多層感知網路 (MULTILAYER PERCEPTRON NEURAL NETWORK)

王豐緒
銘傳大學資工系

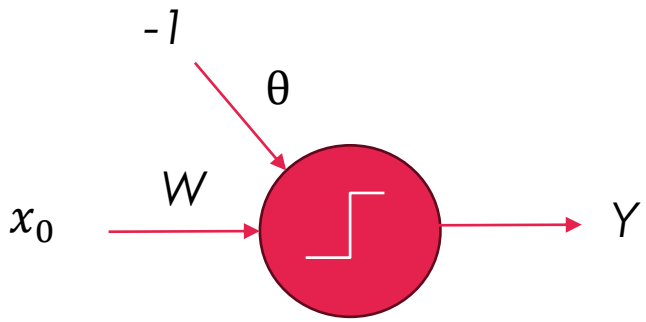
學習目標

- 理解MLP類神經模型的架構
- 理解通用逼近定理
- 理解MLP類神經應用的議題
  - 權重設定初值
  - 激活函數的選擇
  - 避免過度學習
- 熟悉MLP應用範例

大綱

- 多層感知神經網路的架構
- 網路的前向運算程序
- 通用逼近定理
- 類神經網路應用實務建議與議題
- MLP類神經網路應用

THE PERCEPTRON (複習)

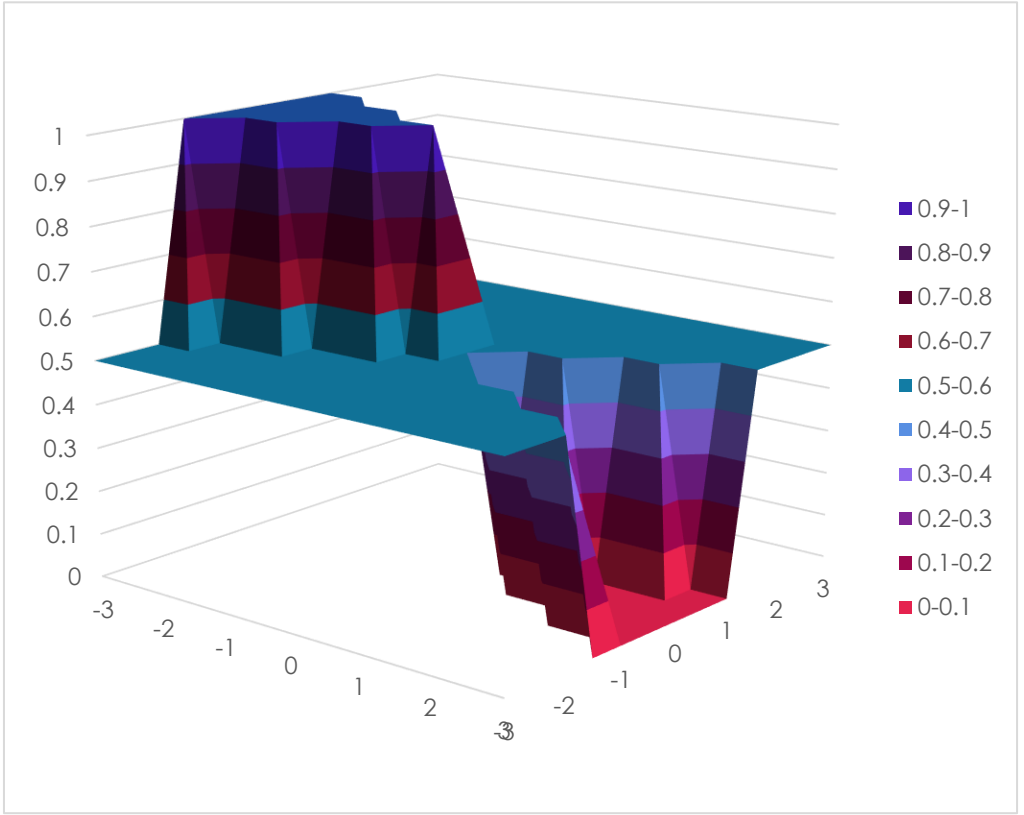


$$z = wx_0 - \theta$$

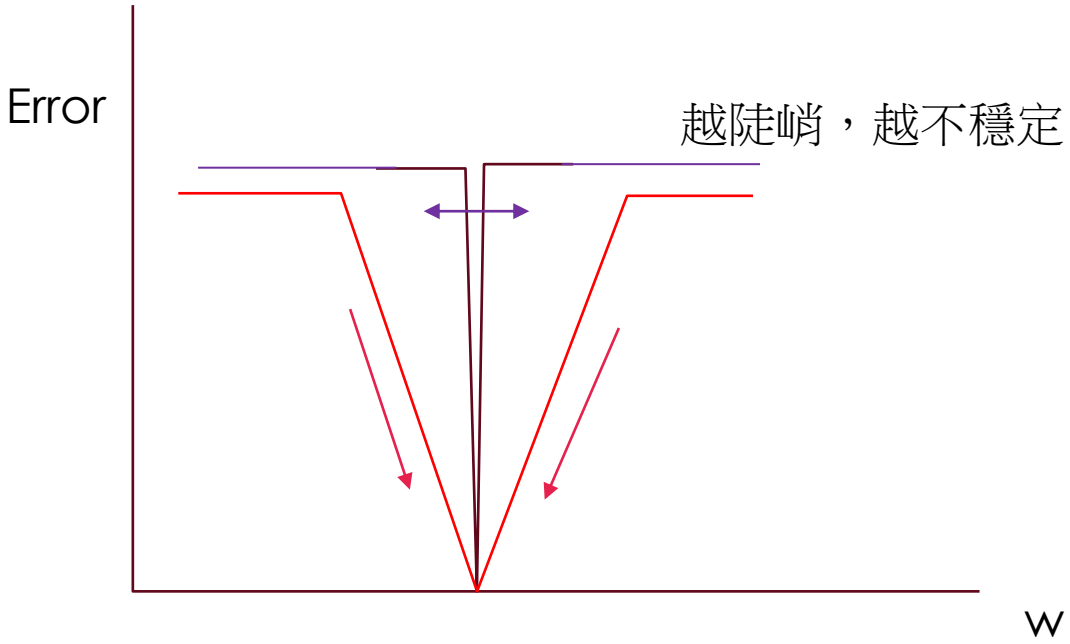
$$y = \begin{cases} 1 & if\ z > 0 \\ 0 & else \end{cases}$$

當x_0=0.5，true vale=1

當x_0=-0.5，true vale=0

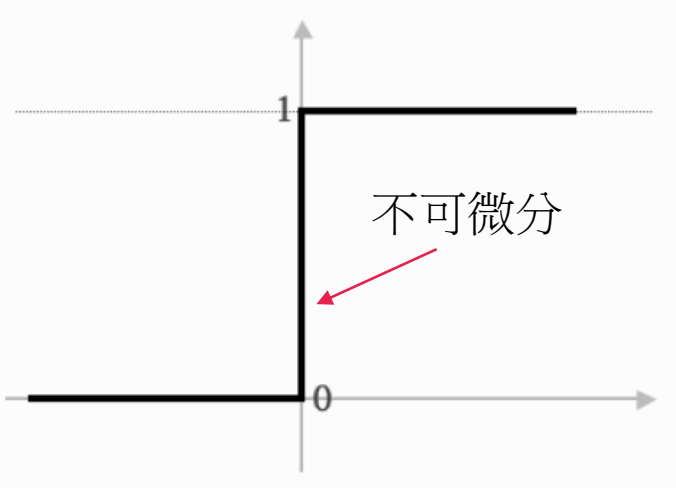


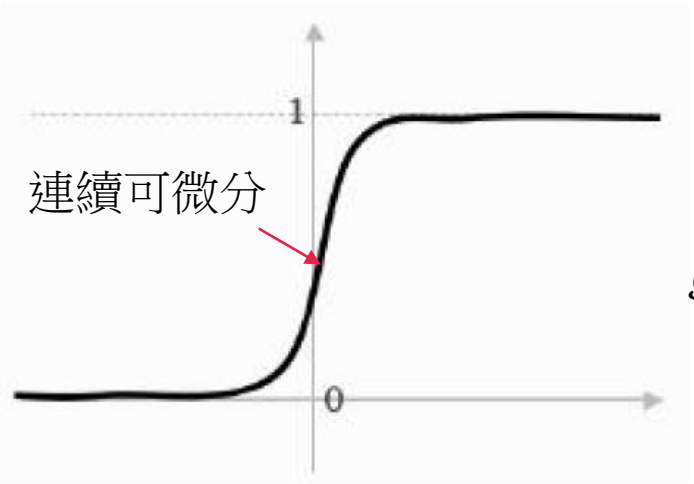
THE PERCEPTRON (複習)



我們應該選擇另一種激活函數！

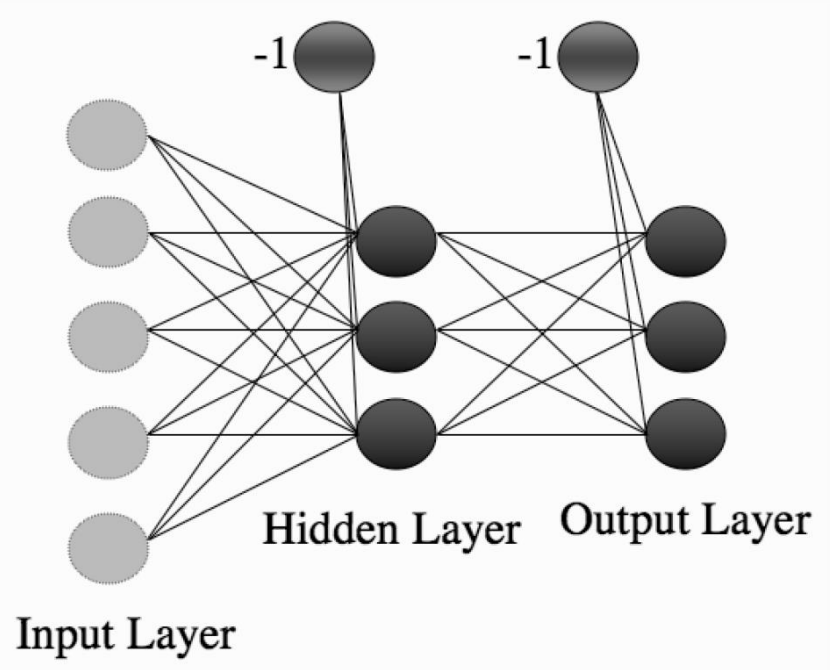
激活函數的選擇

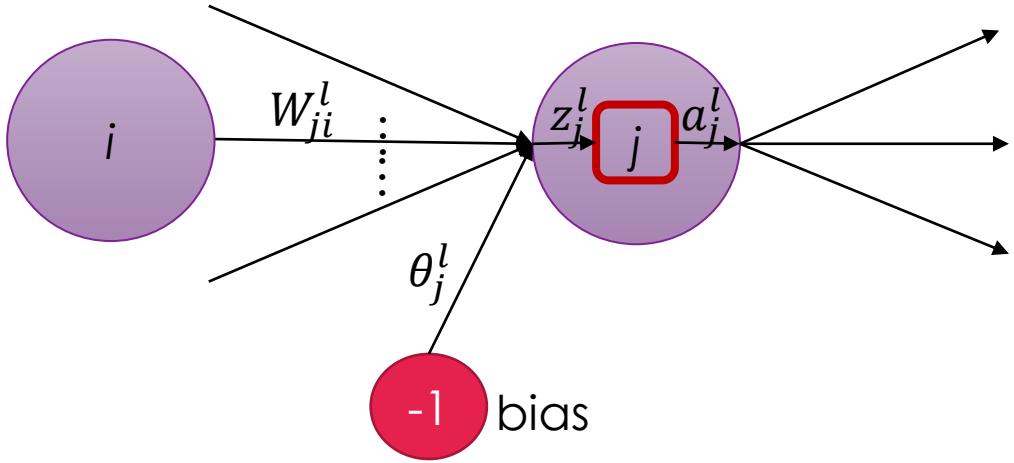




$$g(h) = \frac{1}{1 + e^{-\beta h}}$$

多層感知神經網路的架構



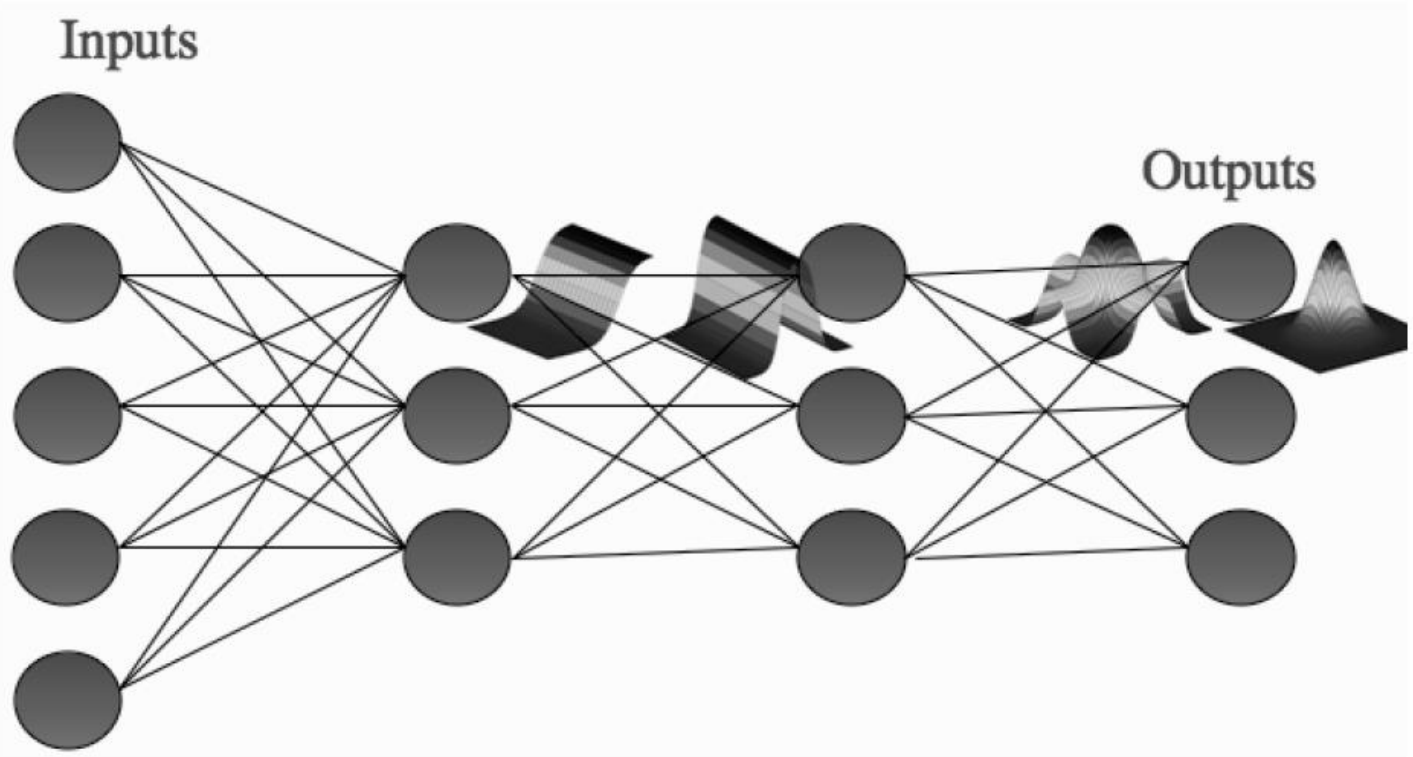


中間隱藏層的神經元激活函數採用**Relu**函數，
輸出端的神經元的激活函數則是問題而定!

神經元的淨輸入z_j^l 變成:

$$z_j^l = \sum_i W_{ji}^l \times x_i - \theta_j^l$$

資料轉換的示例圖



網路的前向運算程序

具有一層隱藏層的神經網路，假設輸入層有有 3 個節點，輸入數據 X 中有 3 筆數據，其標籤為 y ，隱藏層有 2 個節點，隱藏層權重矩陣為 W1 ，線性組合 Z 等於 X • W1，經過激活函數 Relu 的值令為 L = Relu (Z)， 輸出層有 1 個節點，權重矩陣為 W2，線性輸出 O 等於 L W2

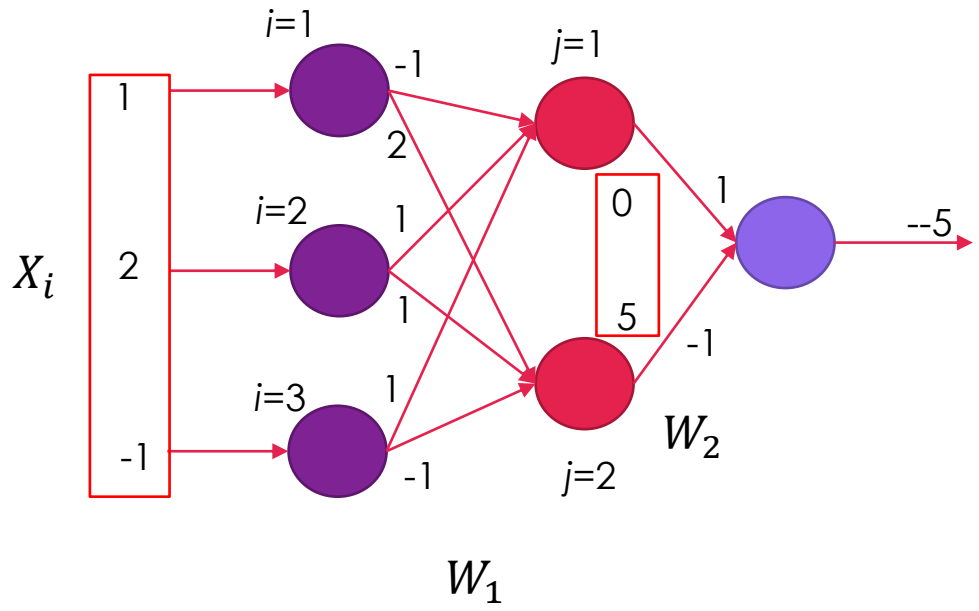
$$X^T = \begin{bmatrix} 1 & 2 & -1 \\ 2 & -3 & 2 \\ -1 & -1 & 3 \end{bmatrix}, y = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$$

$$W1 = \begin{bmatrix} -1 & 2 \\ 1 & 1 \\ 1 & -1 \end{bmatrix}, W2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

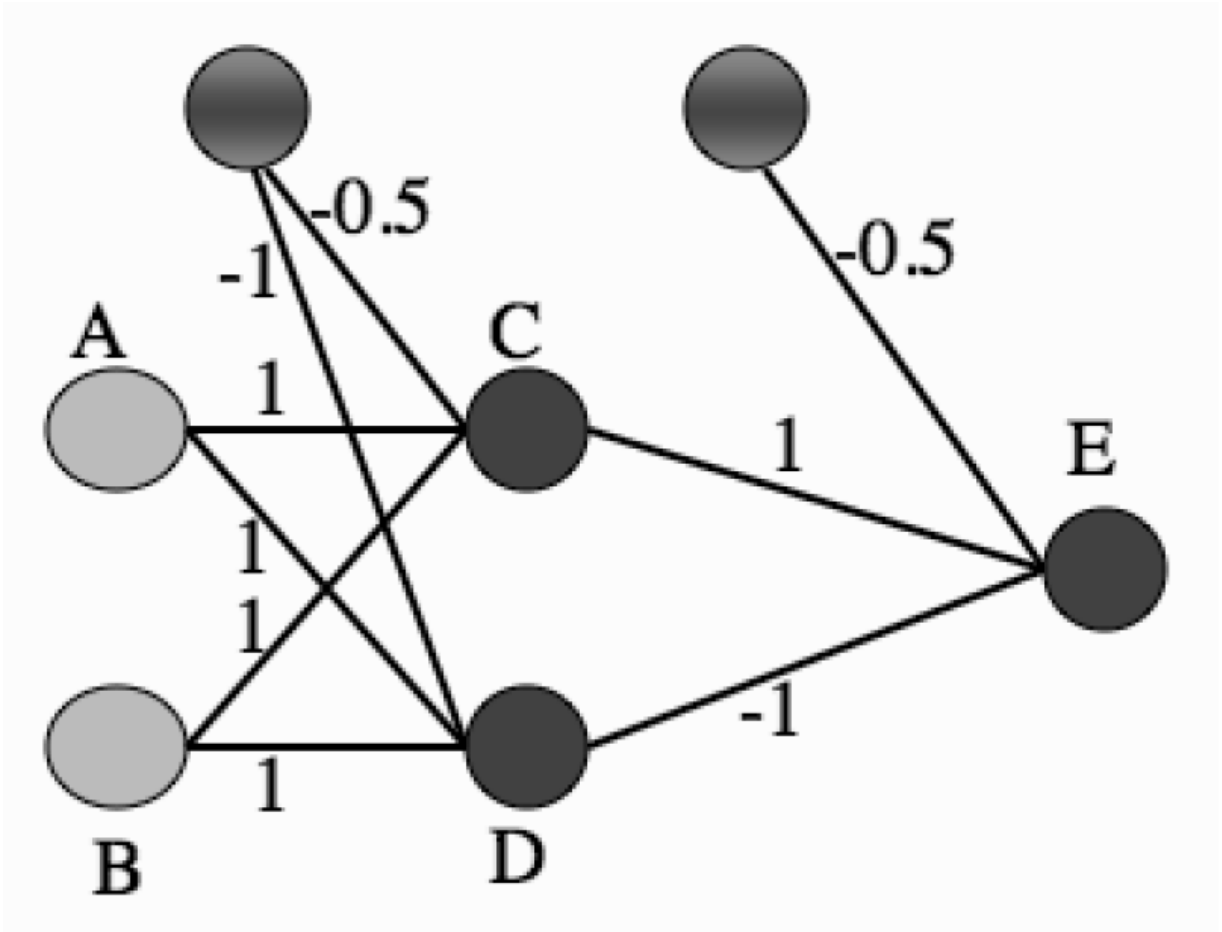
$$Z = X^T \cdot W_1 = \begin{bmatrix} 0 & 5 \\ -3 & -1 \\ 3 & -6 \end{bmatrix}$$

$$L = Relu(Z) = \begin{bmatrix} 0 & 5 \\ 0 & 0 \\ 3 & 0 \end{bmatrix}$$

$$O = L \cdot W_2 = \begin{bmatrix} -5 \\ 0 \\ 3 \end{bmatrix}$$



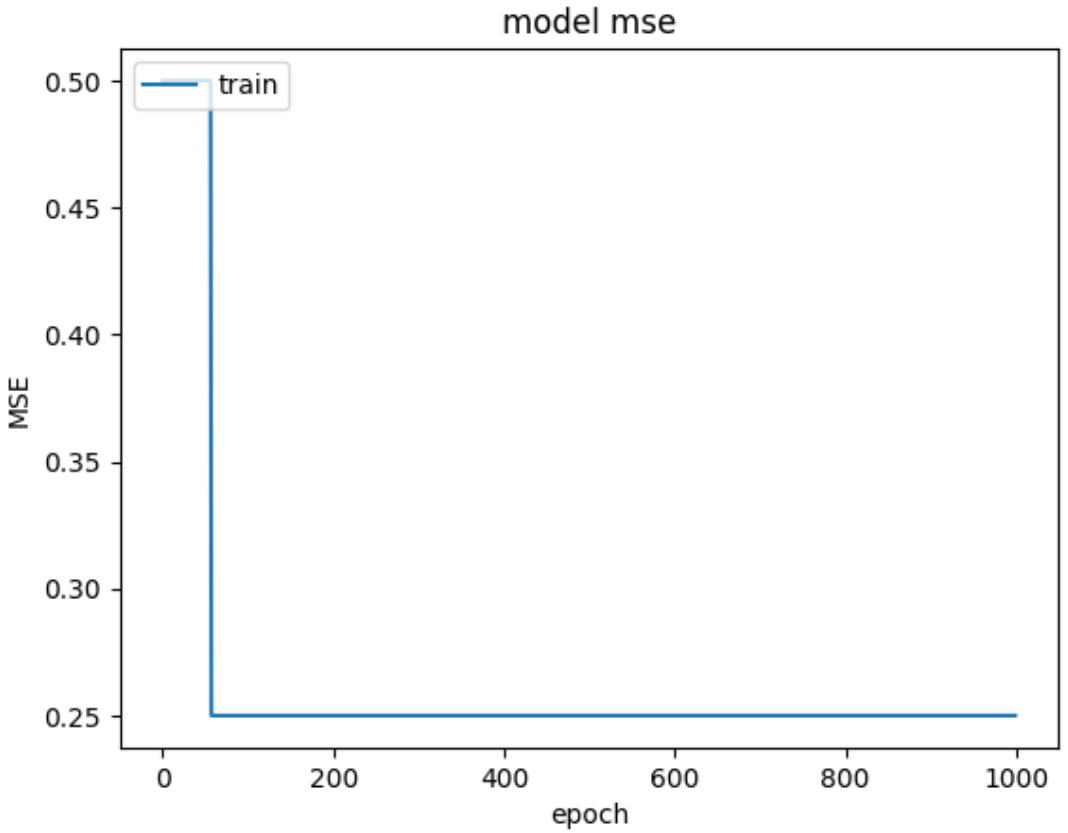
範例：解 XOR問題的MLP類神經網路



A	B	E
0	0	0
0	1	1
1	0	1
1	1	0

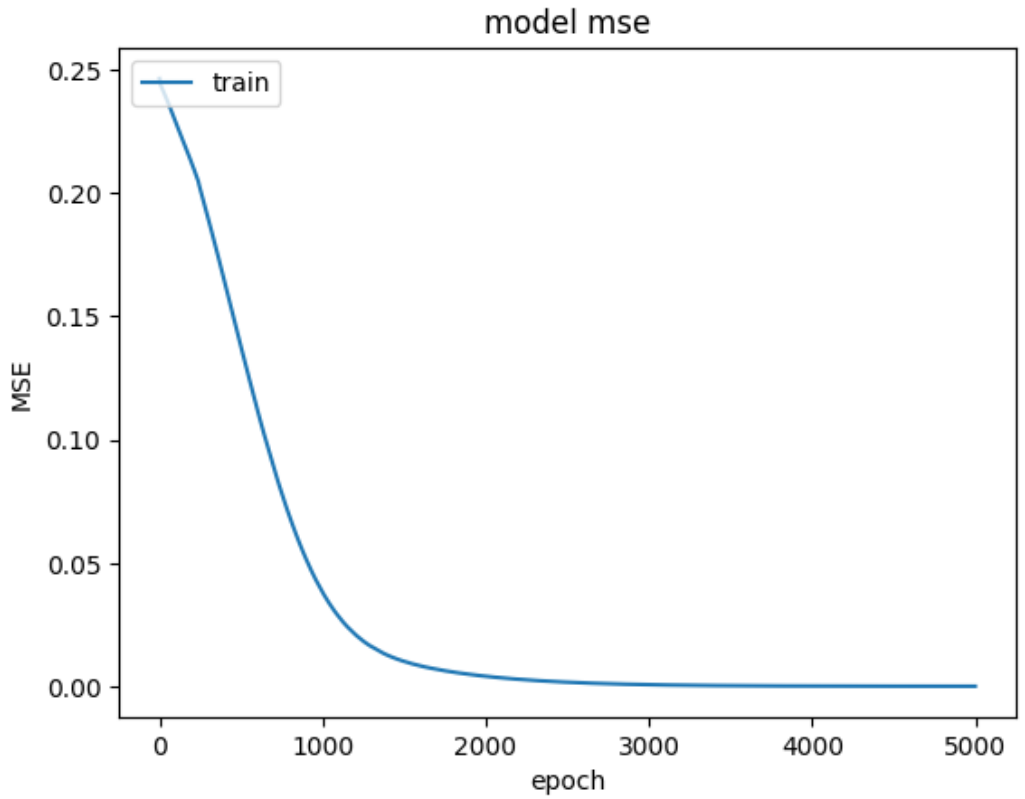
範例：解 XOR問題的PERCEPTRON類神經網路

```
# Create a Sequential model
# Construct an instance of CustomModel
inputs = Input(shape=(2,))
outputs = Dense(1, activation =
custom_activation )(inputs)
model = CustomModel(inputs, outputs)

model.compile( loss="mse", metrics=["mse"] )
#XOR Problem
trainin=np.array([[0.0, 0.0], [0.0, 1.0],[1.0,
0.0],[1.0, 1.0]])
traintgt=np.array([[0.0], [1.0],[1.0],[0.0]])

history= model.fit(x=trainin, y=traintgt,
epochs=1000, verbose=2)
```

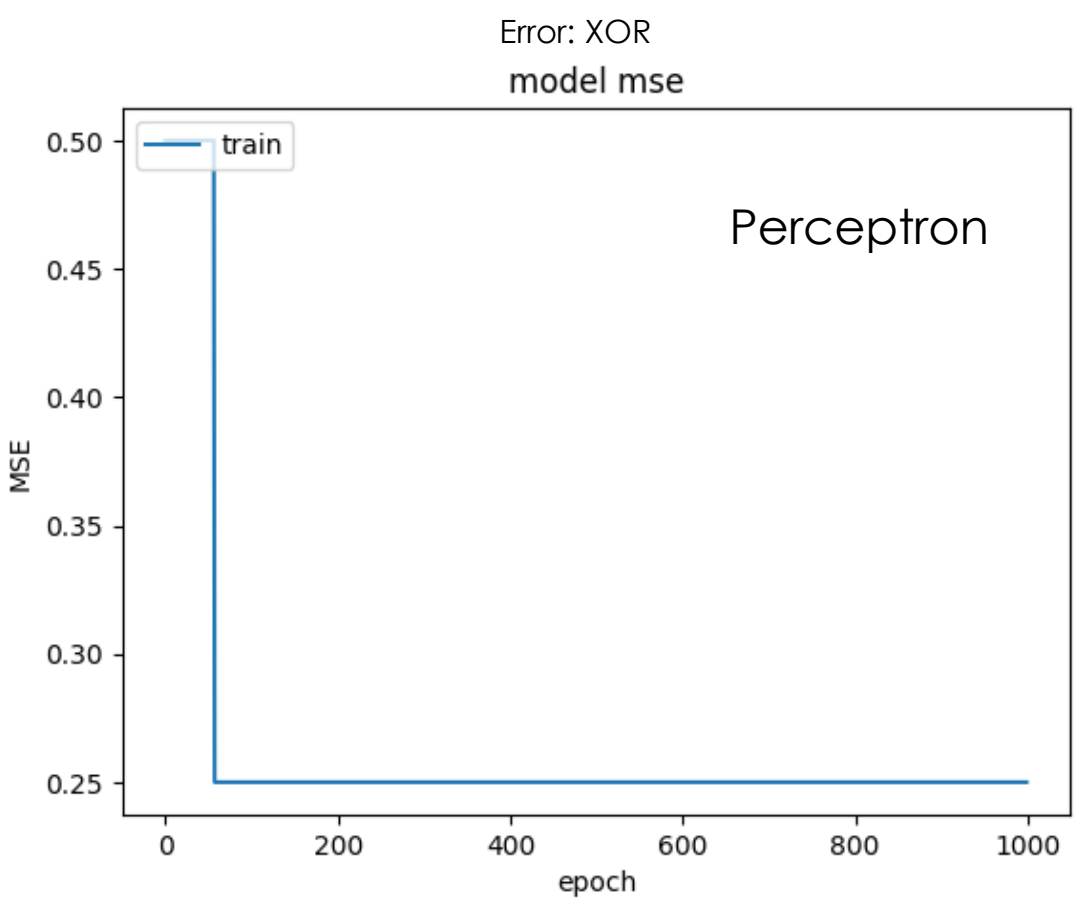


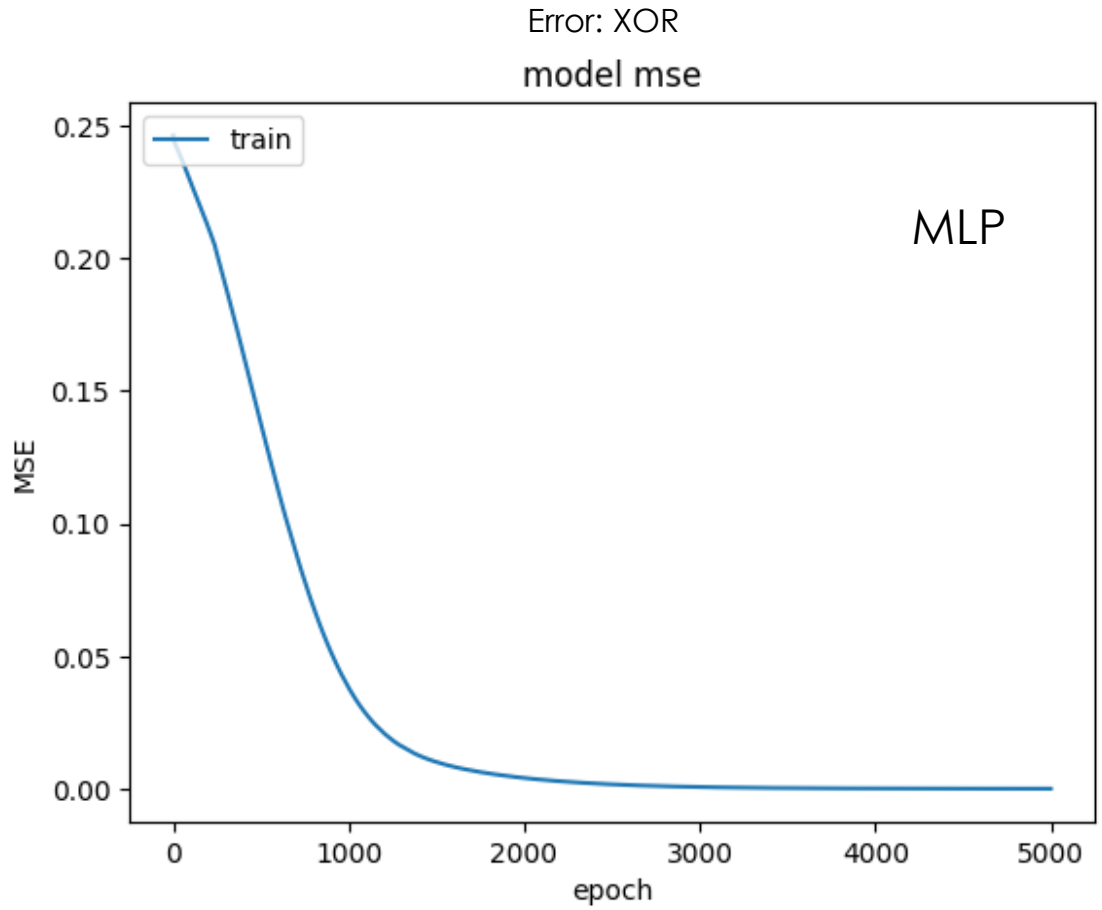
範例：解 XOR問題的MLP類神經網路

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Input
from tensorflow.keras import Model
import numpy as np
# Create a Sequential model
# Construct an instance of CustomModel
inputs = Input(shape=(2,))
hidden=Dense(10, activation = 'relu')(inputs)
outputs = Dense(1, activation = 'sigmoid' )(hidden)
model = Model(inputs, outputs)

model.compile( loss="mse", metrics=["mse"] )
#XOR Problem
trainin=np.array([[0.0, 0.0], [0.0, 1.0],[1.0,
0.0],[1.0, 1.0]])
traintgt=np.array([[0.0], [1.0],[1.0],[0.0]])
history= model.fit(x=trainin, y=traintgt,
epochs=1000, verbose=2)
```



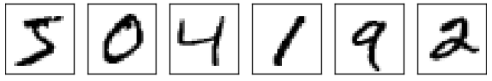
PERCEPTRON 與MLP 訓練結果

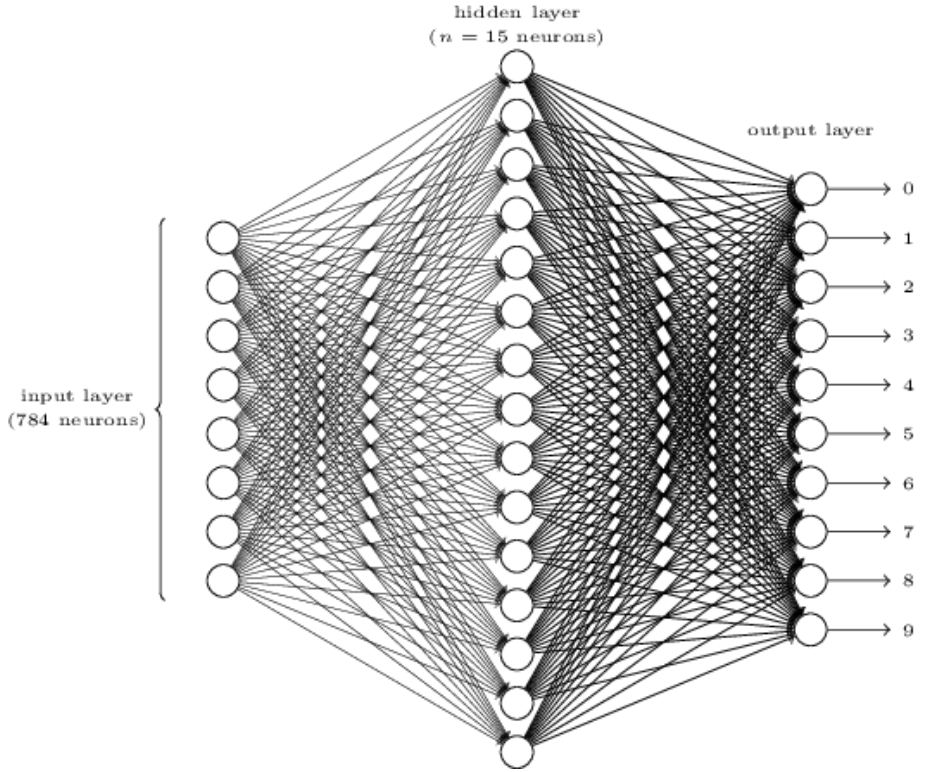




範例：辨識阿拉伯數字的MLP網路

- 辨識手寫阿拉伯數字的神經網路 (MNIST database)
  - 訓練資料包含 28×28 (=**784**) 像素的手寫阿拉伯數字灰階影像
  - 輸出層包含**10**個神經元，分別代表 0~9
  - **均方差**損失函數 $C_{w,b}\ (x) = \frac{1}{2} \times \|(y(x) - \hat{y})^2\|$
    - **均方差** (**Mean Squared Error**, MSE)
    - $y(x)$:對應 x的真實答案
    - $\hat{y}$: 網路預測的輸出答案
    - w:網路權重參數
    - b:網路的偏值(bias)參數





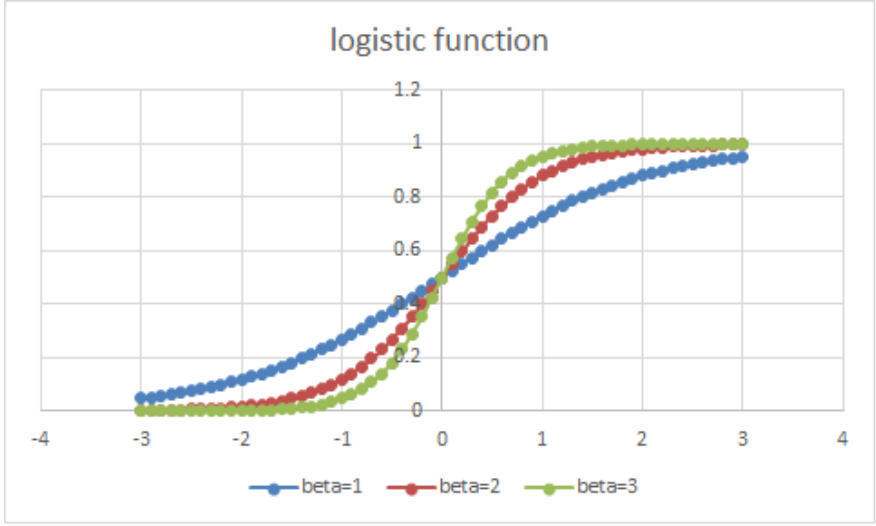
通用逼近定理(UNIVERSAL APPROXIMATION THEOREM)

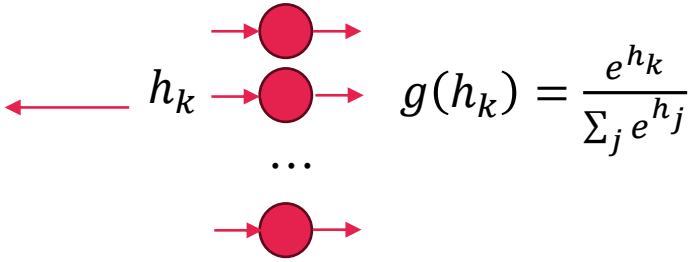
- 包含有限數量神經元的**單一隱藏層**的多層感知器前饋網路可以在激活函數的溫和假設下近似 R^n 的緊湊子集上的連續函數
- 該定理指出當給定適當的參數時，簡單的神經網路可以表示各種有趣的函數（任何平滑映射）
- 然而，它並沒有涉及這些參數的演算法可學習性

Ref: see https://www.youtube.com/watch?v=ljqkc7OLenI

實務建議與議題(1)

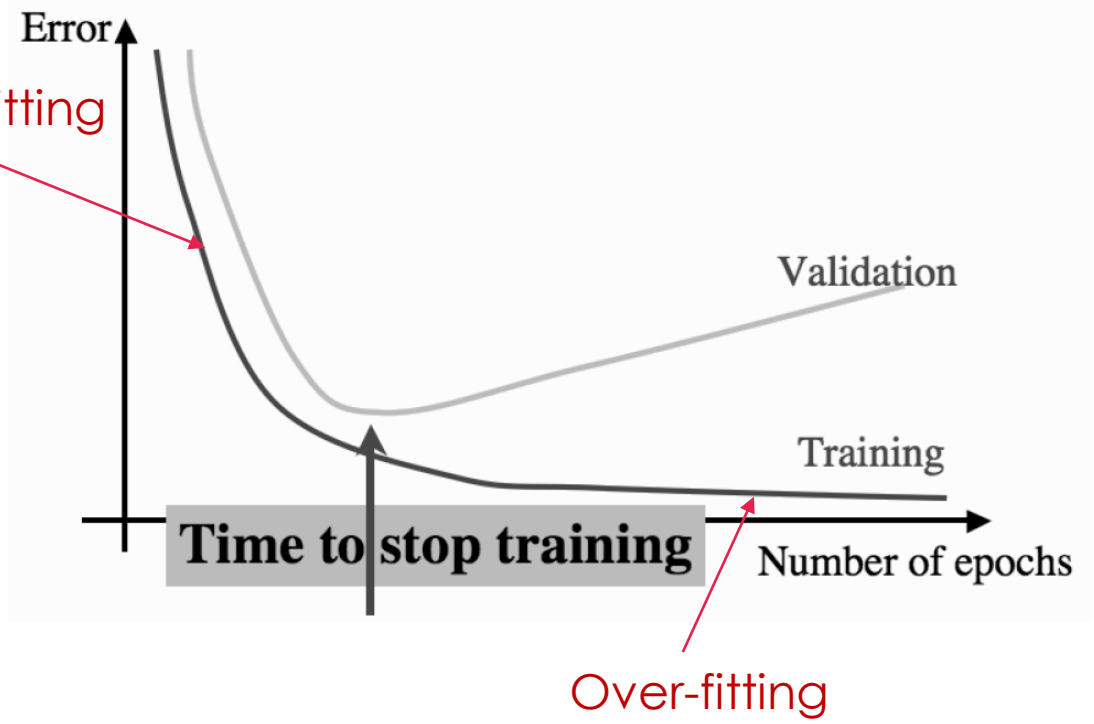
- 權重參數的初始化
  - 假設有 n 連結至一個神經元, 其權重的初值可隨機設為 $\frac{-1}{\sqrt{n}} < w < \frac{1}{\sqrt{n}}$ ，並使其總和為1
- 激活函數
  - 選用 logistic (sigmoid) 函數時, 可使用 $\beta \leq 3$ (不要太陡峭)
  - 對於回歸 (regression) 問題, 輸出層的神經元可選用 linear 函數: $g(h) = h$
  - 對於分類 (classification)問題
    - 二元分類，輸出層的神經元們可選用 sigmoid 函數
    - 多元分類，輸出層的神經元們可選用 soft-max 函數: $g(h_k) = \frac{e^{h_k}}{\sum_j e^{h_j}}$





實務建議與議題(2)

- 何時停止學習?
  - 欠擬合(Under-fitting)
    - 網路沒有學習到訓練資料的正確分佈
  - 過擬合 (Over-fitting)
    - 網路開始學習訓練資料中的雜訊
  - 使用驗證資料集 (validation data set) 檢查停止點

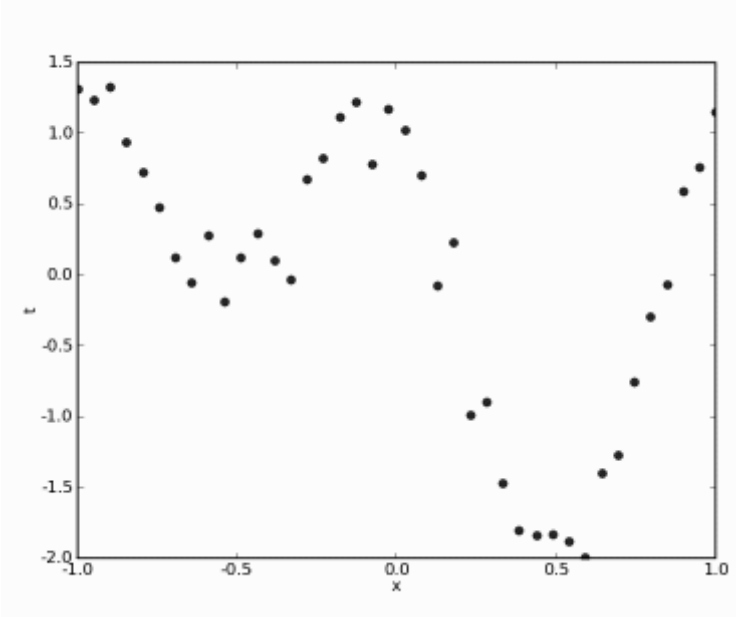


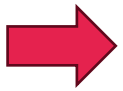
MLP的應用

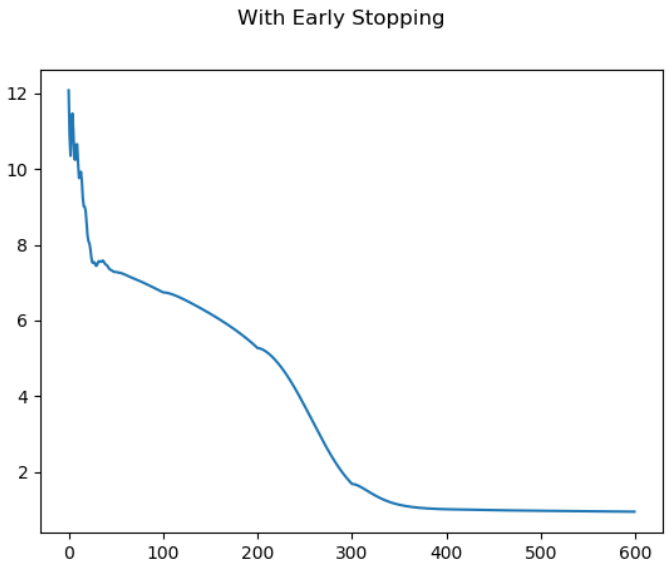
- 回歸 (Regression) 問題
- 分類 (Classification)問題
- 時間序列預測 (Time-series Prediction)
- 資料壓縮與編碼 (Data compression/encoding)

回歸 (REGRESSION) 問題

- Sinewave 回歸案例
  - 輸出層神經元激活函數: linear
  - 包含 訓練, 驗證, 與測試資料集





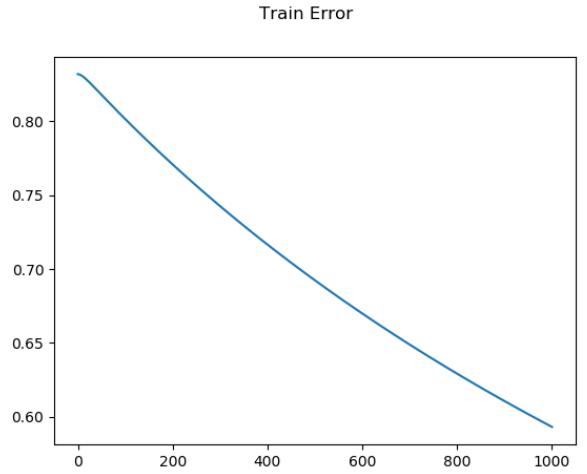


分類 (CLASSIFICATION)問題

- Iris 資料集
  - 每個類別先做 **one-hot-encoding**
    - 4個實數形輸入值
    - 3種鳥類
  - 輸出端神經元採用 soft-max 激活函數
    - 所有輸出值總和為1
    - 挑選輸出值最大者作為預測類別

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
-- Iris Setosa
-- Iris Versicolour
-- Iris Virginica

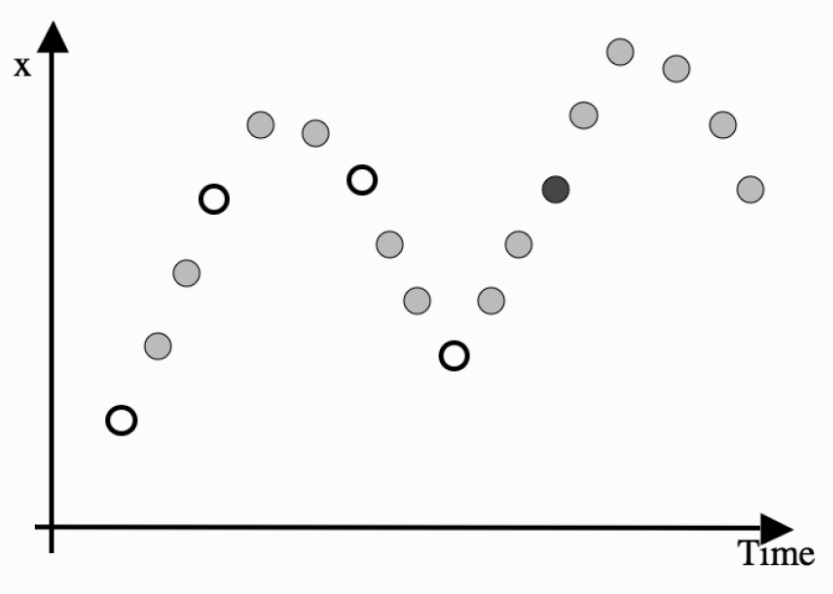
5.1,3.5,1.4,0.2,0
4.9,3.0,1.4,0.2,0
4.7,3.2,1.3,0.2,0
6.7,3.1,4.4,1.4,1
5.6,3.0,4.5,1.5,1
5.8,2.7,4.1,1.0,1
6.2,2.2,4.5,1.5,2
5.6,2.5,3.9,1.1,2
5.9,3.2,4.8,1.8,2



時間序列預測

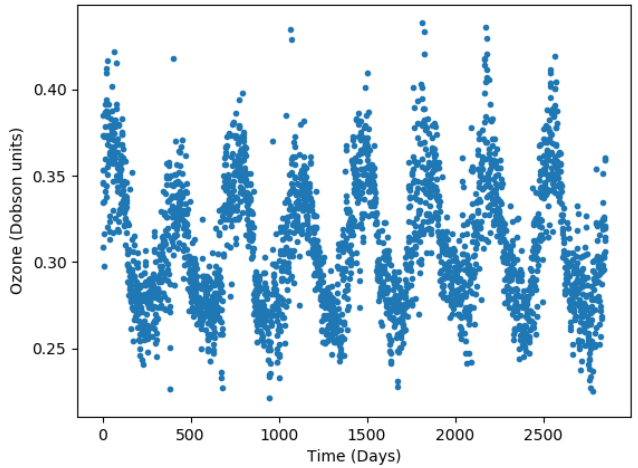
- 給定資料: $x(1), x(2), x(3), \ldots\ldots$, 學習下列函數：
  - $y = x(t+\tau) = f\left(x(t), x(t-\tau), \ldots, x(t-k\tau)\right)$
  - τ: 時間間隔
  - k : 參考資料個數

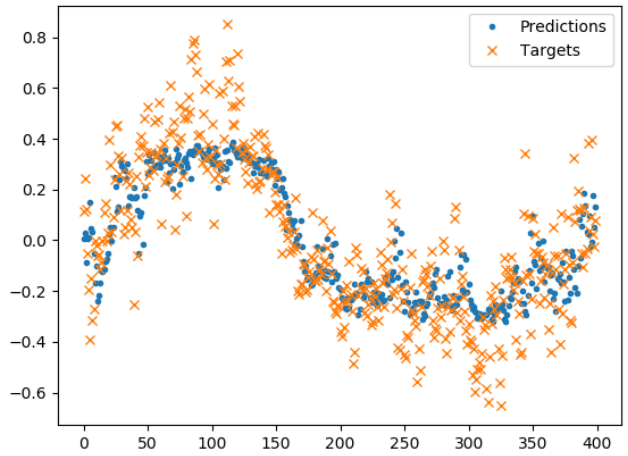
例如, 給定 τ=2, k=3
則 :
x(1), x(3), x(5) => x(7) = ?
x(2), x(4), x(6) => x(8) = ?
….



時間序列預測

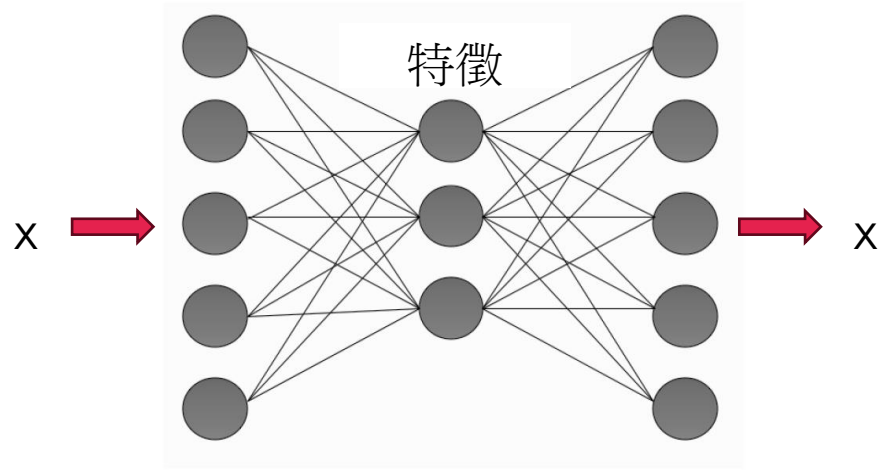
- 資料集: PNOZ.dat
  - 2855 筆資料
  - 4個變數, 但本例子只使用第3個:臭氧層的厚度
  - 輸出端神經元使用 linear 激活函數
  - 使用最後400筆資料做測試t
  - 給定 τ=2, k=3,
    - 有3個輸入，1個輸出

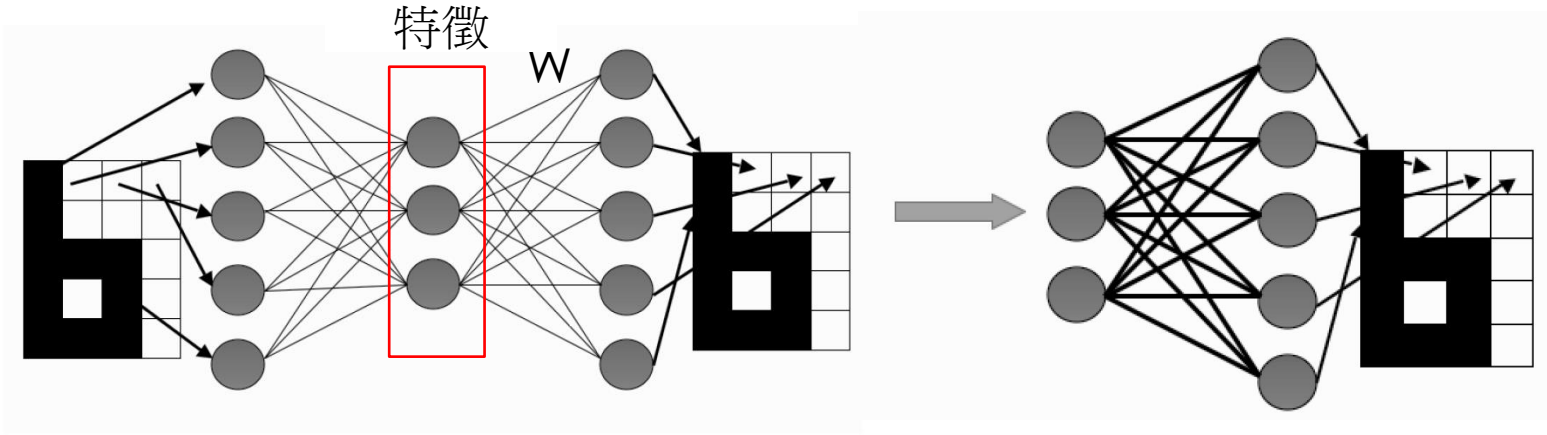




資料壓縮/編碼

- 自我編碼器
  - 訓練網路可以重製輸入資料
- 應用
  - 壓縮
    - 2D 影像 to 1D 向量特徵
  - 解碼
    - 儲存第二層權重W的集合
    - 從特徵回復原始圖片





小結

- MLP類神經模型可以解決非線性可分問題
- 通用逼近定理
- 一些實際問題
  - 權重設定初值
  - 激活函數的選擇
  - 儘早停止學習
- 各種MLP應用練習

REFERENCE

- Online Book
  - http://neuralnetworksanddeeplearning.com/chap2.html

Q&A