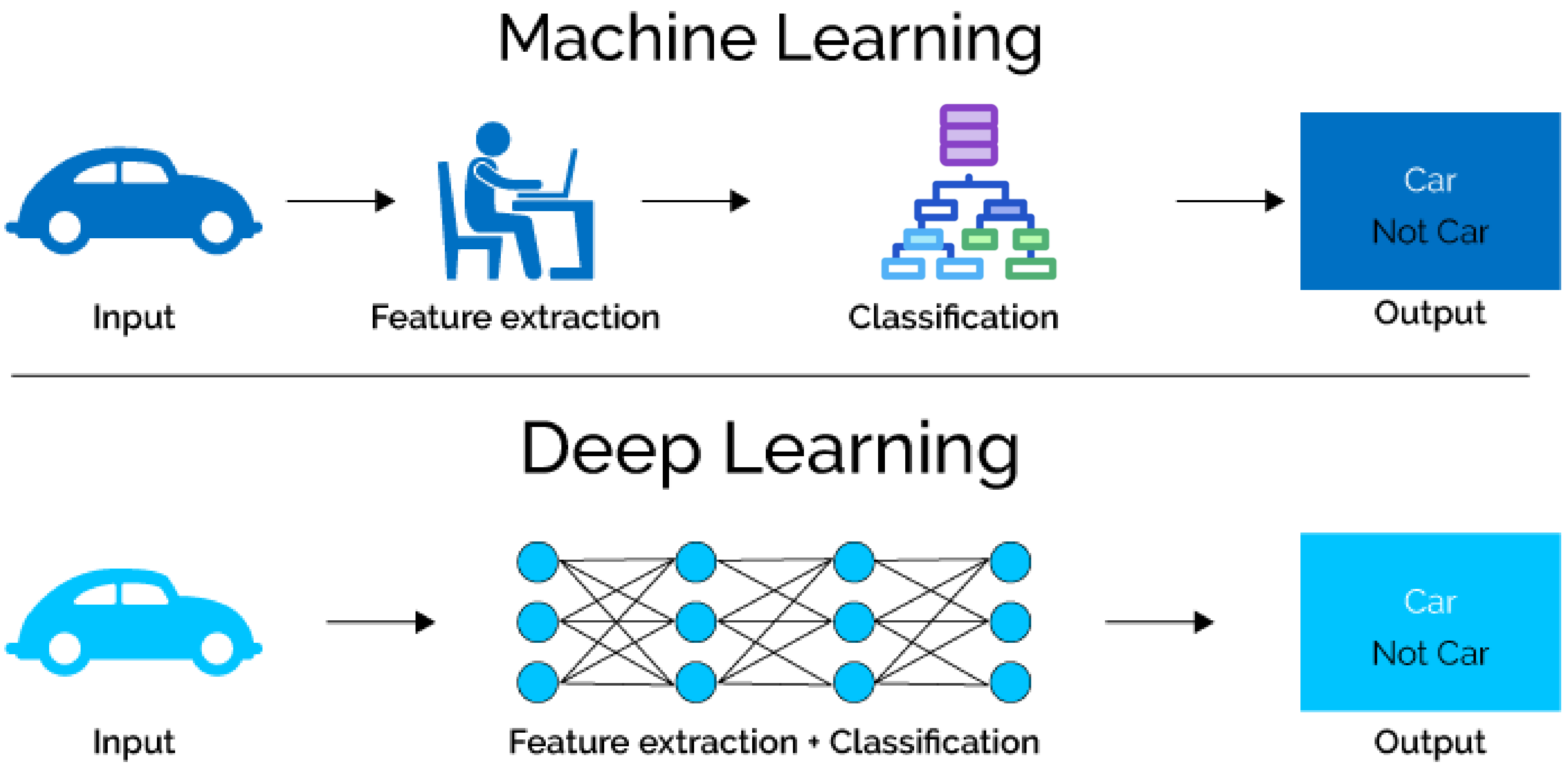
機器學習：從資料中學習資料間的關聯，並利用此關聯對未知資料進行預測

1. 依據目標決定訓練任務：
- 分類：目標為類別變數，ex. 文本分類、瑕疵檢測
- 迴歸：目標為連續數值，ex. 房價預測、股價預測
- 分群：無目標，期望找出樣本間的關係，ex. 喜好分群

2. 決定機器學習演算法：
- 分類 : Logistic Regression
- 迴歸 : Linear Regression
- 分群 : K-Means、DBSCAN
- 可做分類與迴歸： SVM、KNN、Tree Based Model

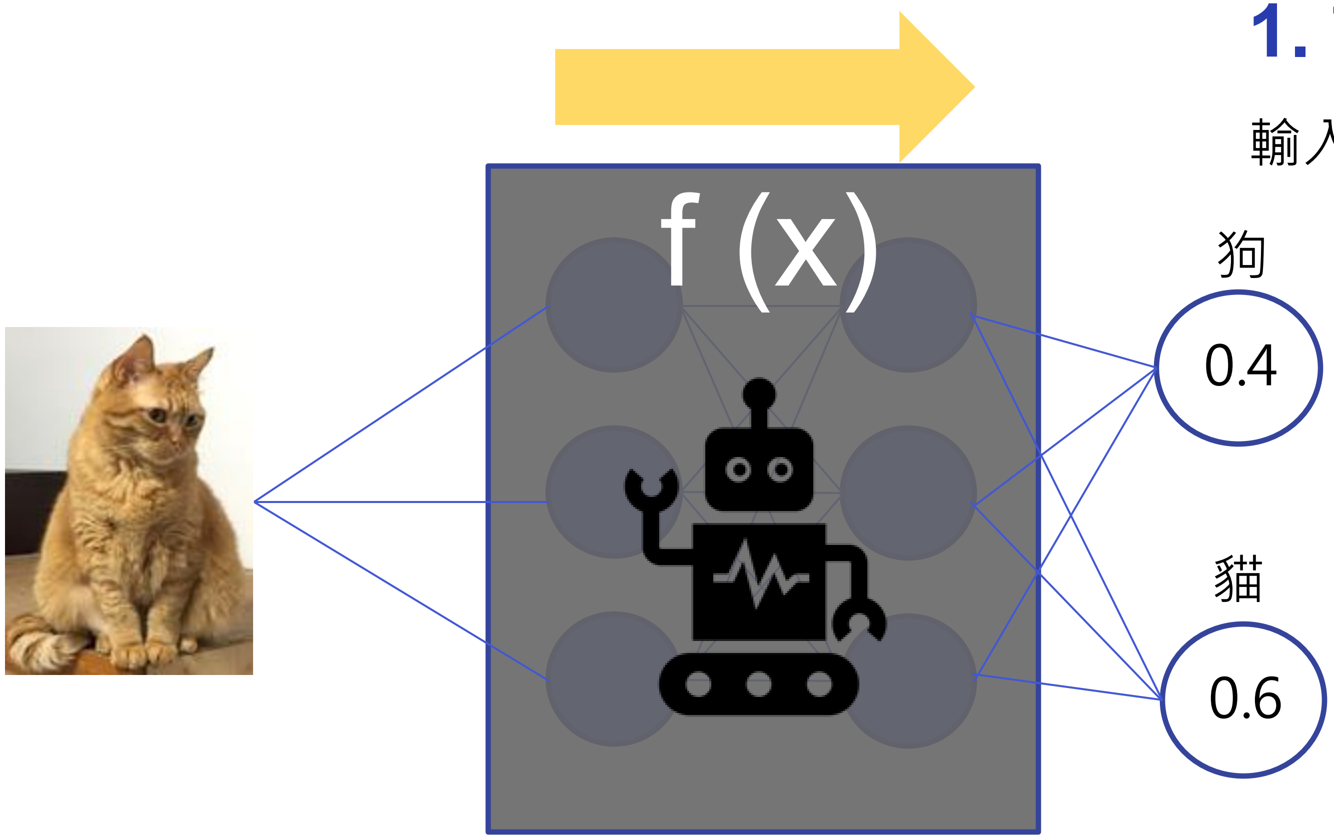
3. 評估效果指標：
- 分類 : Accuracy、Precision、Recall、F1-Score
- 迴歸 : MSE、MAE、R^2

機器學習 vs 深度學習



圖片來源：Xenonstack

深度學習訓練流程



1. 前向傳播(Forward Propagation)

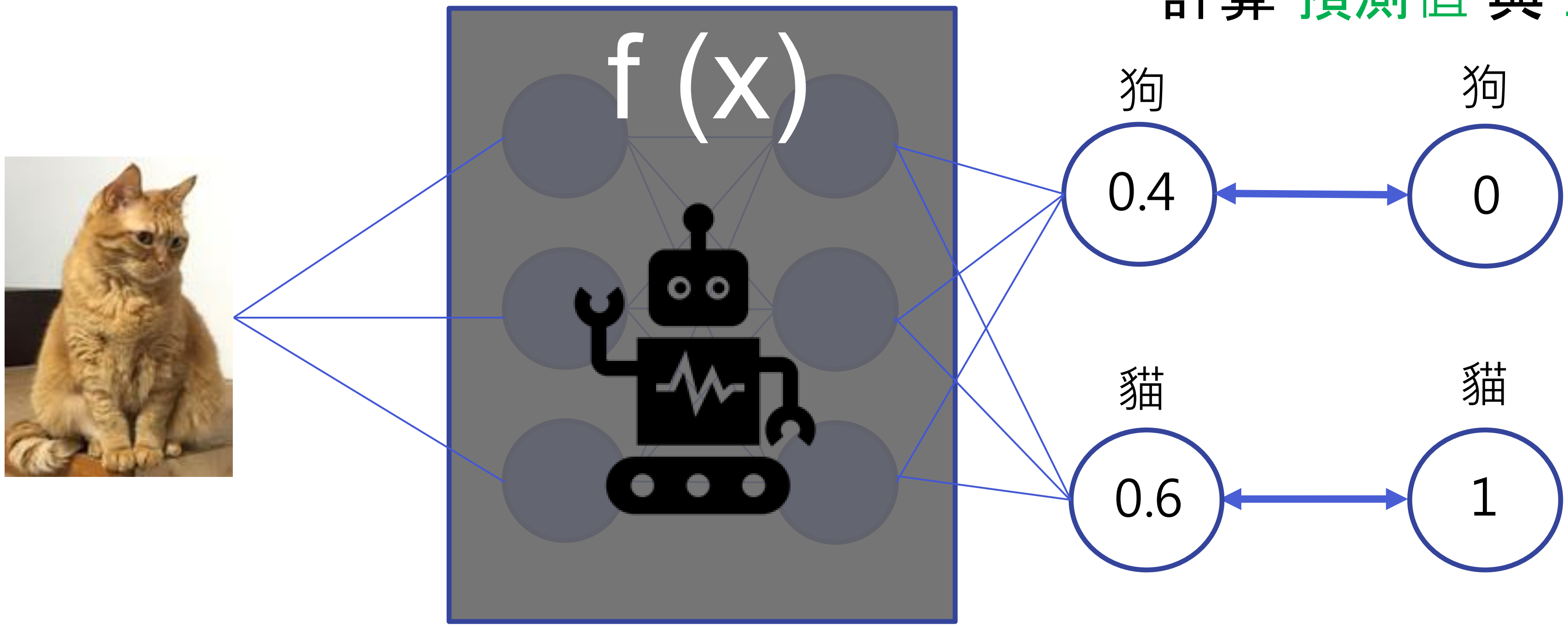
輸入層透過模型參數運算得到輸出層預測值

前向傳播 計算誤差 反向傳播

深度學習訓練流程

2. 計算誤差

計算 預測值 與 正確答案 的落差



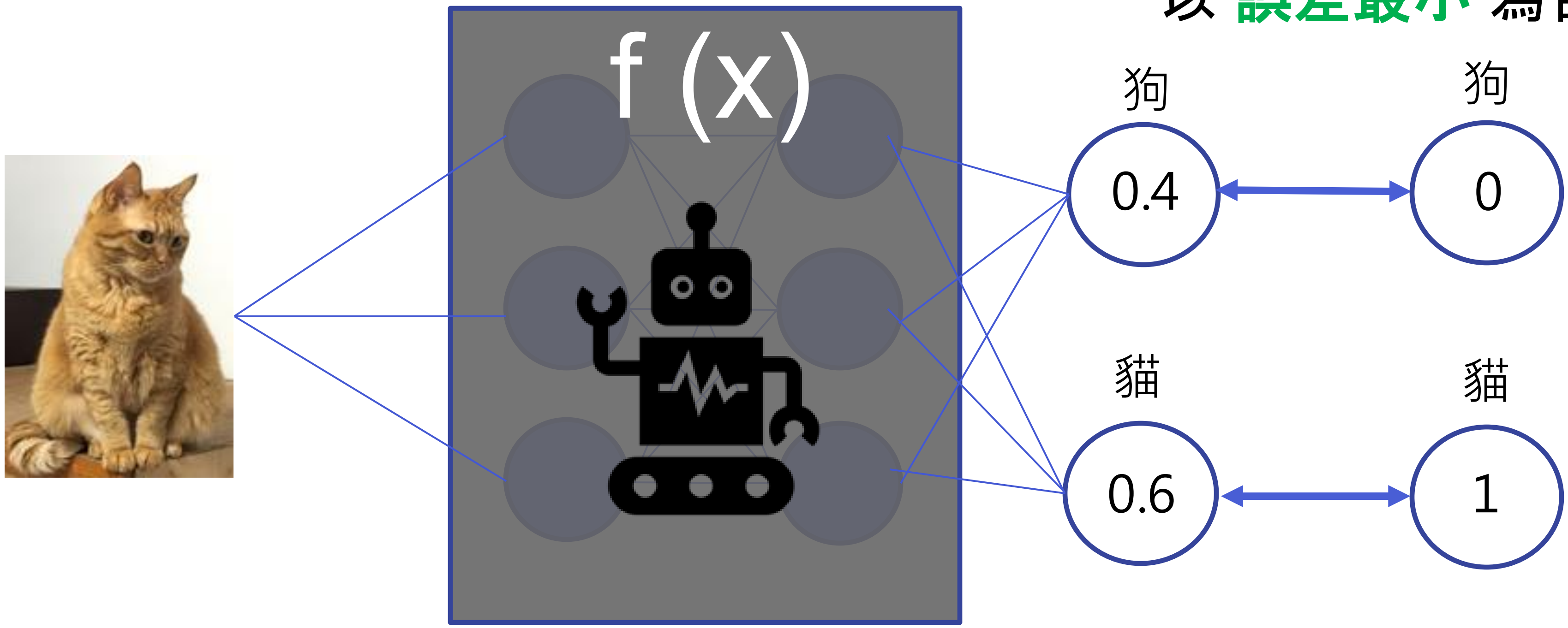
前向傳播 計算誤差 反向傳播

深度學習訓練流程

調整模型參數

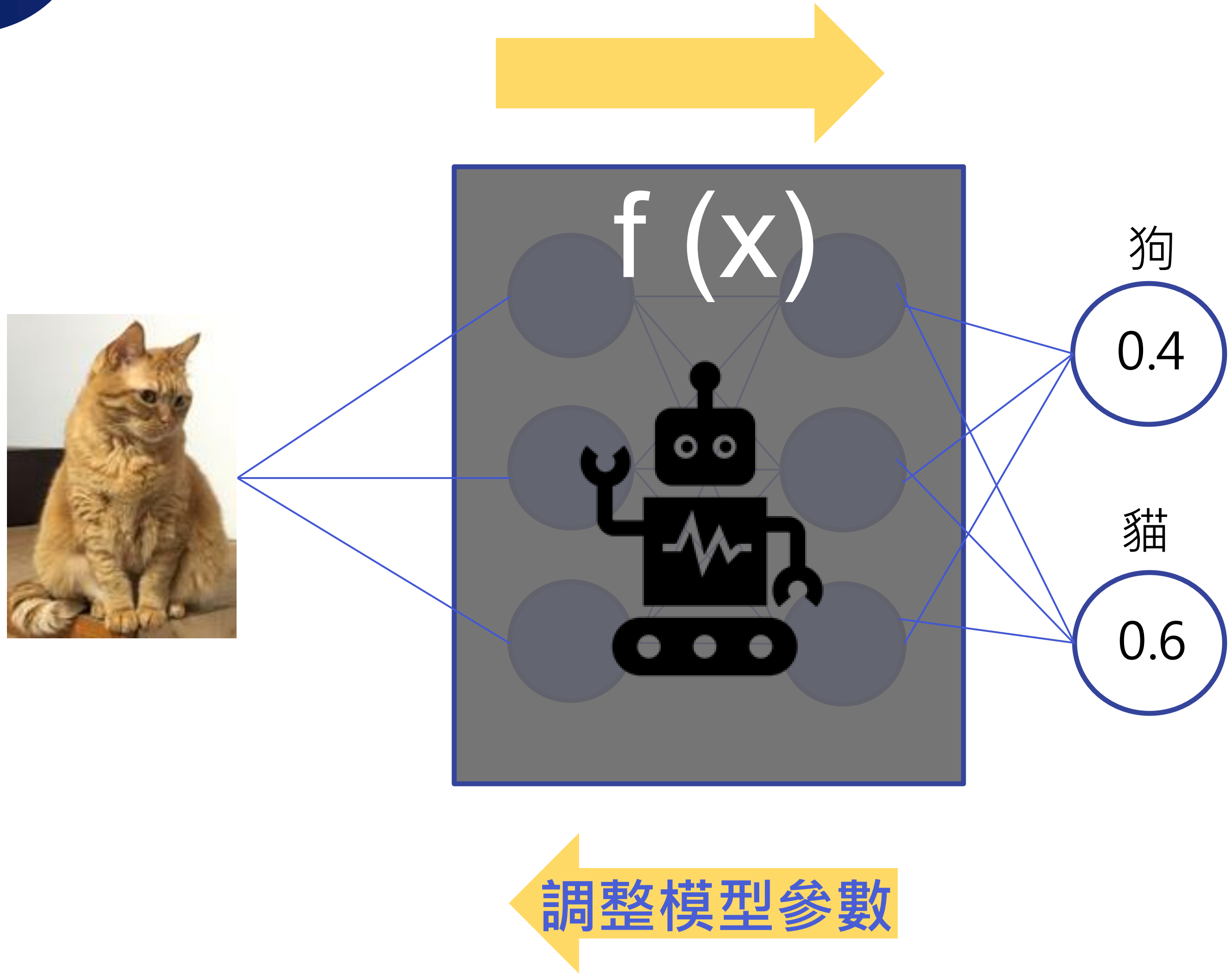
3. 反向傳播 (Backward-Propagation)

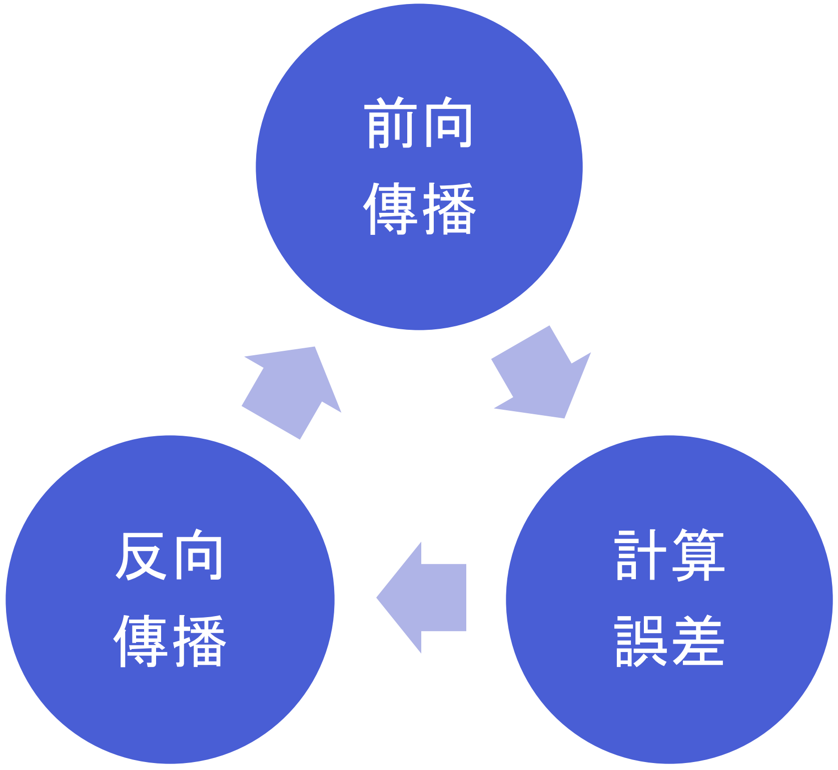
以 **誤差最小** 為目標，往回調整模型參數



前向傳播 → 計算誤差 → **反向傳播**

深度學習訓練流程





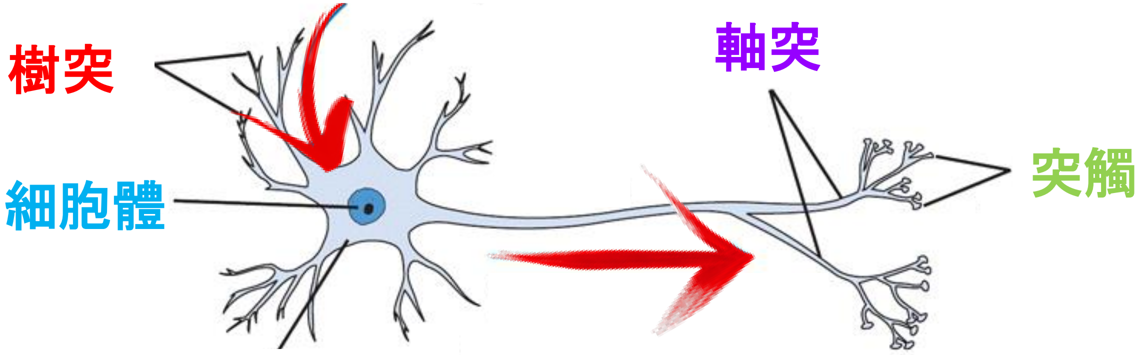
直到誤差足夠低....

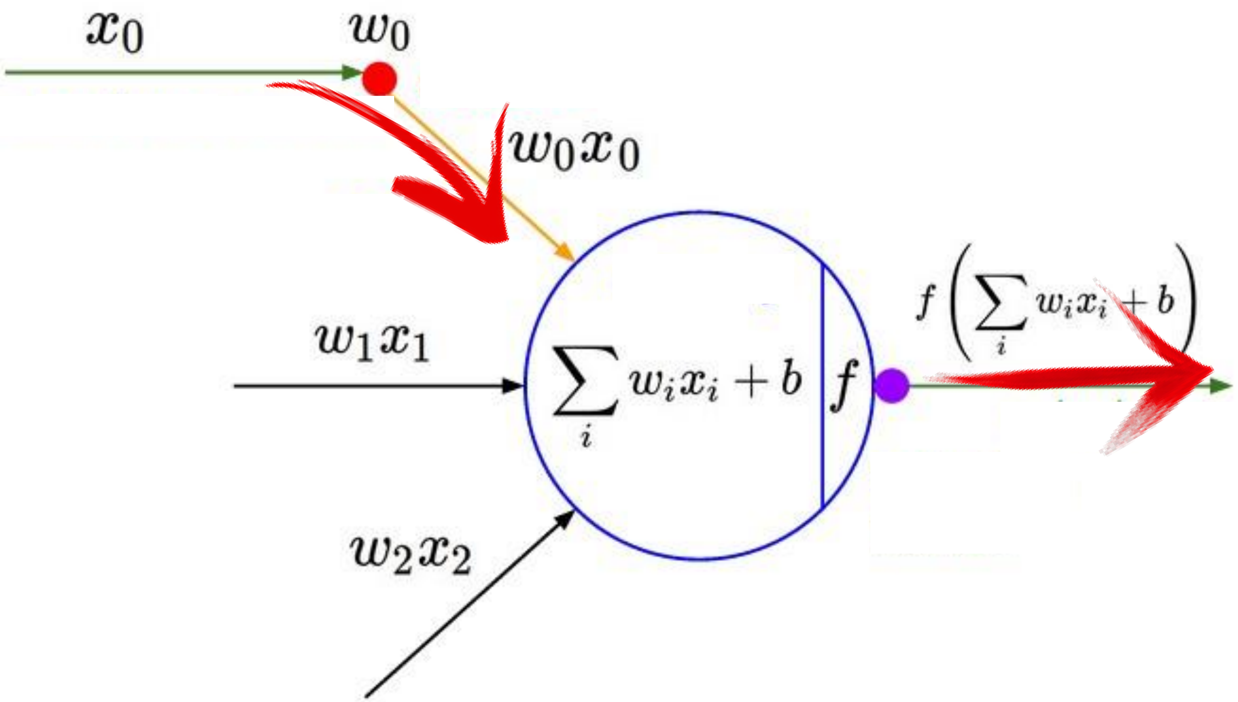
Multilayer Perceptron, MLP

多層感知器的模型架構

神經網路的最小運算單位：神經元

模型架構的想法：仿造生物神經網路的結構及運作方式



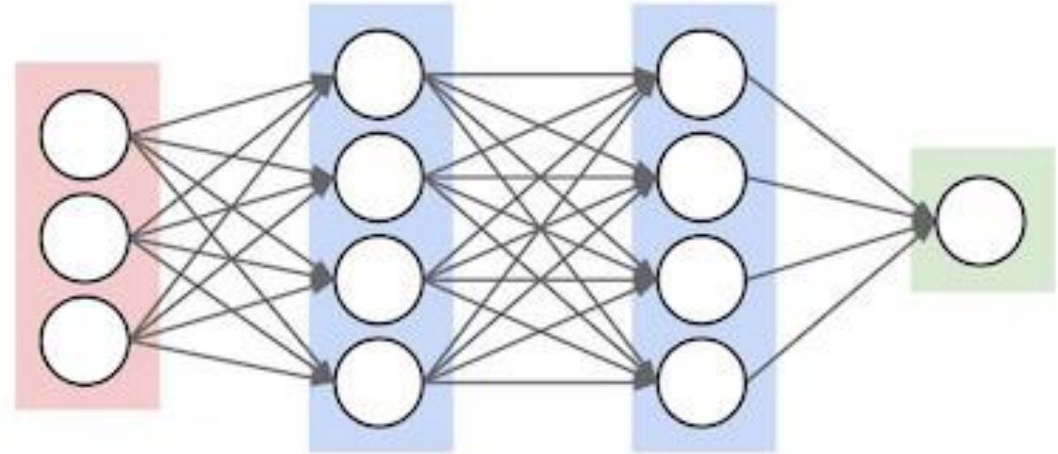


資料來源：cs231n

神經網路的架構

生物神經細胞組合
(Biological Neurons)

神經網路
(Neural Network)



資料來源：cs231n

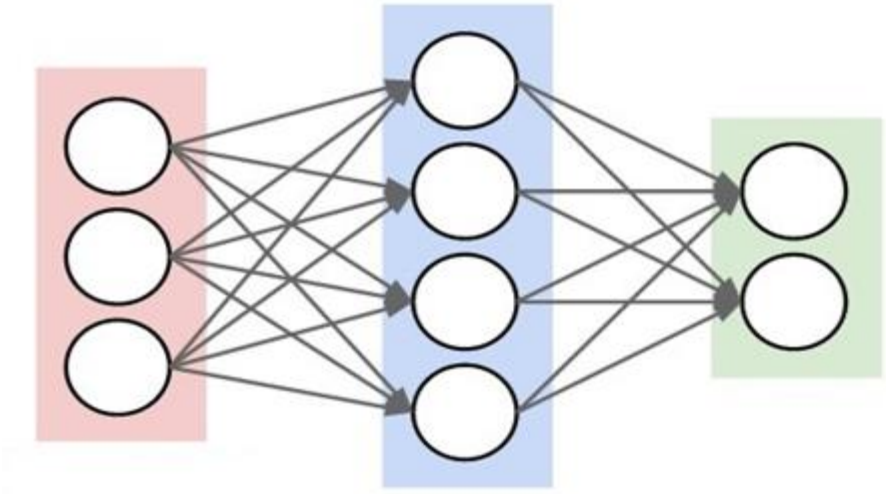
深度學習的模型架構

模型架構的想法：仿造生物神經網路的結構及運作方式

多層感知器(Multilayer Perceptron, MLP)

感知器
(Perceptron)

多層感知器
(Multilayer Perceptron, MLP)



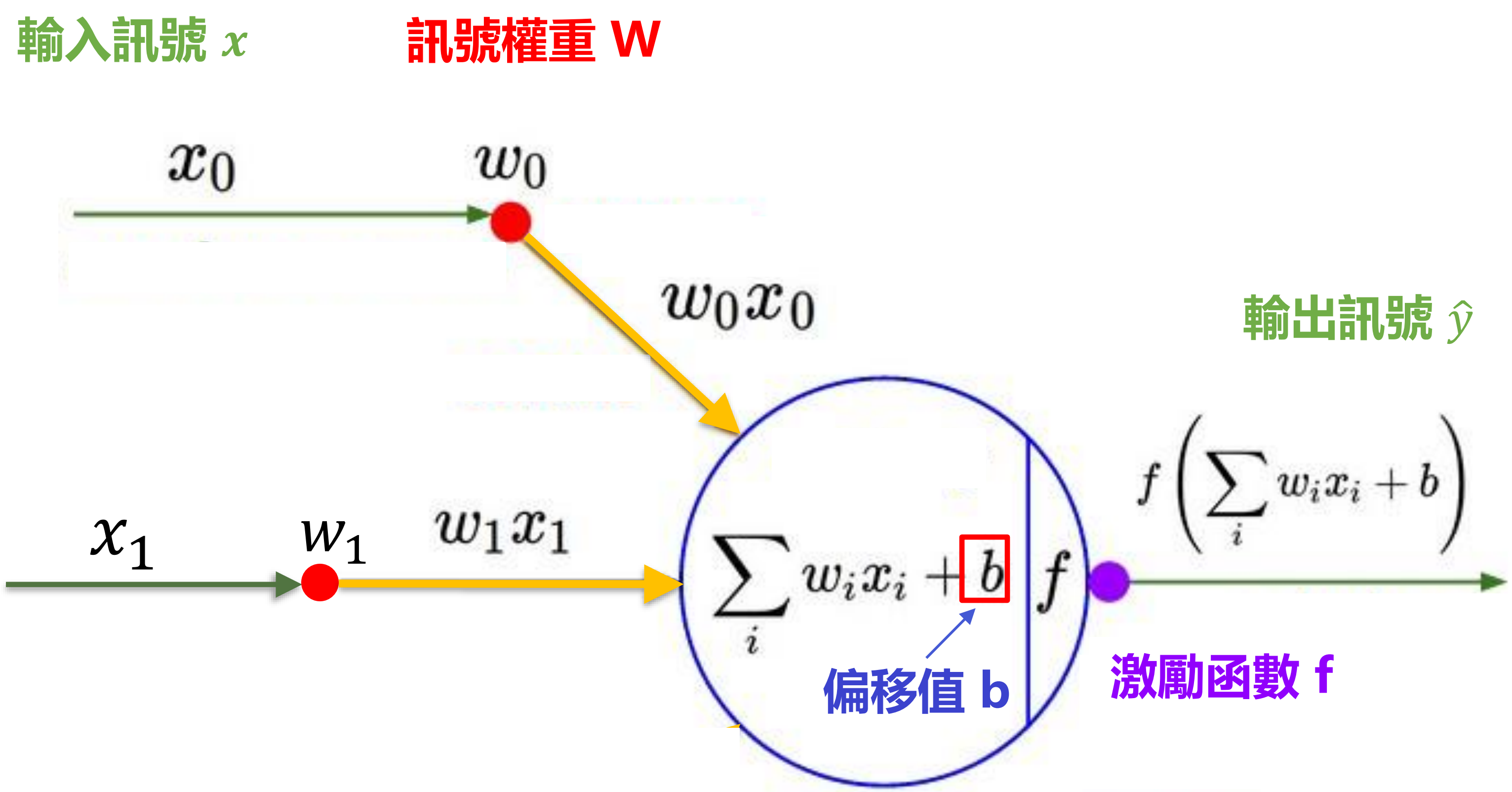
(類)神經網路(Neural Network, NN)、
人工神經網路(Artificial Neural Network, ANN)

神經元
(Neuron)

神經網路
(Neural Network, NN)

資料來源：cs231n

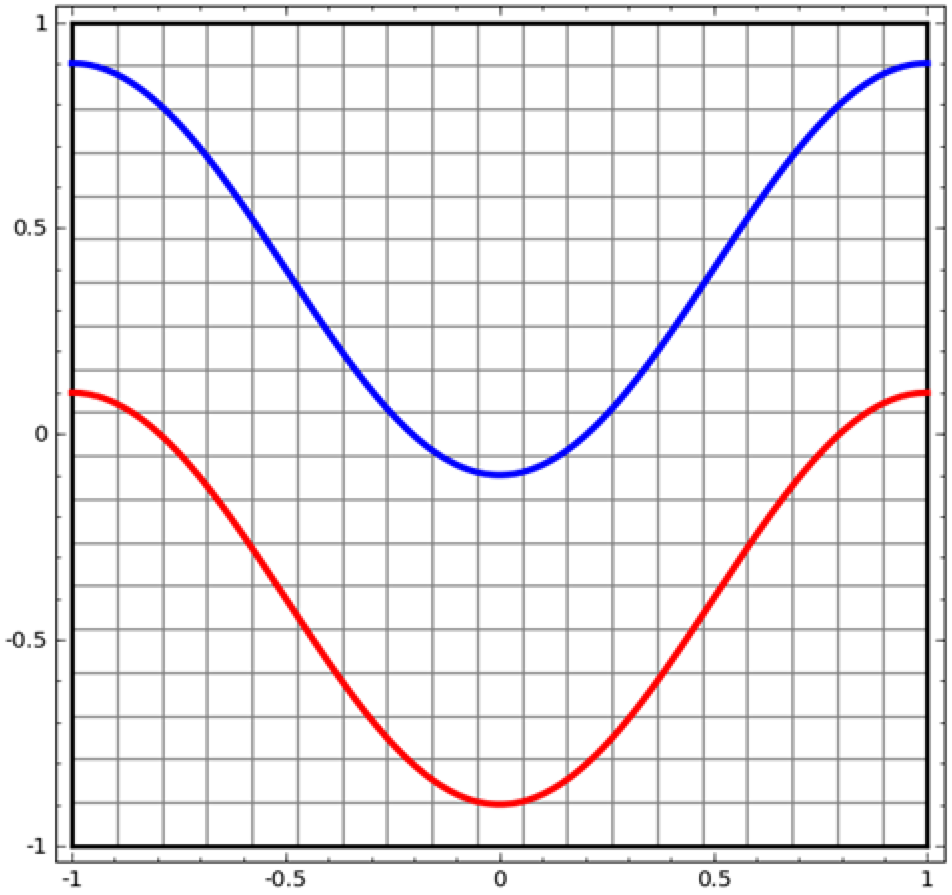
神經元架構



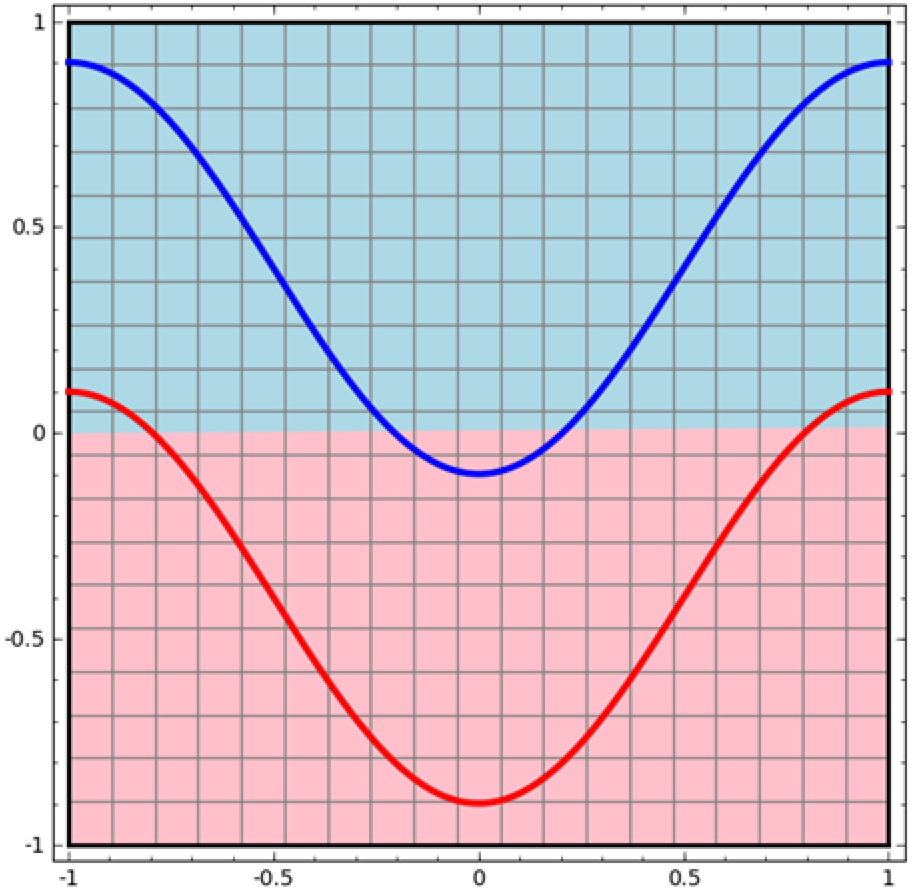
資料來源：cs231n

激勵函數

Linear non-separability：

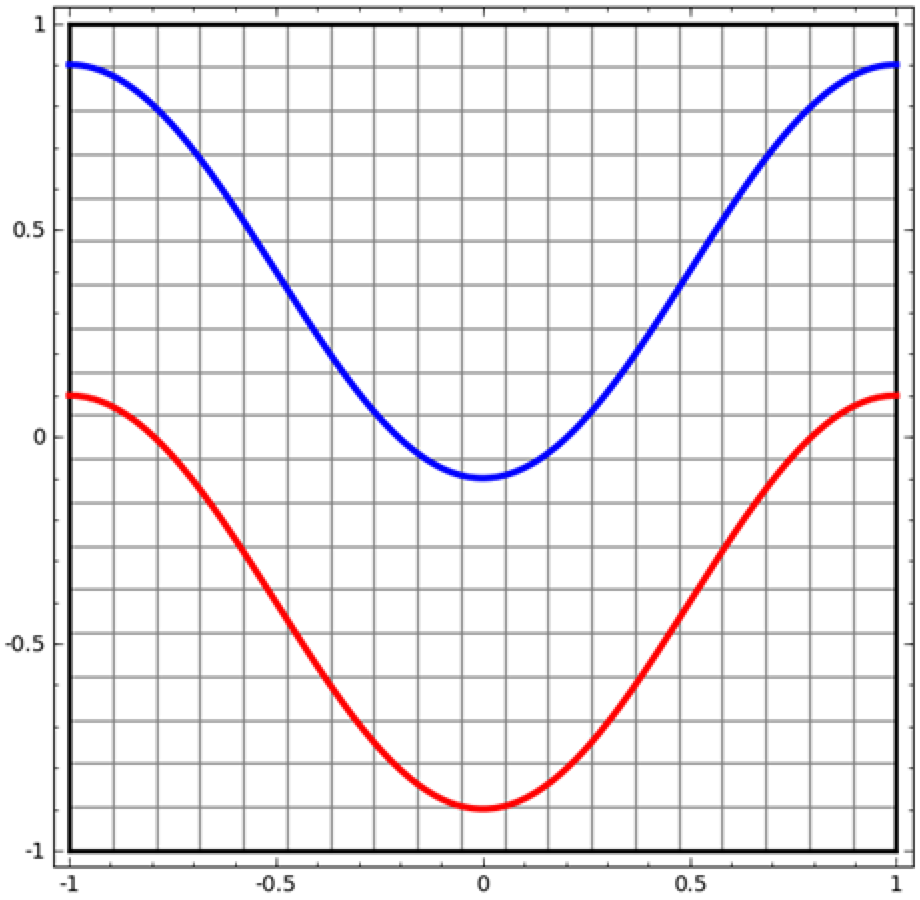


Without Activation function：



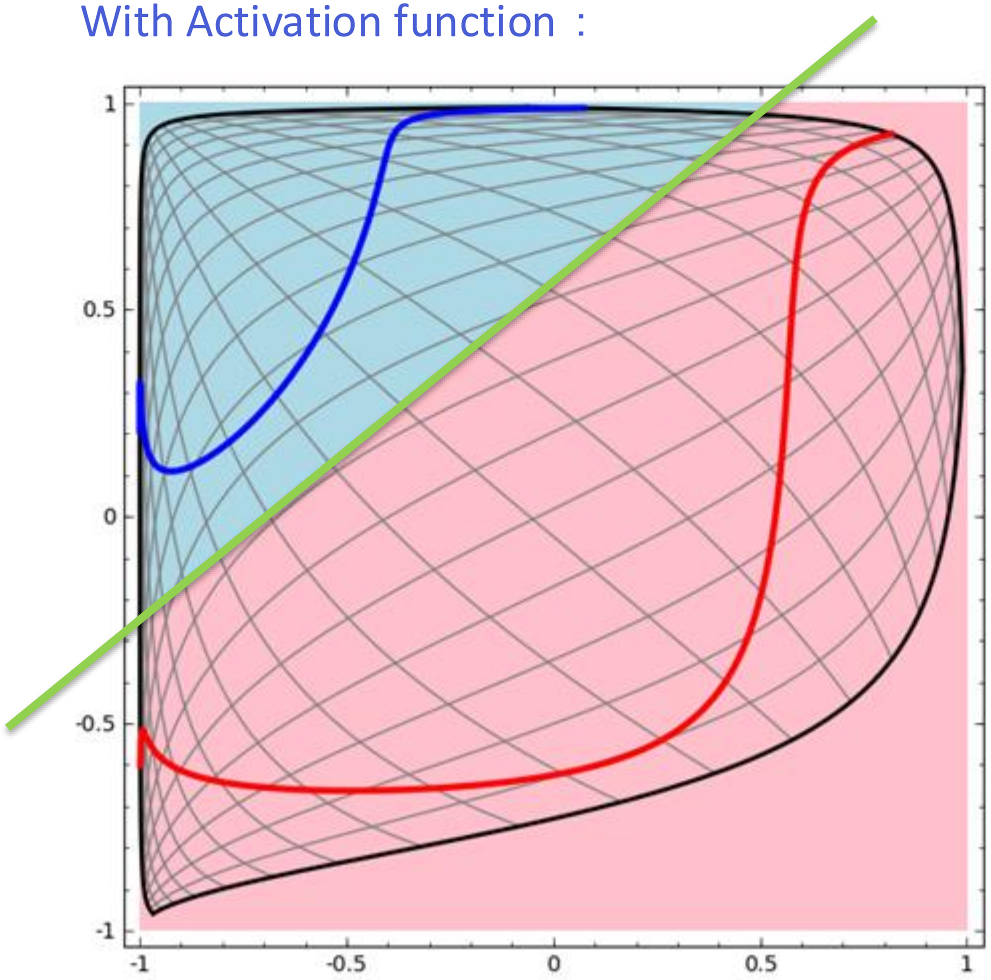
激勵函數

Linear non-separability：



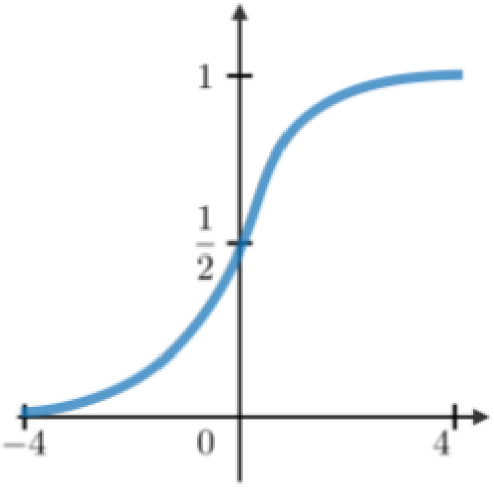
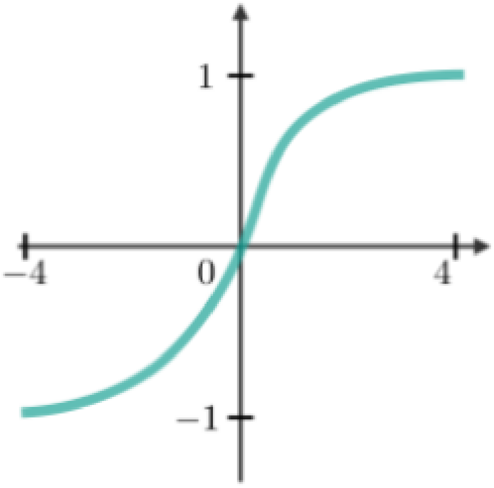
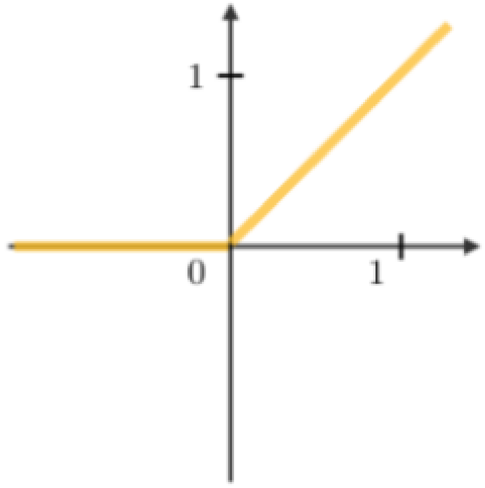
將非線性的性質引入神經網路

With Activation function：



激勵函數

$\hat{y} = f(\mathrm{W}^{\mathrm{T}}\mathrm{X} + b)$：帶入非線性關係

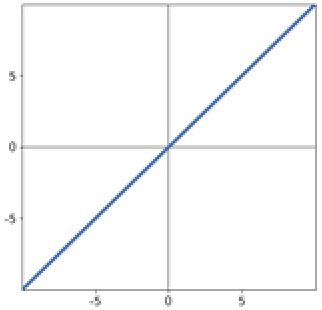
Sigmoid 55	Tanh 55	ReLU 55
$f(z) = \dfrac{1}{1+e^{-z}}$ 1	$f(z) = \dfrac{e^z - e^{-z}}{e^z + e^{-z}}$ 1	$f(z) = \max(0, z)$
		

資料來源：cs231n, Moradnejad

激勵函數

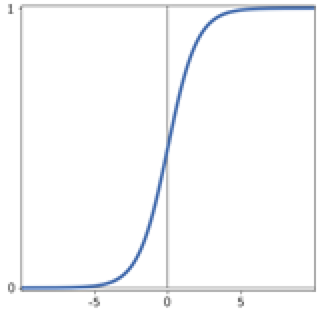
Linear

$$f(x) = x$$



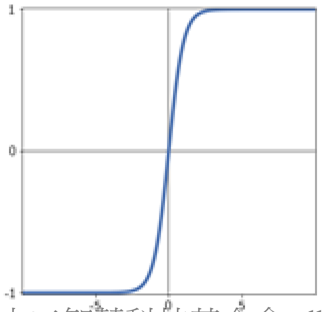
Sigmoid

$$f(x) = \frac{1}{1 + e^{-x}}$$



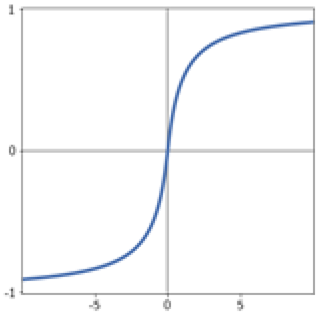
tanh

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$



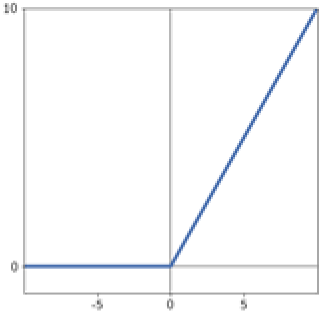
Softsign

$$f(x) = \frac{x}{1 + |x|}$$



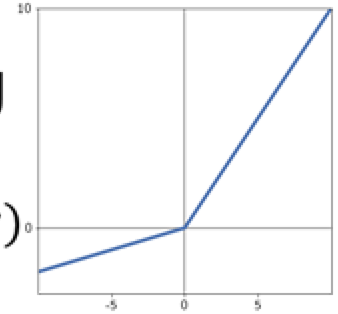
ReLU

$$f(x) = \max(0, x)$$



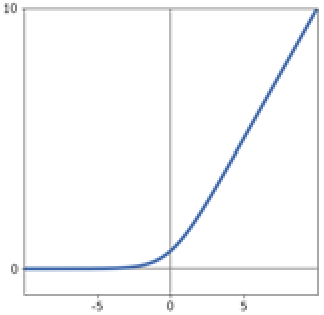
Leaky ReLU

$$f(x) = \max(ax, x)$$



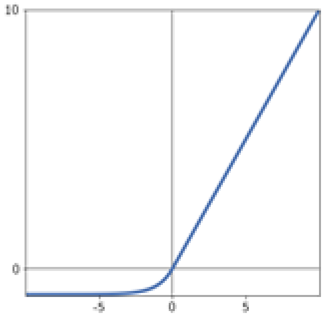
Softplus

$$f(x) = \frac{1}{\beta}\log(1 + e^{\beta x})$$



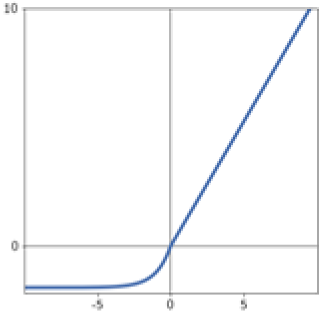
ELU

$$f(x) = \begin{cases} x & x \geq 0 \\ \alpha(e^{x} - 1) & x < 0 \end{cases}$$



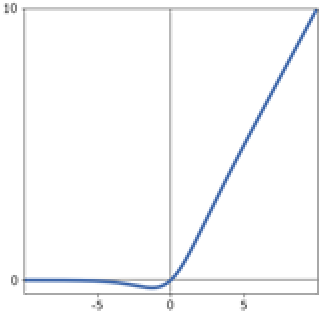
SELU

$$f(x) = \begin{cases} \lambda x & x \geq 0 \\ \lambda\alpha(e^{x} - 1) & x < 0 \end{cases}$$



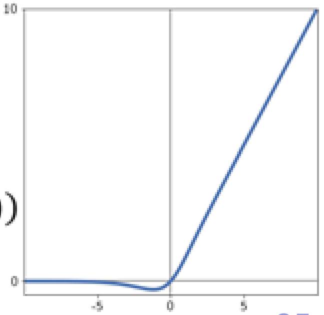
Swish

$$f(x) = \frac{x}{1 + e^{-x}}$$



Mish

$$f(x) = x \cdot \tanh(softplus(x))$$



矩陣乘法小練習

$$\begin{bmatrix} 5 & 2 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 3 \\ 3 \end{bmatrix}$$

$$= 5 \times (-1) + 2 \times 3 + 1 \times 3$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} 1 \\ \text{[illegible]} \\ 3 \end{bmatrix}$$

$$= \begin{bmatrix} 14 \\ 32 \\ 50 \end{bmatrix}$$

矩陣乘法小練習

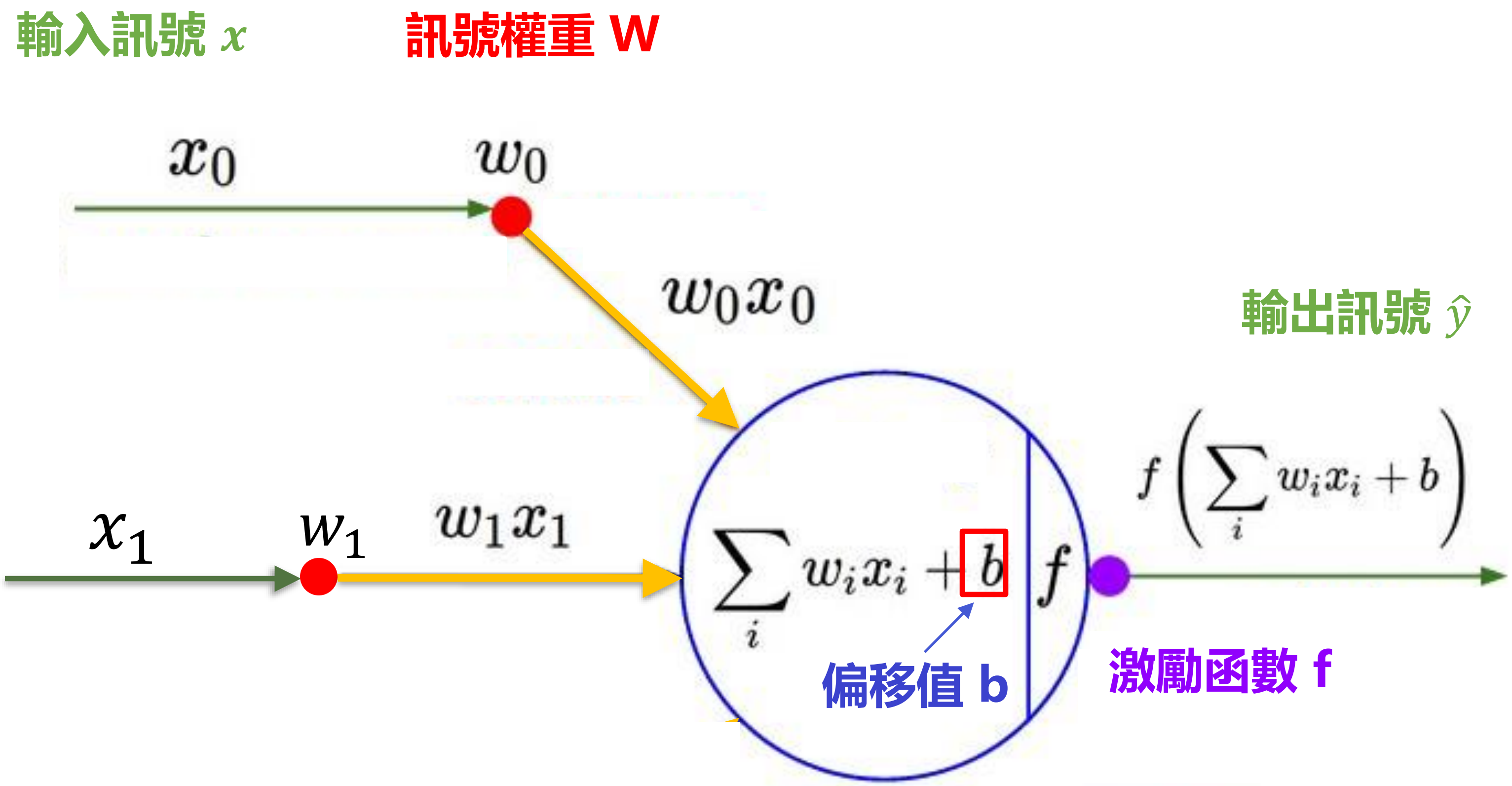
$$h_1 = \begin{bmatrix} 5 & 2 \end{bmatrix} \begin{bmatrix} -0.5 & 0.2 & -0.1 \\ 0.5 & 0.4 & -0.2 \end{bmatrix} + \begin{bmatrix} -1.5 & 0.2 & 0.8 \end{bmatrix}$$

P

$$= \begin{bmatrix} \underline{-1.5} & \underline{1.8} & \underline{-0.9} \end{bmatrix} + \begin{bmatrix} -1.5 & 0.2 & 0.8 \end{bmatrix}$$

$$= \begin{bmatrix} -3 & 2 & -0.1 \end{bmatrix}$$

神經元計算



$$\hat{y} = f(w_0 x_0 + w_1 x_1 + b)$$

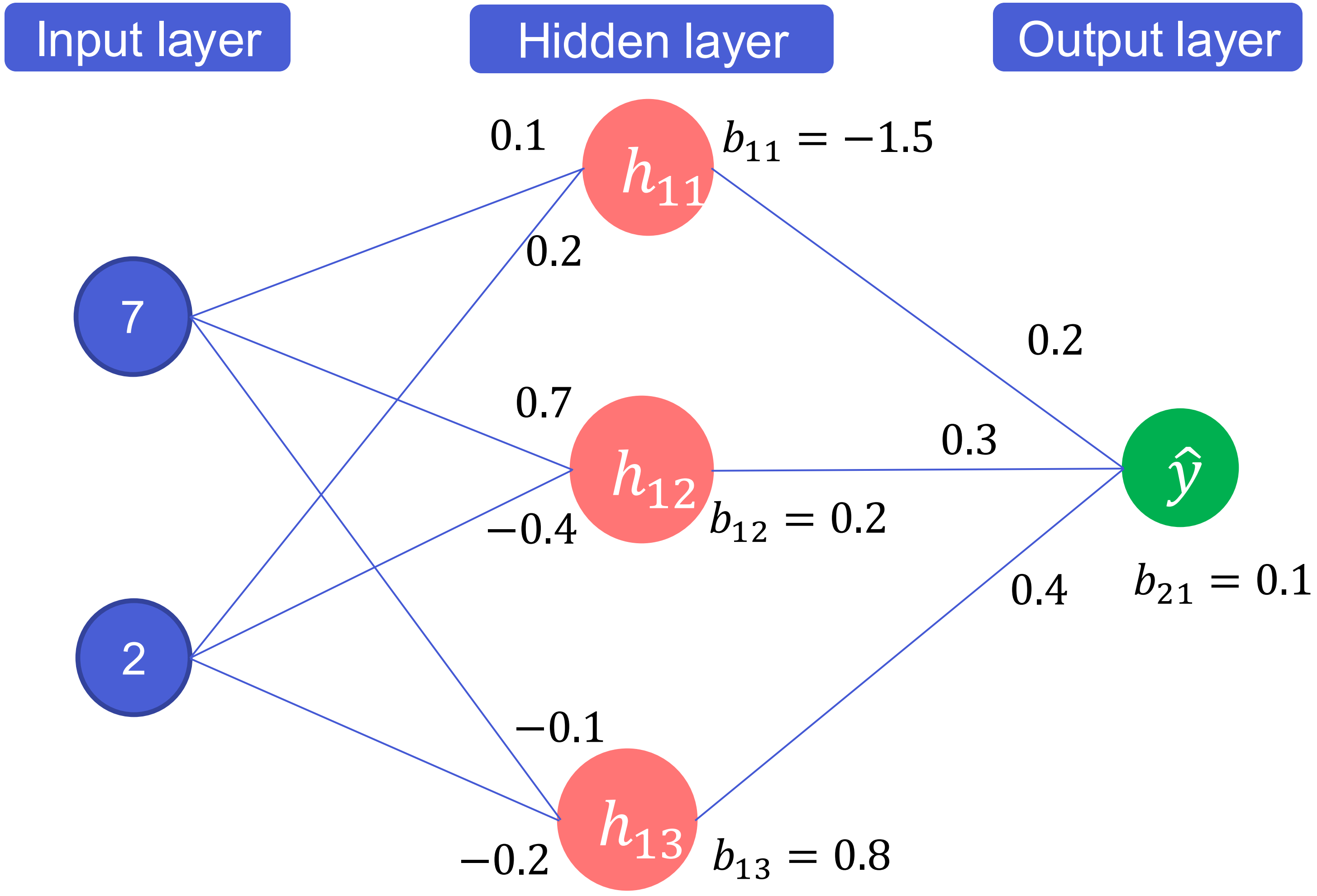
$$= f\left(\begin{bmatrix} w_0 \\ w1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} + b\right)$$

$$= f\left(\begin{bmatrix} w_0 & w_1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} + b\right)$$

$$= f(\mathrm{W}^{\mathrm{T}}\mathrm{X} + b)$$

資料來源：cs231n

多層感知器(MLP)

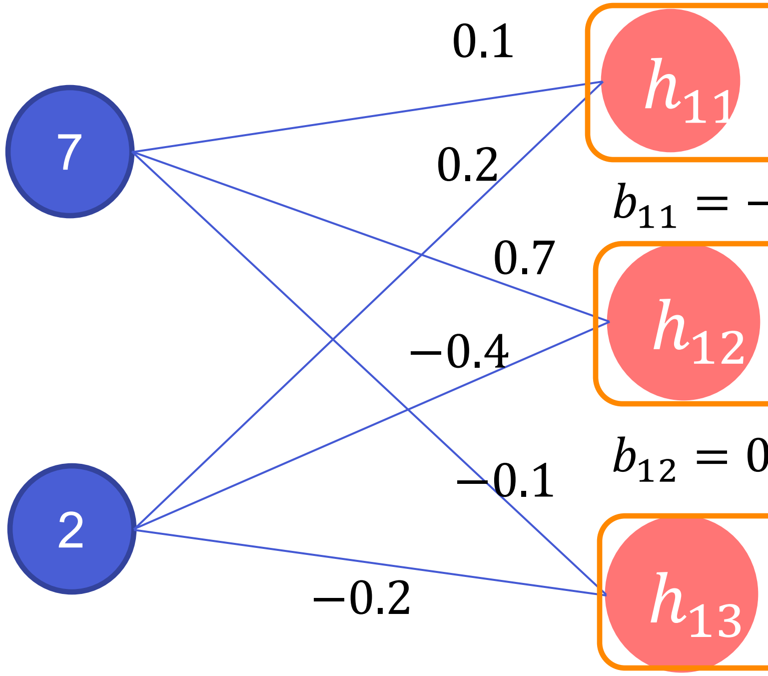


Input layer
Hidden layer
Output layer
7
2
h_{11}
h_{12}
h_{13}
$\hat{y}$
0.1
0.2
0.7
−0.4
−0.1
−0.2
$b_{11} = -1.5$
$b_{12} = 0.2$
$b_{13} = 0.8$
0.2
0.3
0.4
$b_{21} = 0.1$

多層感知器運作方式

Input layer

Hidden layer



$$h_{11} = f\big((0.1 \times 7) + (0.2 \times 2)) + (-1.5)\big) = -0.4$$

$b_{11} = -1.5$

$$h_{12} = f\big((0.7 \times 7) + (-0.4 \times 2)) + (0.2)\big) = 4.3$$

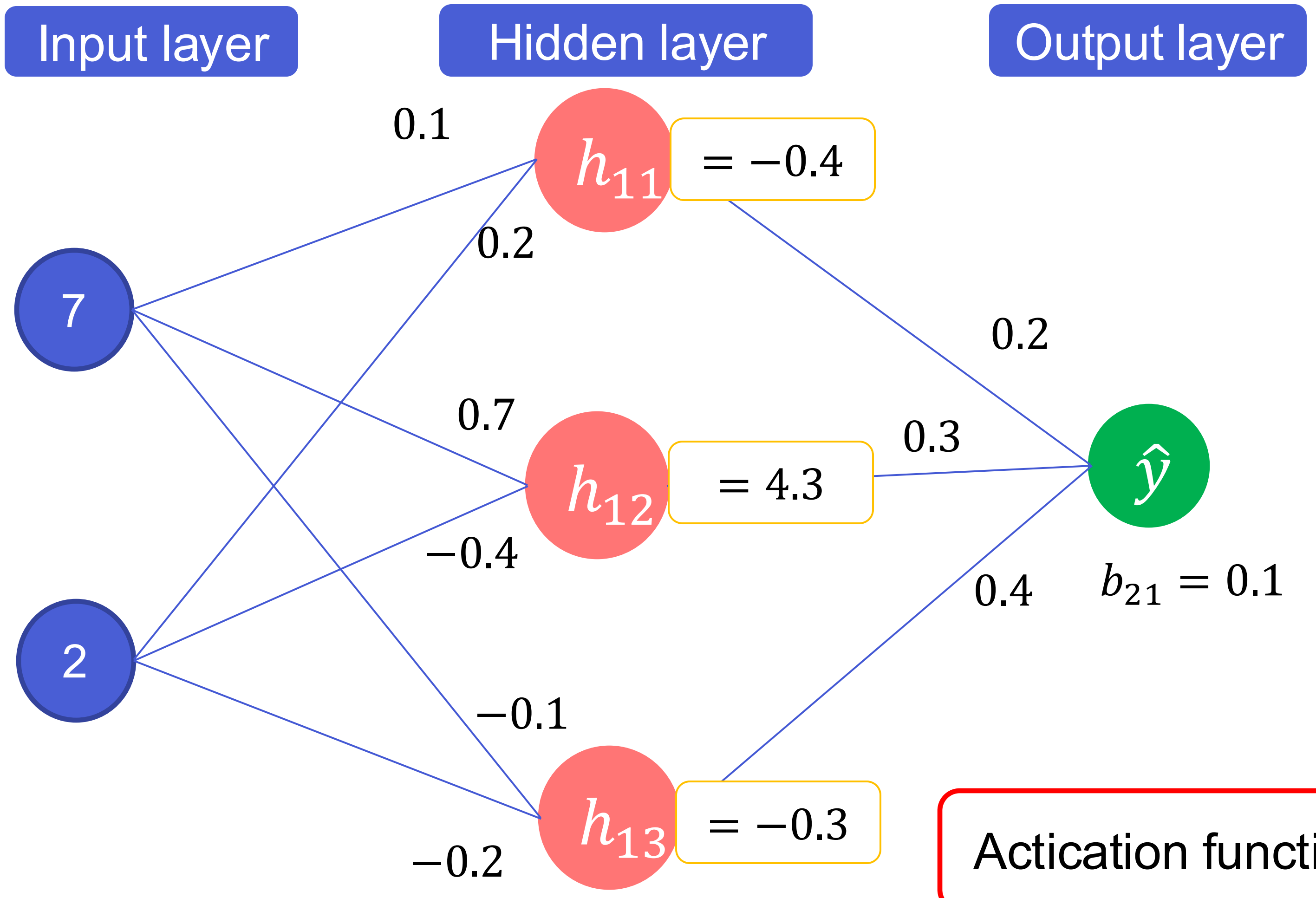
$b_{12} = 0.2$

$$h_{13} = f\big((-0.1 \times 7) + (-0.2 \times 2)) + (0.8)\big) = -0.3$$

$b_{13} = 0.8$

Actication function $f(z)$ is lInear function: $f(z) = z$

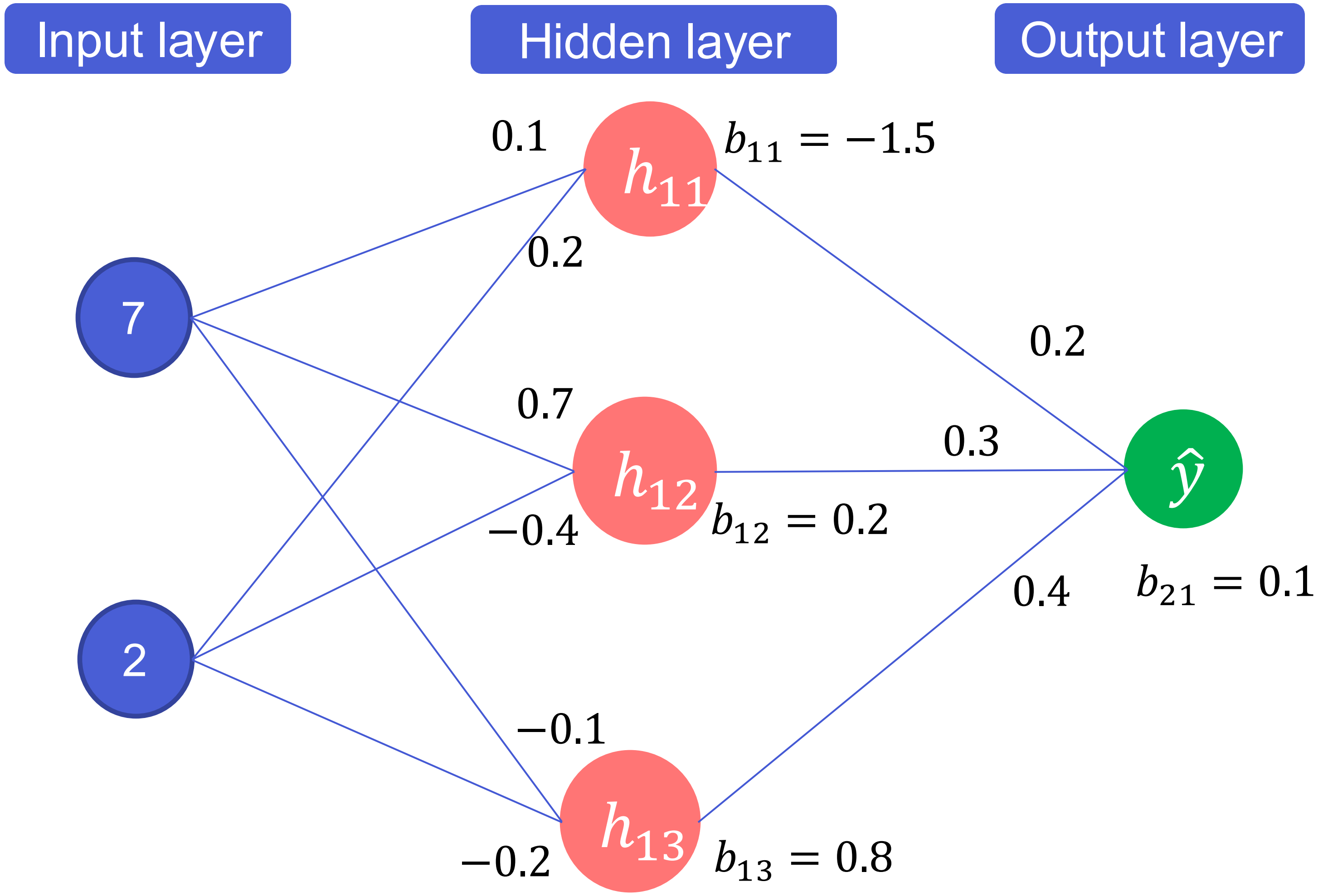
多層感知器運作方式



$$= f\Big(\big(0.2 \times (-0.4)\big) + (0.3 \times 4.3) + (0.4 \times (-0.3)) + (0.1)\Big) = 1.19$$

Actication function $f(z)$ is lInear function: $f(z) = z$

多層感知器運作方式



Input layer:

$$X = \begin{bmatrix} 7 \\ 2 \end{bmatrix} \in R^{2\times1}$$

Hidden layer:

$$W_1 = \begin{bmatrix} 0.1 & 0.7 & -0.1 \\ 0.2 & -0.4 & -0.2 \end{bmatrix} \in R^{2\times3}$$

$$b_1 = \begin{bmatrix} -1.5 \\ 0.2 \\ 0.8 \end{bmatrix} \in R^{3\times1}$$

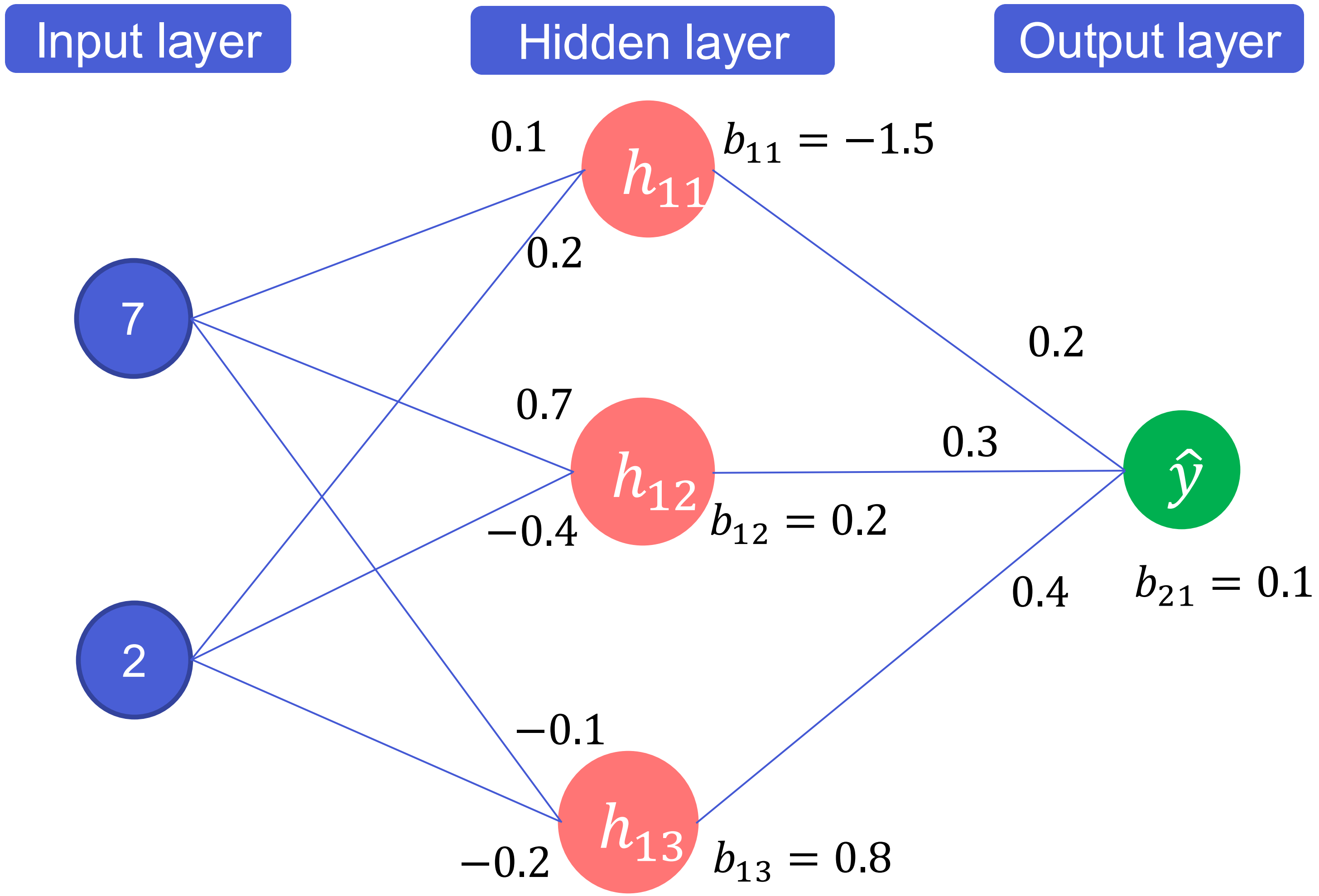
Output layer:

$$W_2 = \begin{bmatrix} 0.2 \\ 0.3 \\ 0.4 \end{bmatrix} \in R^{3\times1}$$

$$b_2 = [0.1] \in R^1$$

$$\hat{y} = f_2(W_2^T(f_1(W_1^T X + b_1)) + b_2)$$

多層感知器運作方式



$$\hat{y} = f_2(W_2^T(f_1(W_1^T X + b_1)) + b_2)$$

Hidden layer output:

$$h = \begin{bmatrix} 0.1 & 0.2 \\ 0.7 & -0.4 \\ -0.1 & -0.2 \end{bmatrix} \begin{bmatrix} 7 \\ 2 \end{bmatrix} + \begin{bmatrix} -1.5 \\ 0.2 \\ 0.8 \end{bmatrix}$$

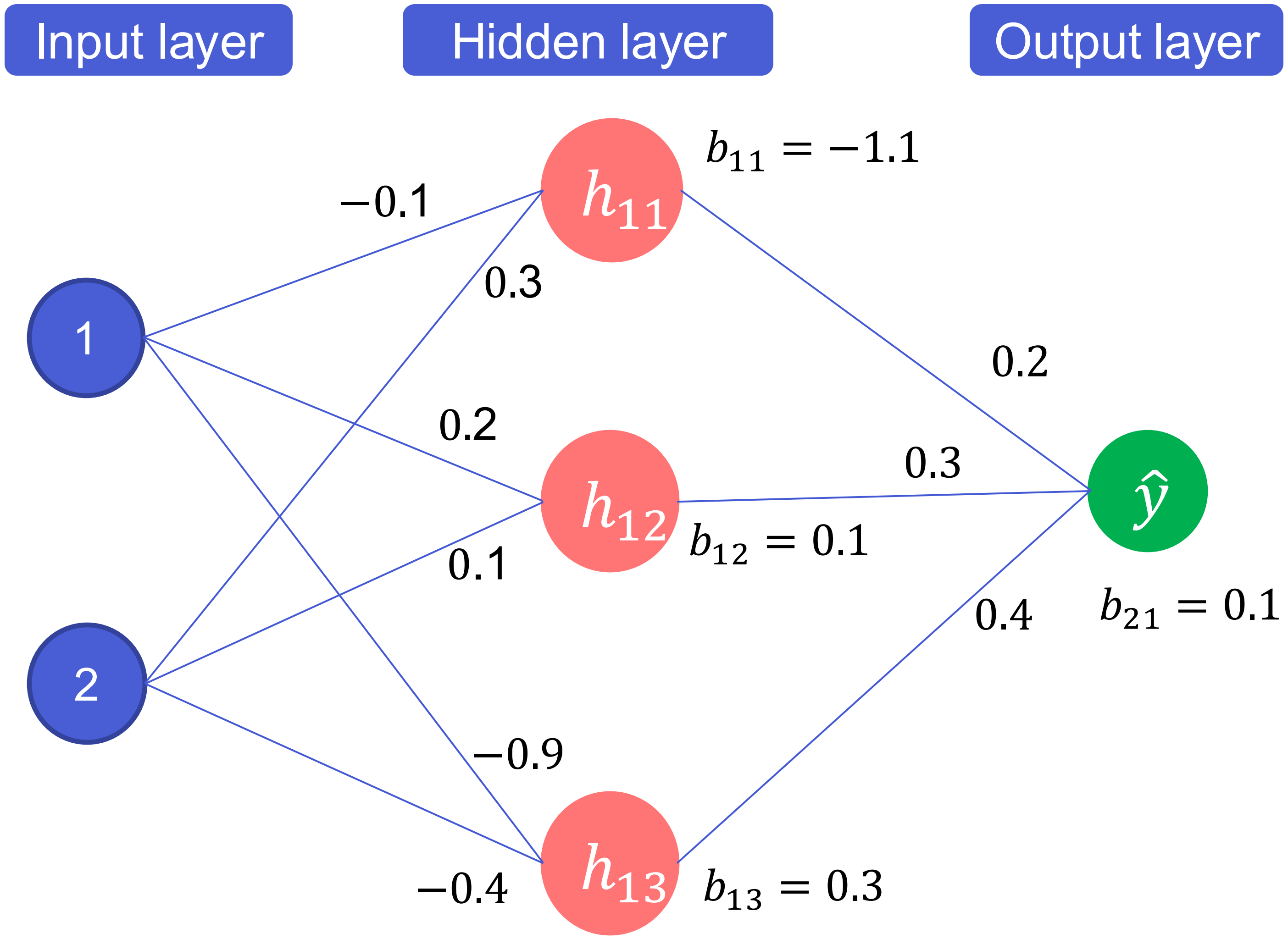
$$= \begin{bmatrix} 1.1 \\ 4.1 \\ 0.1 \end{bmatrix} + \begin{bmatrix} -1.5 \\ 0.2 \\ 0.8 \end{bmatrix} = \begin{bmatrix} -0.4 \\ 4.3 \\ -0.3 \end{bmatrix}$$

Output layer:

$$\hat{y} = \begin{bmatrix} 0.2 & 0.3 & 0.4 \end{bmatrix} \begin{bmatrix} -0.4 \\ 4.3 \\ -0.3 \end{bmatrix} + [0.1]$$

$$\hat{y} = 1.19$$

多層感知器- Quiz1



$$\hat{y} = f_2(\mathrm{W}_2^{\mathrm{T}} \boxed{f_1(\mathrm{W}_1^{\mathrm{T}}\mathrm{X} + b_1)} + b_2)$$

Quiz 1：請問 $\boxed{f_1(\mathrm{W}_1^{\mathrm{T}}\mathrm{X} + b_1)}$ 是多少？

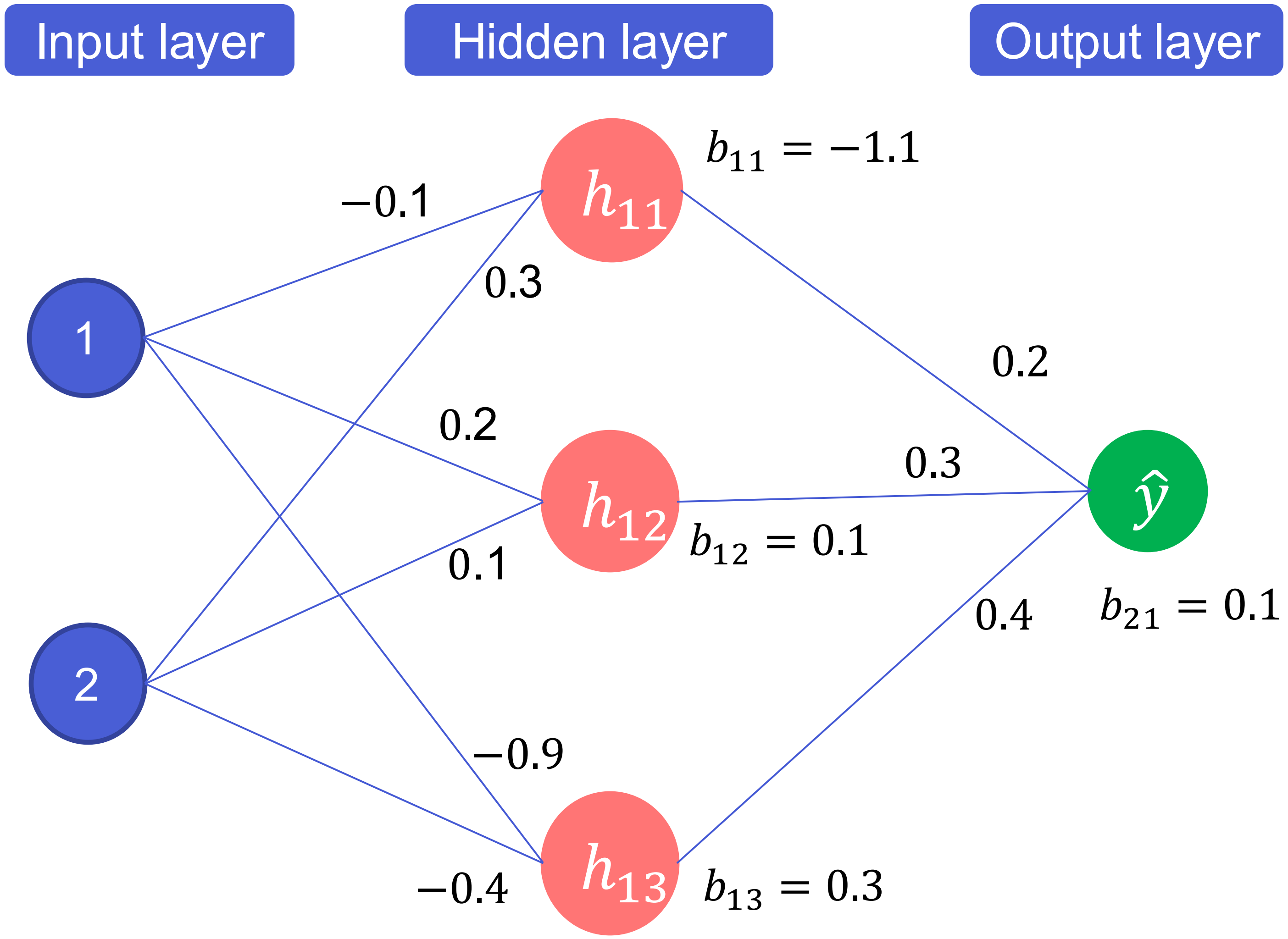
Activation Function 是 ReLU：f_1(z)=max(0,z)

$$h_1 = \begin{bmatrix} -0.1 & 0.3 \\ 0.2 & 0.1 \\ -0.9 & -0.4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} -1.1 \\ 0.1 \\ 0.3 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 \\ 0.4 \\ -1.7 \end{bmatrix} + \begin{bmatrix} -1.1 \\ 0.1 \\ 0.3 \end{bmatrix} = \begin{bmatrix} -0.6 \\ 0.5 \\ -1.4 \end{bmatrix}$$

$$f_1(h_1) = f_1\left(\begin{bmatrix} -0.6 \\ 0.5 \\ -1.4 \end{bmatrix}\right) = \begin{bmatrix} 0.0 \\ 0.5 \\ 0.0 \end{bmatrix}$$

多層感知器- Quiz2



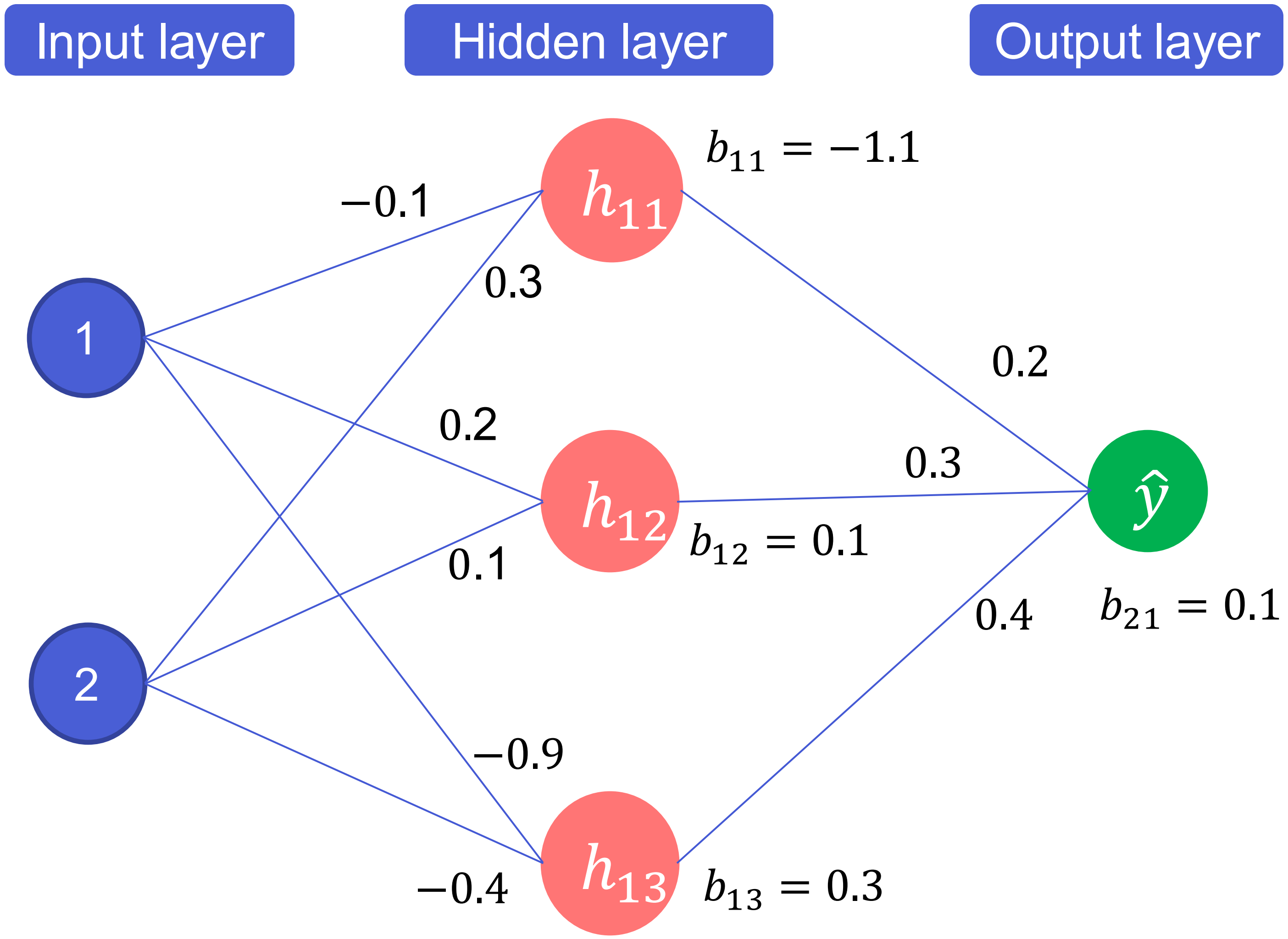
$$\boxed{\hat{y}} = f_2(\mathrm{W}_2^{\mathrm{T}}(f_1(\mathrm{W}_1^{\mathrm{T}}\mathrm{X} + b_1)) + b_2)$$

Quiz 2:請問 $\boxed{\hat{y}}$ 是多少？

Activation Function 是 Linear : f(z)=z

$$\hat{y} = [0.2 \quad 0.3 \quad 0.4]\begin{bmatrix}0.0\\0.5\\0.0\end{bmatrix} + [0.1] = 0.25$$

多層感知器應用



$$\hat{y} = f_2(W_2^T(f_1(W_1^T X + b_1)) + b_2)$$

$$\hat{y} = 0.25$$

迴歸問題，直接將 $\hat{y}$ 作為預測值：

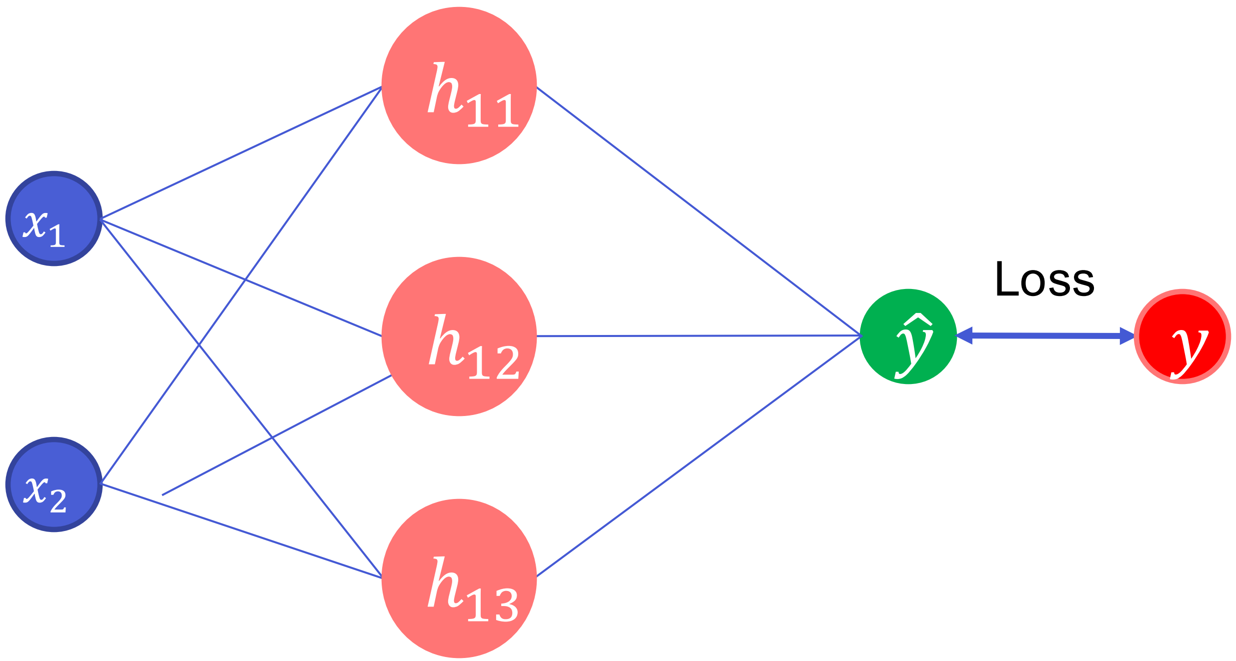
- 明天股價漲 0.25
- 房價漲了 25%
- …

分類問題，決定一個 Threshold 進行分類：

$$\hat{y} = f(x) = \begin{cases} 0, & \sum_i w_i x_i < threshold \text{ 閾值} \\ 1, & \sum_i w_i x_i \geq threshold \text{ 閾值} \end{cases}$$

迴歸損失函數 (Loss function、Cost function)

計算預測值 ŷ 與 標籤 y 之間的差距。



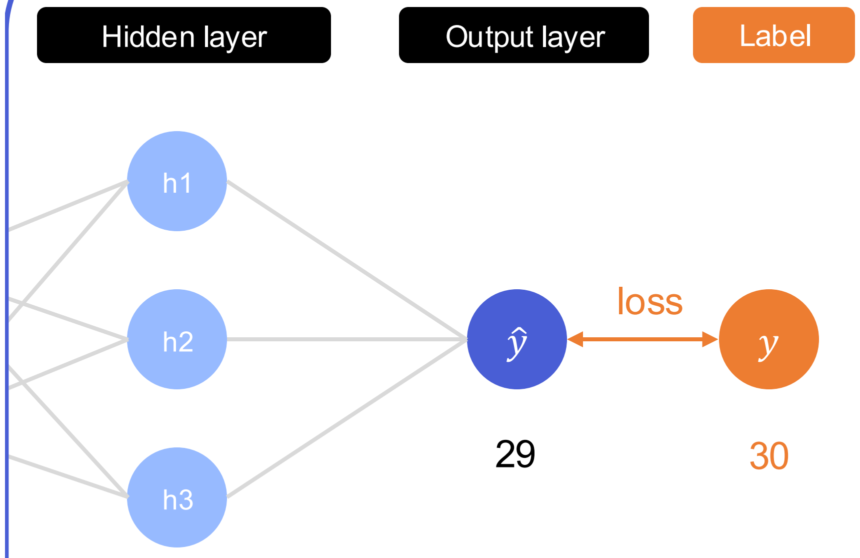
(根據預測的目標有不同的公式)

迴歸問題

- 均方差 (Mean Square Error, MSE)
- 平均絕對差 (Mean Absolute Error, MAE)

損失函數 (迴歸問題)

迴歸問題



$MSE = (30-29)^2$ $\quad$ $MAE = |30-29|$

Loss function

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

範例:

data	output
1	29
2	53
3	98

loss

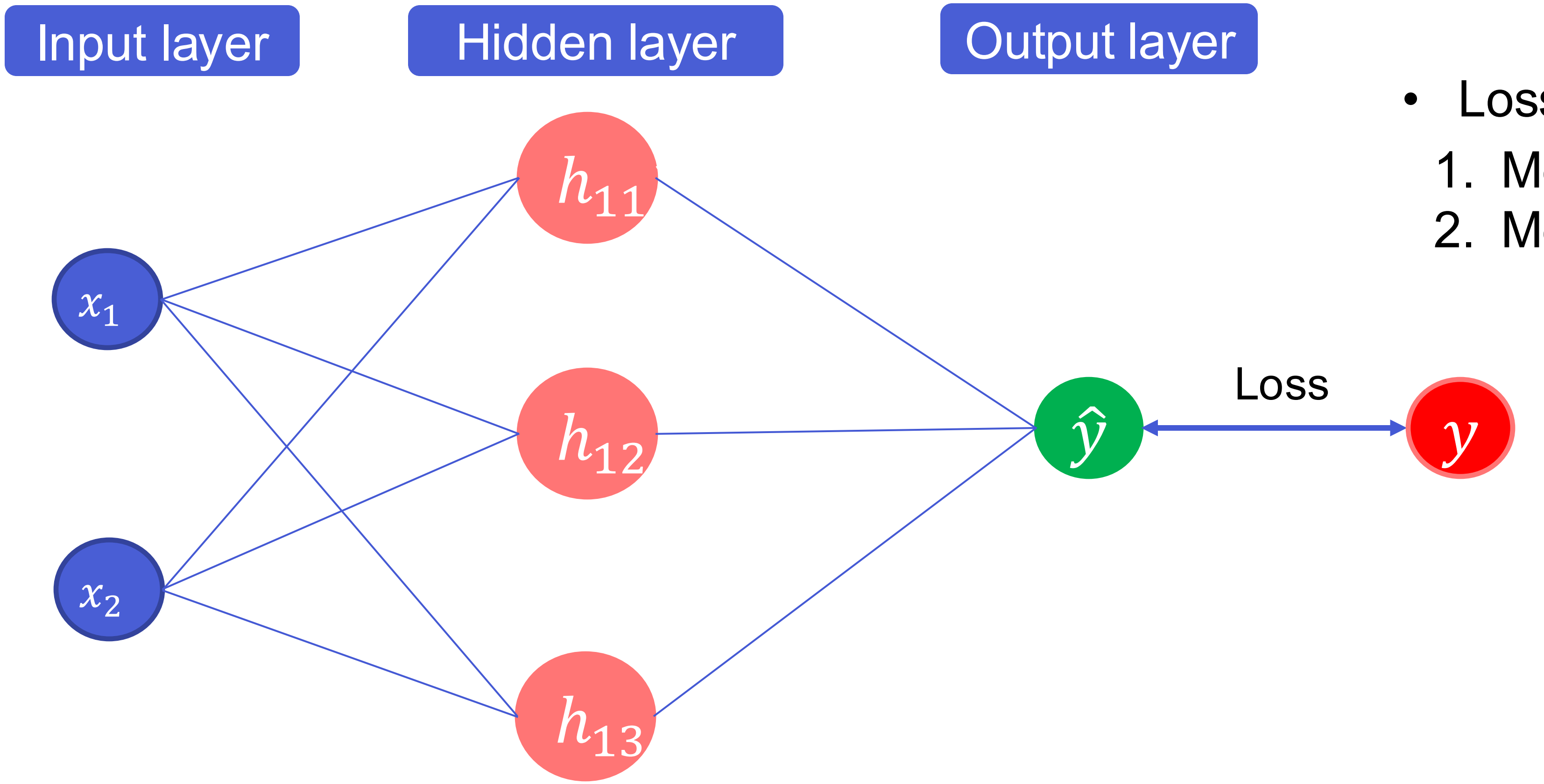
data	label
1	30
2	50
3	100

$$MSE = \frac{(30-29)^2 + (50-53)^2 + (100-98)^2}{3}$$

$$MAE = \frac{|30-29| + |50-53| + |100-98|}{3}$$

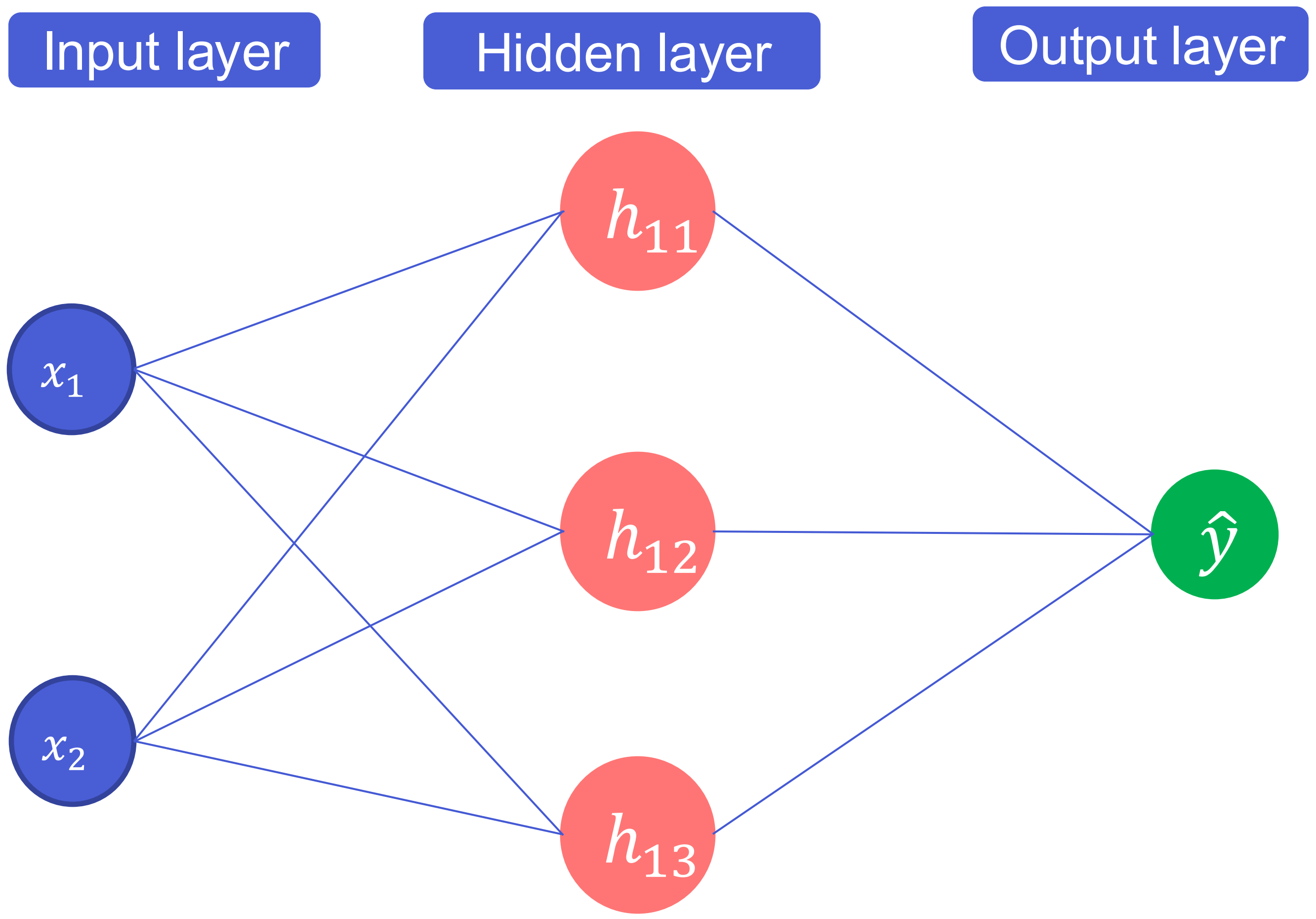
迴歸問題

- Output layer
  1. 一顆神經元
  2. 使用 Linear 的 Activation Function



- Loss function:
  1. Mean square error (MSE)
  2. Mean absolute error (MAE)

多層感知器應用



$$\hat{y} = f_2(\mathrm{W}_2^{\mathrm{T}}(f_1(\mathrm{W}_1^{\mathrm{T}}\mathrm{X} + b_1)) + b_2)$$

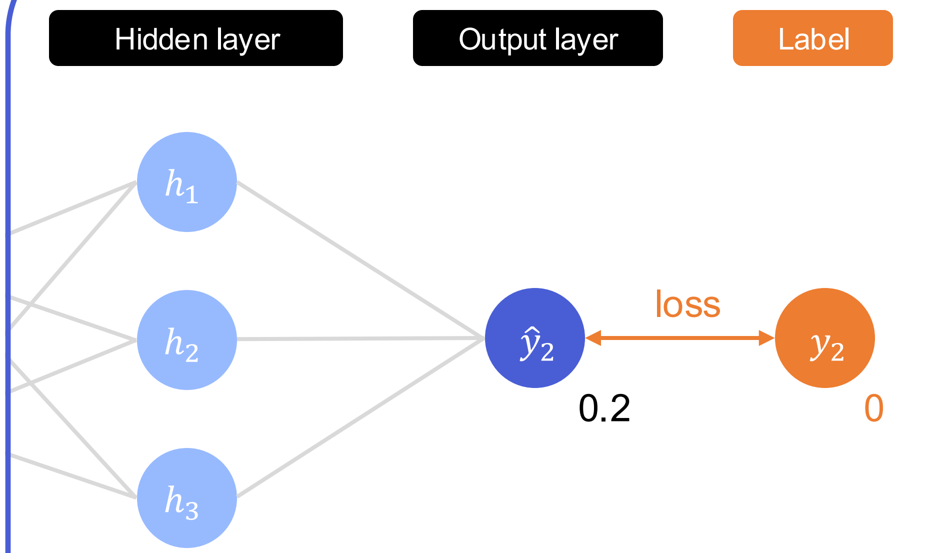
分類問題，決定一個 Threshold 進行分類：

二元分類

$$\hat{y} = f(x) = \begin{cases} 0, & \sum_i w_i x_i < threshold \text{ 閾值} \\ 1, & \sum_i w_i x_i \geq threshold \text{ 閾值} \end{cases}$$

損失函數 (分類問題)

分類問題



$$Binary\ Cross\ Entropy = -\log(1-0.2)$$

Loss function

$$Binary\ Cross\ Entropy = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i))$$

BCE 有兩個模式，label = 1 的時候會計算 $\log \hat{y}_i$ ，label = 0 的時候會計算 $\log(1-\hat{y}_i)$ 。

範例:

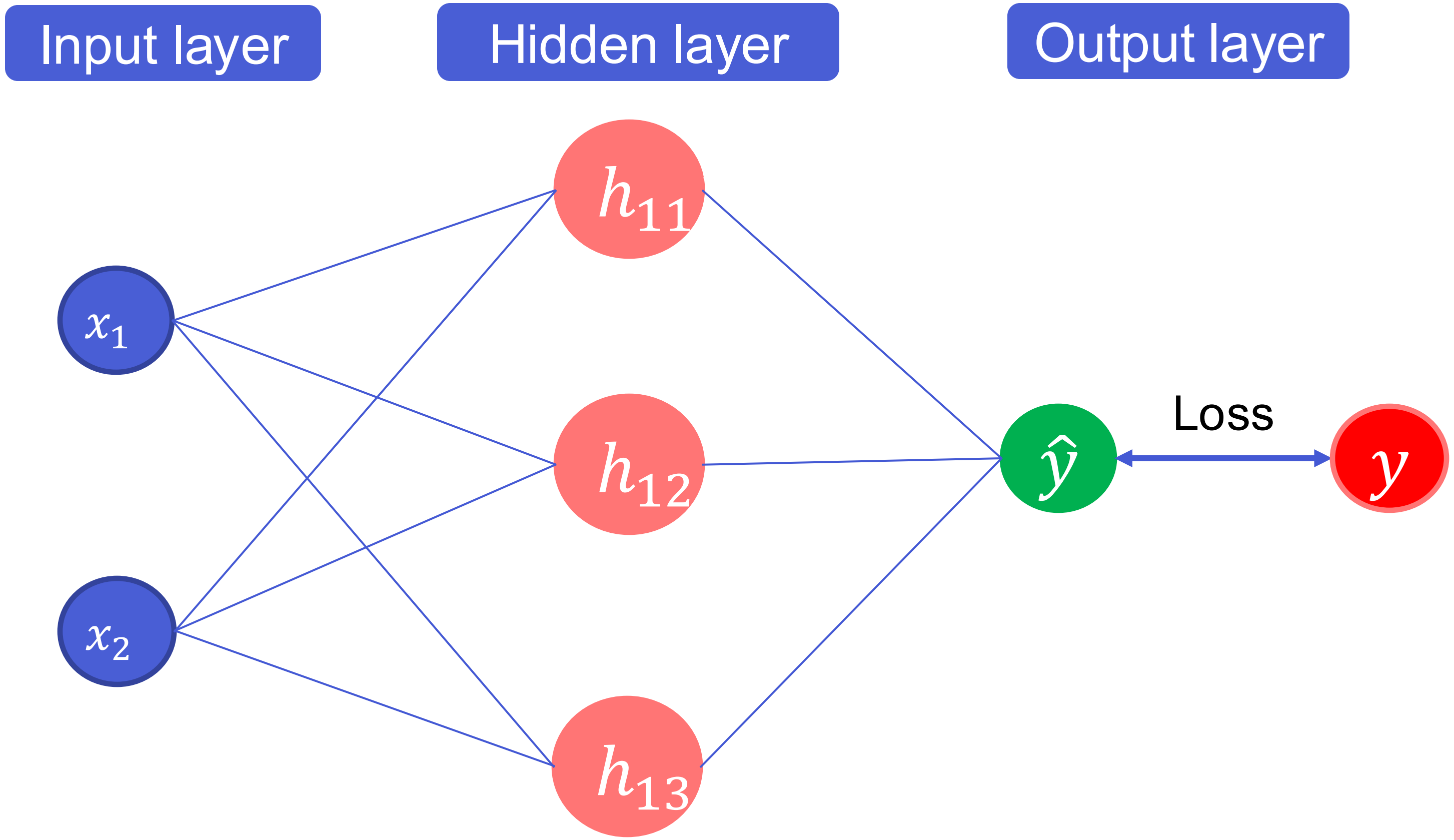
data	output
1	0.2
2	0.9
3	0.3

loss

data	label
1	0
2	1
3	0

$$Binary\ Cross\ Entropy = -\frac{\log(1-0.2) + log0.9 + \log(1-0.3)}{3}$$

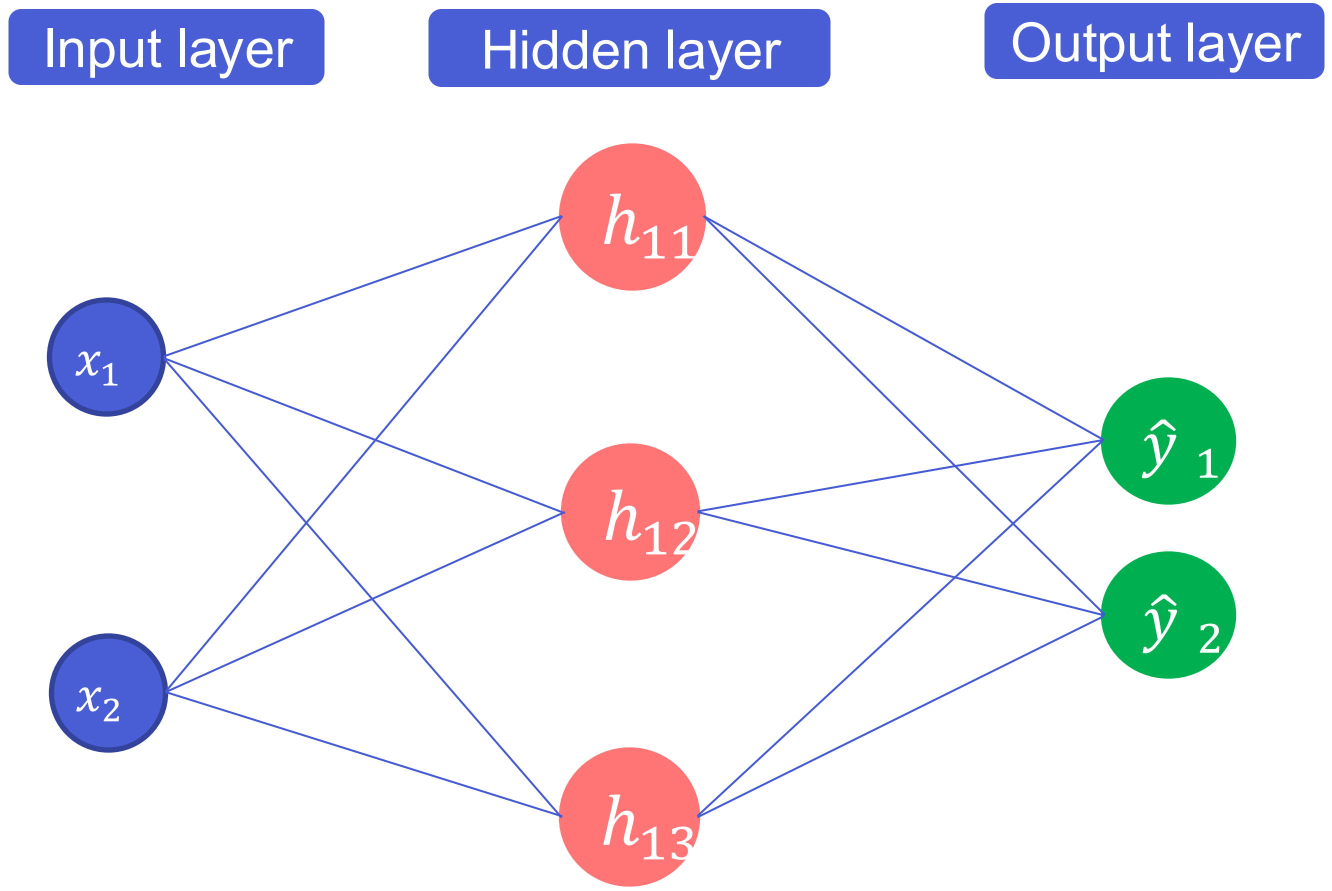
分類問題(二元分類)



- Output layer
  1. 一顆神經元
  2. 使用 Sigmoid Function

- Loss function:
  1. Binary Cross Entropy (BCE)

多層感知器應用



$$\hat{y} = f_2(\mathrm{W}_2^{\mathrm{T}}(f_1(\mathrm{W}_1^{\mathrm{T}}\mathrm{X} + b_1)) + b_2)$$

分類問題，決定一個 Threshold 進行分類：

二元分類

$$\hat{y} = f(x) = \begin{cases} 0, & \sum_i w_i x_i < threshold \text{ 閾值} \\ 1, & \sum_i w_i x_i \geq threshold \text{ 閾值} \end{cases}$$

多元分類

$$softmax(\hat{y}_j) = \frac{e^{\hat{y}_j}}{\sum_{k=1}^{n} e^{\hat{y}_k}}, j = 1, \ldots, K$$

$$\hat{y}_1 = softmax(\hat{y}_1) = \frac{e^{-1}}{e^{-1}+e^2} = \frac{0.368}{0.368+7.389} = 0.0474$$

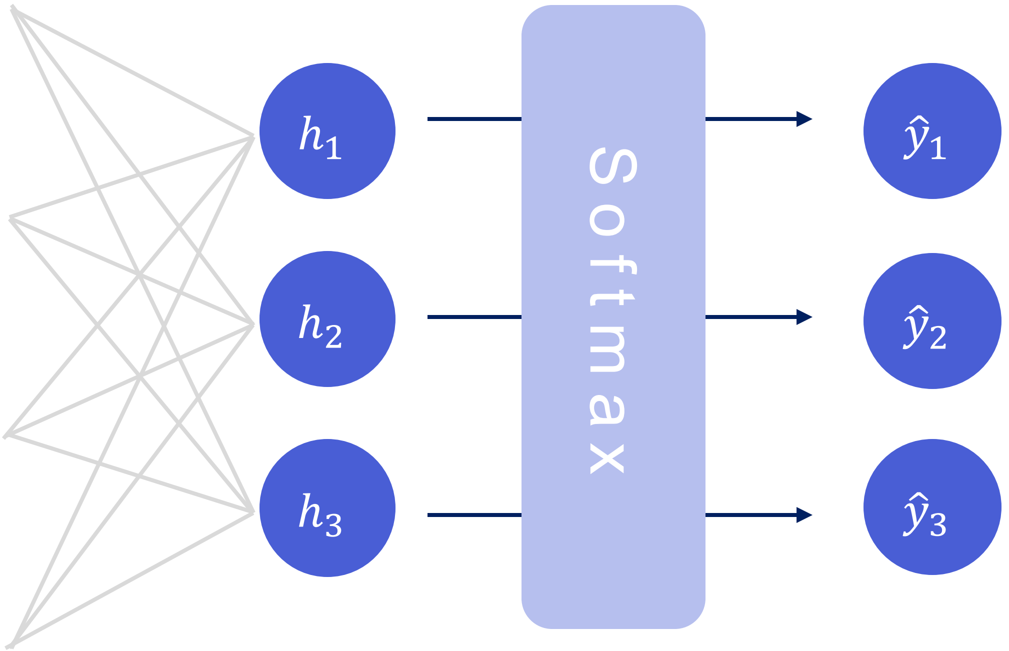
$$\hat{y}_2 = softmax(\hat{y}_2) = \frac{e^{2}}{e^{-1}+e^2} = \frac{7.389}{0.368+7.389} = 0.9526$$

Softmax (分類問題)

Softmax

$$y_i = \frac{e^{h_i}}{\sum_{j=1}^{k} e^{h_i}} \quad \text{for k classes}$$

Output layer



$$= \frac{e^{h_1}}{e^{h_1} + e^{h_2} + e^{h_3}}$$

$$= \frac{e^{h_2}}{e^{h_1} + e^{h_2} + e^{h_3}}$$

$$= \frac{e^{h_3}}{e^{h_1} + e^{h_2} + e^{h_3}}$$

類似機率的性質:

- 輸出介於 0 到 1 之間
- 加總為 1

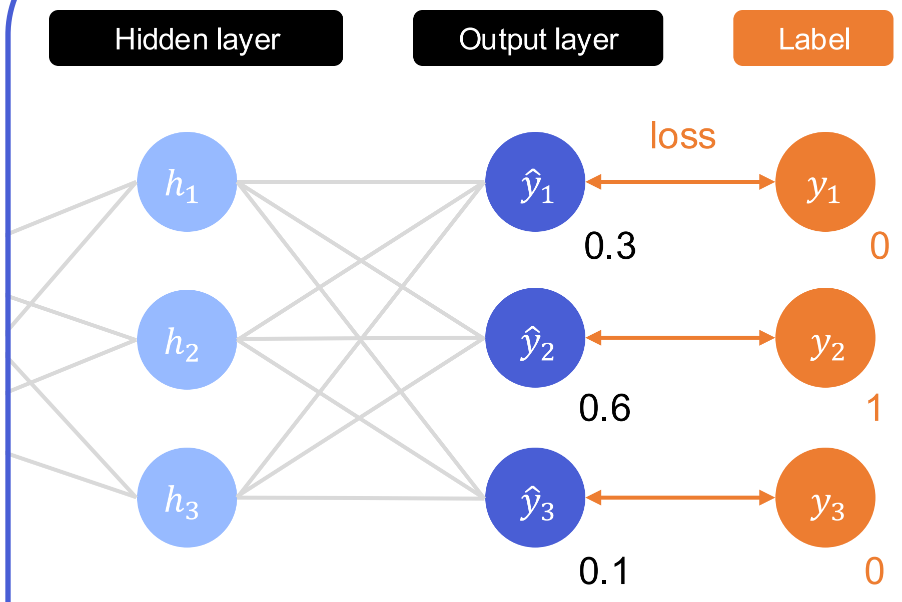
損失函數 (分類問題)

Loss function

$$Cross\ Entropy = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{k} y_i^c \cdot log\hat{y}_i^c$$

CE 只有在 label = 1 的時候才會計算 $\log \hat{y}_i$。

分類問題



$$Cross\ Entropy = -log0.6$$

範例:

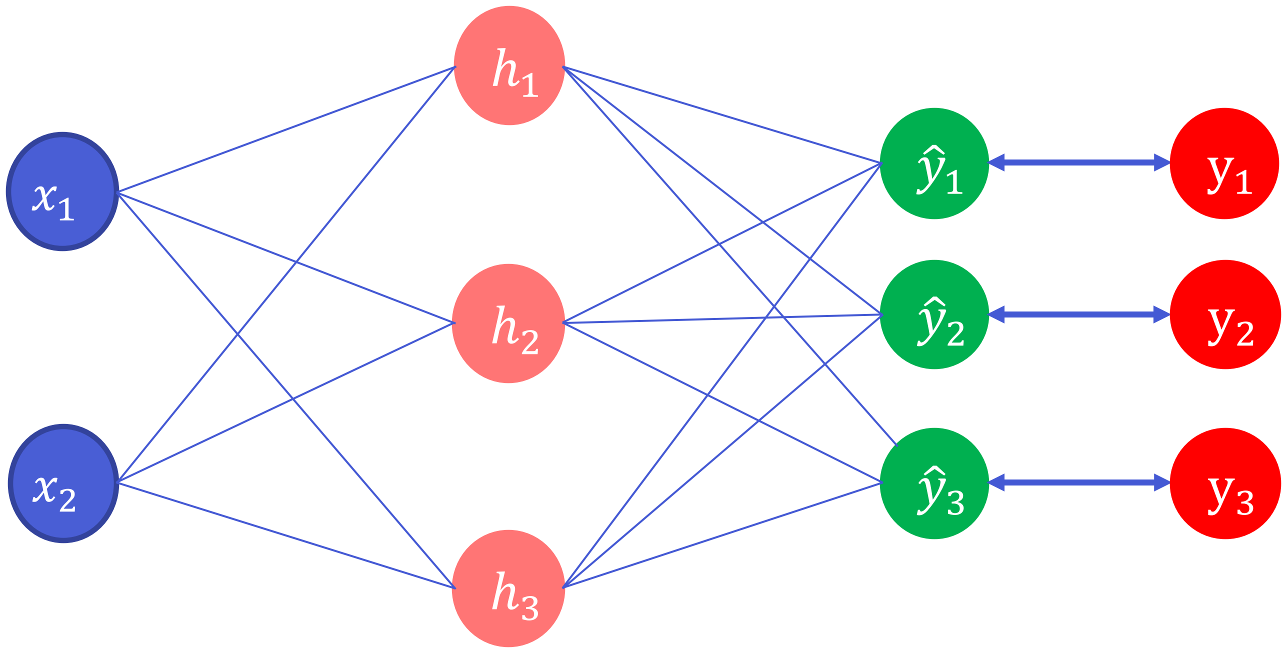
data	output		
	c1	c2	c3
1	0.3	0.6	0.1
2	0.8	0.1	0.1
3	0.2	0.3	0.5

loss

data	label		
	c1	c2	c3
1	0	1	0
2	1	0	0
3	0	0	1

$$Cross\ Entropy = -\frac{log0.6 + log0.8 + log0.5}{3}$$

分類問題(多元分類)



- Output layer
  1. n 個類別， 就要放 n 顆神經元
  2. 使用 Softmax Function
  3. 輸出為類似機率的數值，加總等於 1
- Loss function:
  1. Categorical Cross Entropy (CE)

一個類神經網路訓練及預測的流程

訓練階段

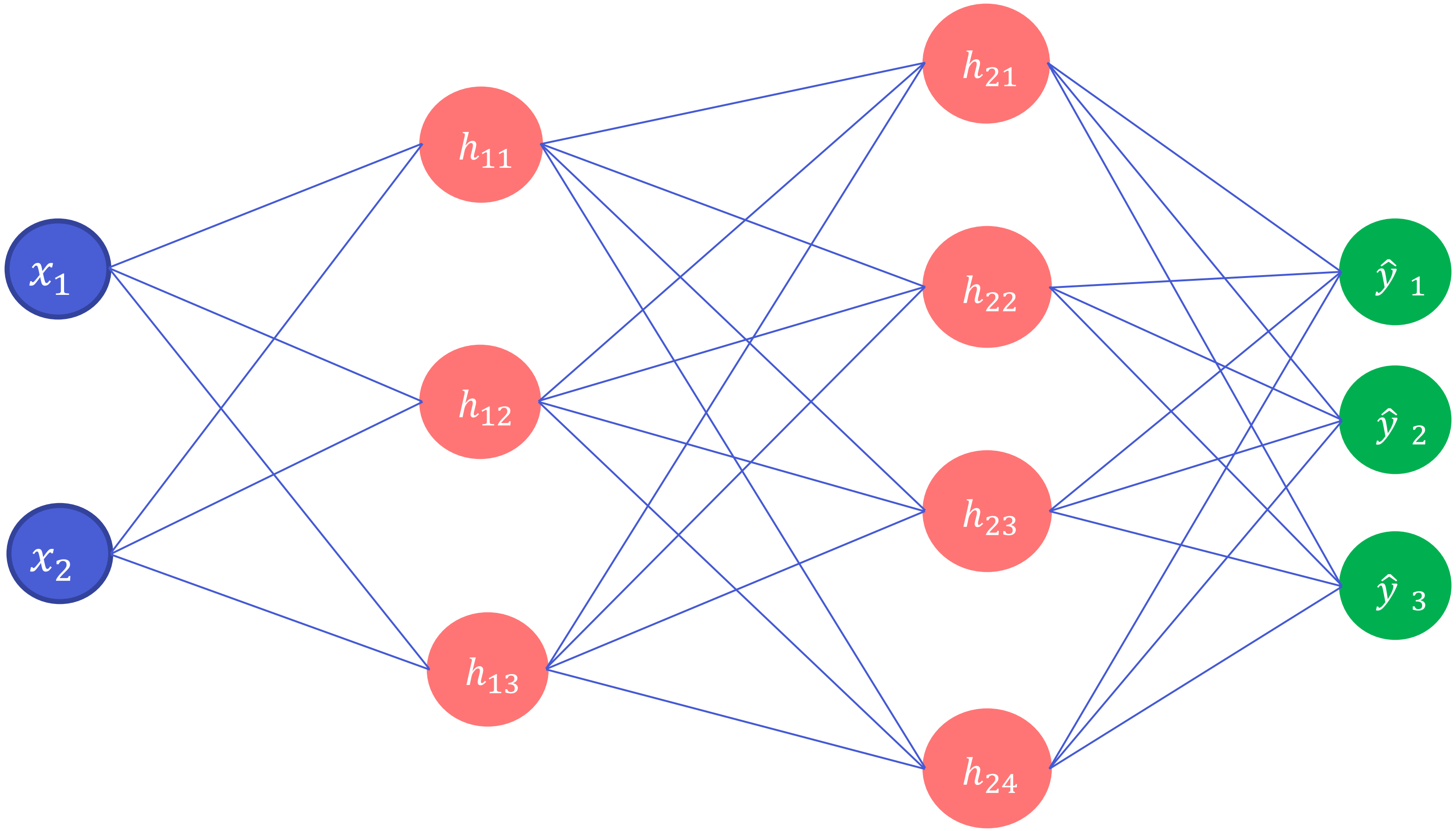
資料集做預處理後切分成訓練集與測試集 → 模型架構的設計以及激活函數的設定 → 編譯模型，包含設定loss function 及 optimizer → 開始訓練

預測階段

準備需測試的資料集並做預處理 → 準備好訓練好的模型權重 → 將訓練好的模型權重讀入 → 開始預測

實作時間

請用 numpy，實作以下的多層感知器結構



Thank you