

人工智慧實務應用1

2021/8/16

課程內容

Day 1	Day 2	Day 3
人工智慧簡介 機器、深度學習介紹 機器學習實作1 [感知神經網路]	機器學習演算法 機器學習實作2 [wine資料集之機器學習分析] 深度學習演算法	深度學習實作 [kaggle影像辨識]

Day 1 大綱

- 人工智慧
- 機器學習
- 深度學習

人工智慧

自然語言
電腦視覺
數據分析

機器學習

監督式學習

- 分類：決策樹、邏輯迴歸、SVM
- 迴歸：線性、非線性迴歸

強化式學習

- Q-learning、DQN

非監督式學習

- 分群：K-means
- 降維：PCA

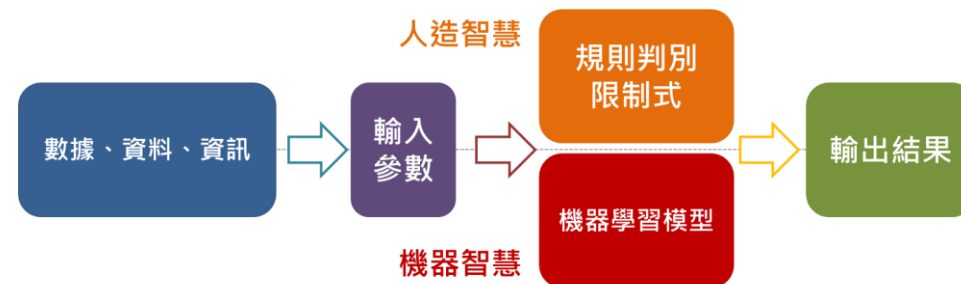
深度學習

- DNN、CNN、RNN

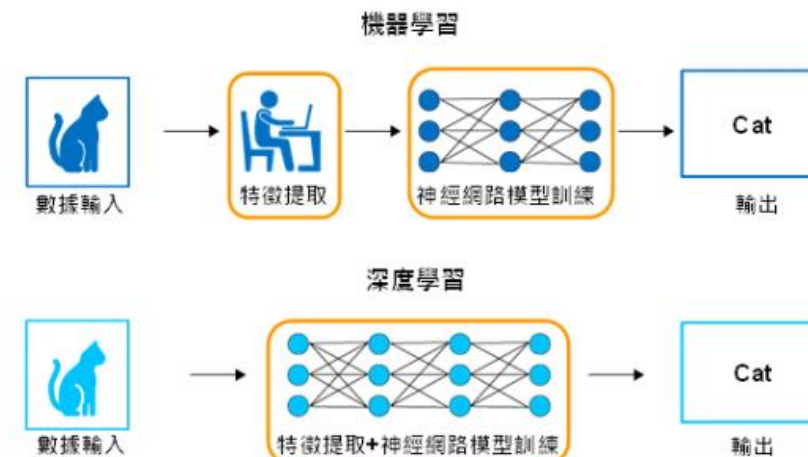
人工智慧簡介

所謂的人工智慧(Artificial Intelligence) 是人類建立於機器上的類似大腦智慧的一種判斷機制。其目的以編寫程式的方式，模擬出人類大腦中的決策，並模仿、理解、學習等等特性，而形成類似人類的「智慧」。

- Rule-based：由專家與工程師依經驗制定規則，設計程式。
- 機器學習：透過人從資料中萃取特徵，將制定規則轉換成ML模型，用學好的模型去判斷。
- 深度學習：與機器學習差別在於，特徵提取改由機器自行學習。

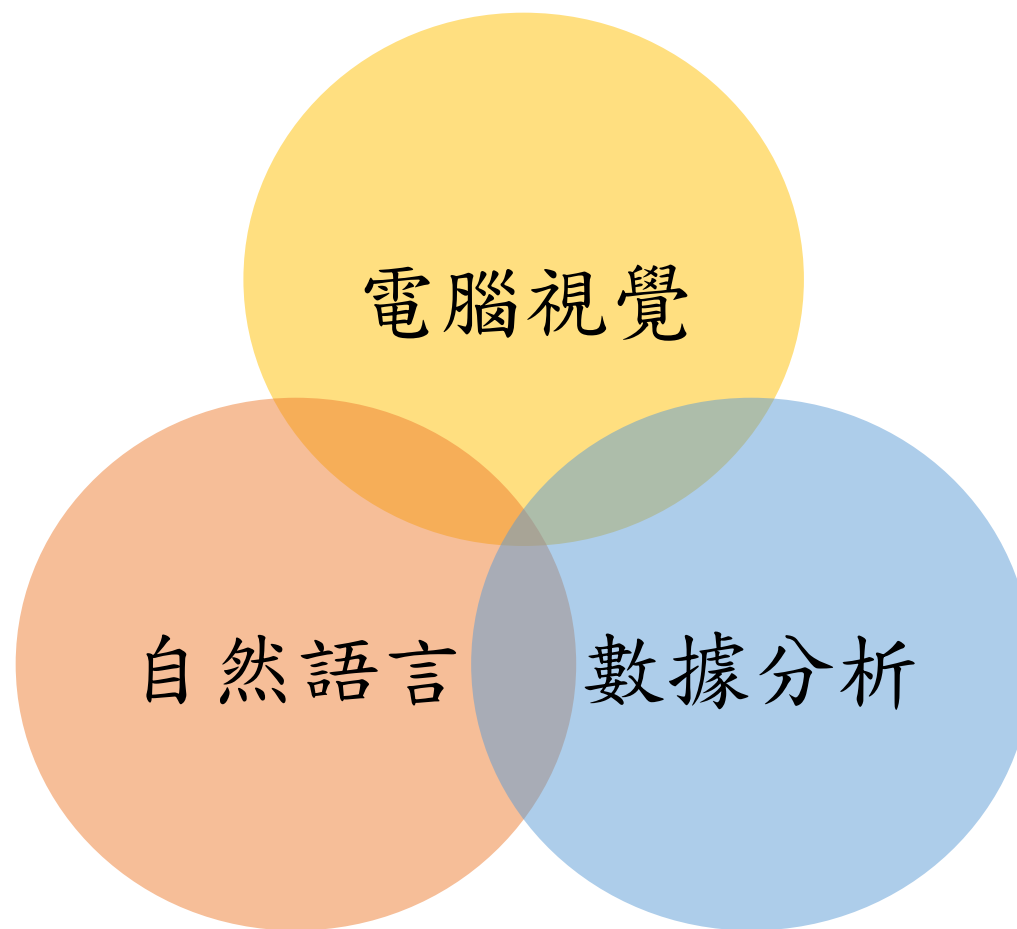


https://aifreeblog.herokuapp.com/posts/59/Machine_Learning_Intro/



https://www.digitimes.com.tw/tech/showimg.asp?sourcetype=2&filename=0000020189922_0_U74HLH0ZVE.png&source=20180930-204

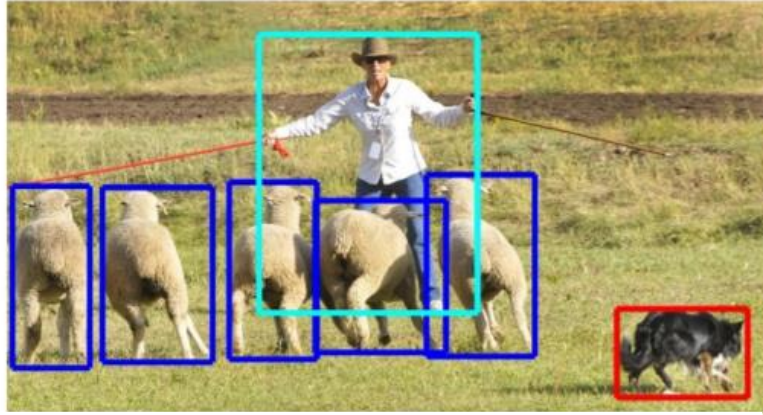
應用情境



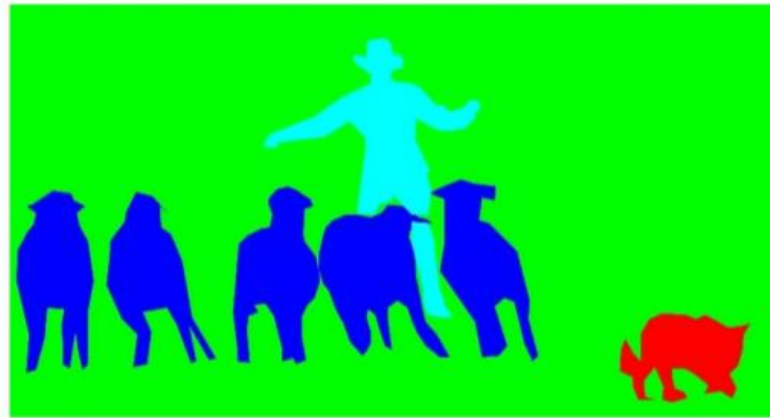
電腦視覺



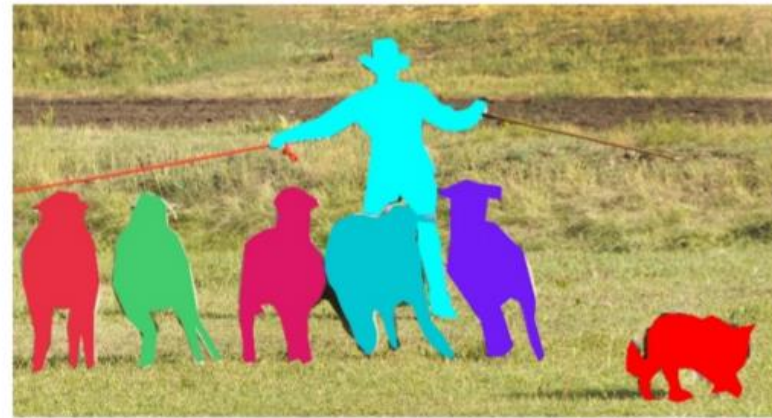
(a) Image Classification



(b) Object Localization



(c) Semantic Segmentation

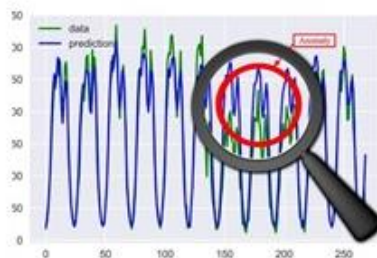


(d) Instance Segmentation

自然語言



數據分析



異常偵測
(Anomaly Detection)



推薦系統
(Recommended System)



預測維修系統
(Predictive Maintenance System)



消費行為
(Consumer Behavior)



資訊安全
(Information Security)



機器學習

- 從資料中找出一個函數

function(輸入資料)=輸出

溫度預測

$f(10)=10$

$f(\text{昨天溫度})=25\text{度}$

$f(20)=20$

$f(\text{今天溫度})=27\text{度}$

$f(33)=33$

$f(\text{預測明天})=28\text{度}$

語音辨識

$f(\text{audio waveform})='Joy'$

$f(\text{audio waveform})='Peter'$

$f(\text{audio waveform})='Toms'$

影像辨識

$f(\text{cat image})='cat'$

$f(\text{dog image})='dog'$

$f(\text{duck image})='duck'$

分類問題中了解機器學習過程

- 設定：監督式學習、分類任務
- 問題：預測照片為貓或狗
- 根據已知的貓跟狗的照片給機器做訓練，期望機器可以預測未知的照片是貓還是狗
- 機器學習的三步驟：
 - Step 1：定義函數
 - Step 2：定義找函數的機制
 - Step 3：找出最好的函數

Step1: 定義函數

f_1, f_2, f_3, \dots

- function set (函數的集合)
- 通常稱為 *model*
- 理論上有無限多個 *function*

$f_1(\text{cat}) = \text{'dog'}$

$f_2(\text{cat}) = \text{'cat'}$

$f_3(\text{cat}) = \text{'cat'}$

$f_1(\text{dog}) = \text{'cat'}$

$f_2(\text{dog}) = \text{'cat'}$

$f_3(\text{dog}) = \text{'dog'}$

Step2: 定義找函數的機制

loss function

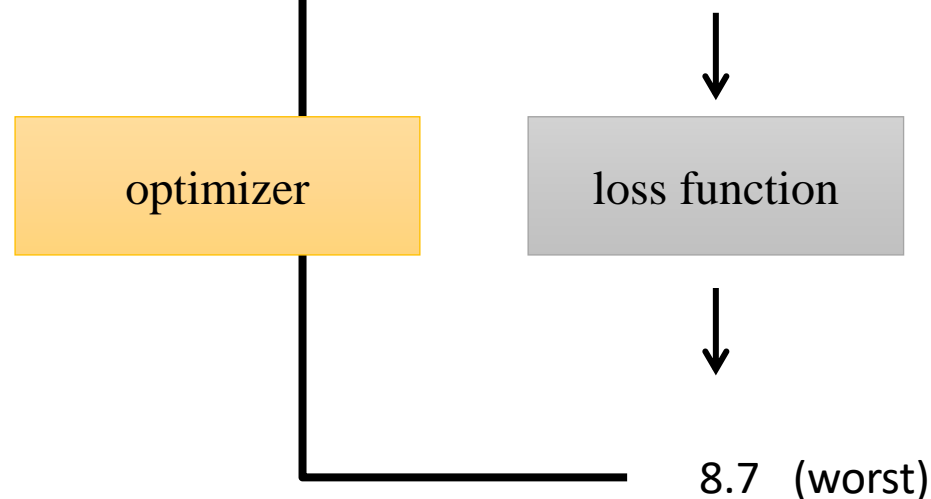
loss function (損失函數)
評鑑模型學的好壞

optimizer

optimizer (優化器)
依據loss function的結果，
修正模型

$f_1(\text{cat image}) = \text{'dog'}$
答案: cat

$f_1(\text{dog image}) = \text{'cat'}$
答案: dog



loss function

loss function (損失函數)
評鑑模型學的好壞

optimizer

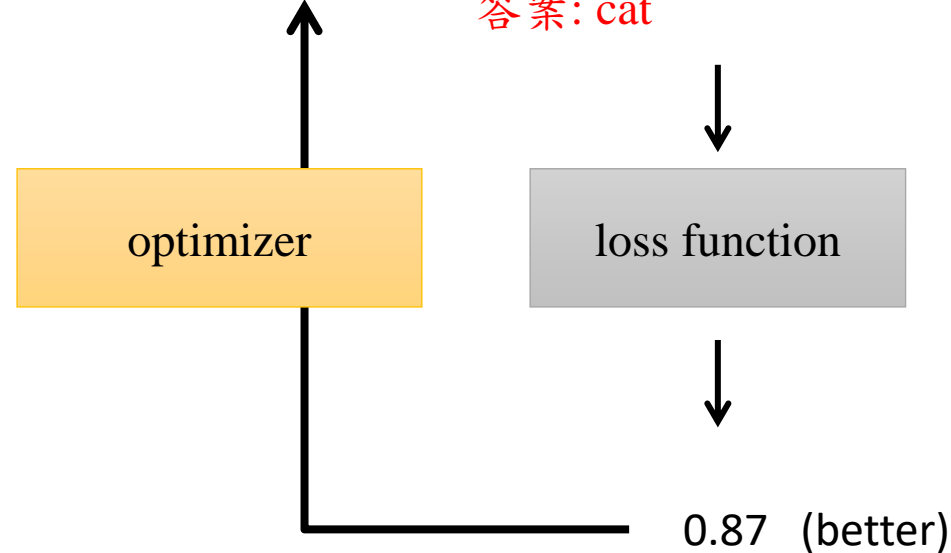
optimizer (優化器)
依據loss function的結果，
修正模型

$f_2(\text{cat image}) = \text{'cat'}$

答案: cat

$f_2(\text{dog image}) = \text{'cat'}$

答案: dog



Step3: 找出最好的函數

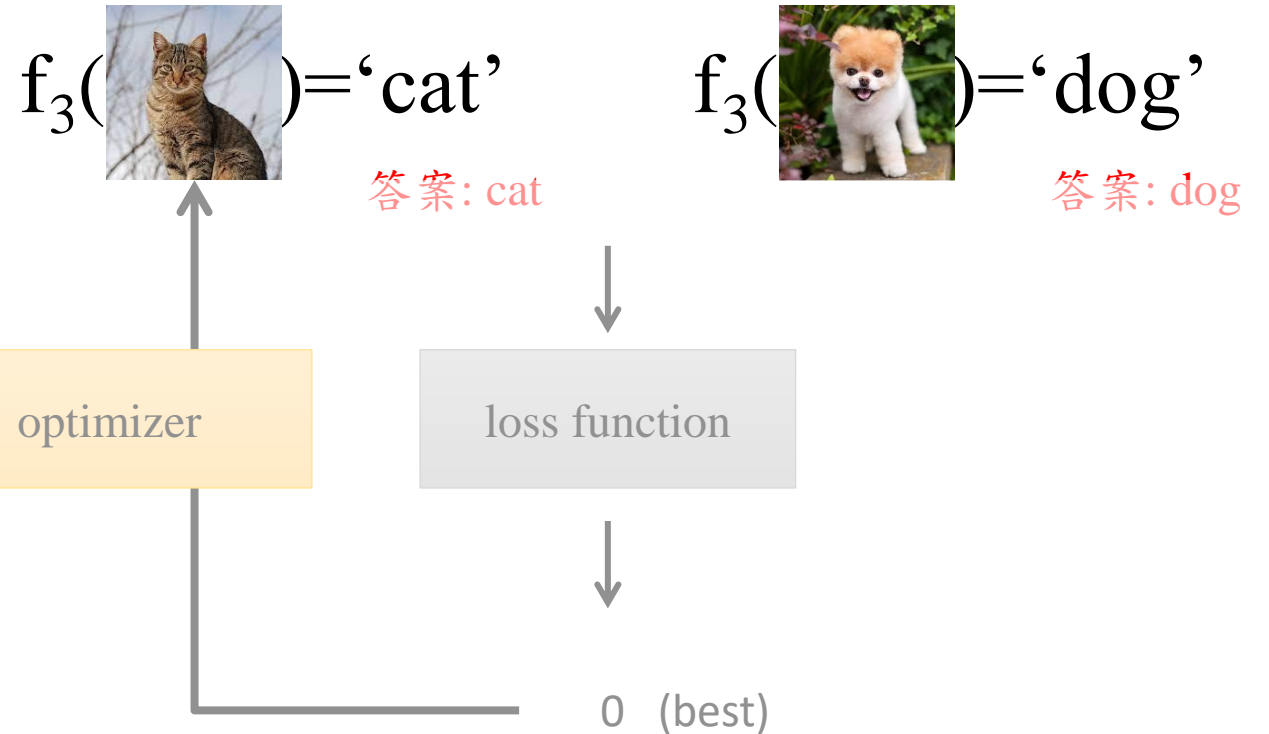
Best model
 $f_{\text{best}} = f_3$

loss function

loss function (損失函數)
評鑑模型學的好壞

optimizer

optimizer (優化器)
依據loss function的結果，
修正模型

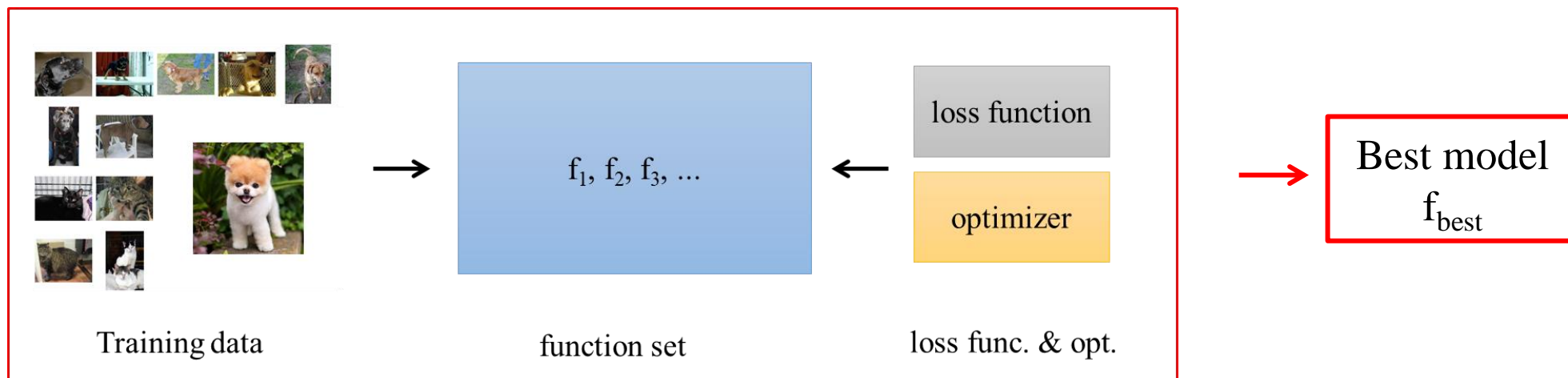


Training 訓練

Step1: 定義一組 function set

Step2: 定義找函數的機制(loss function/optimizer)

Step3: 找出最好的函數

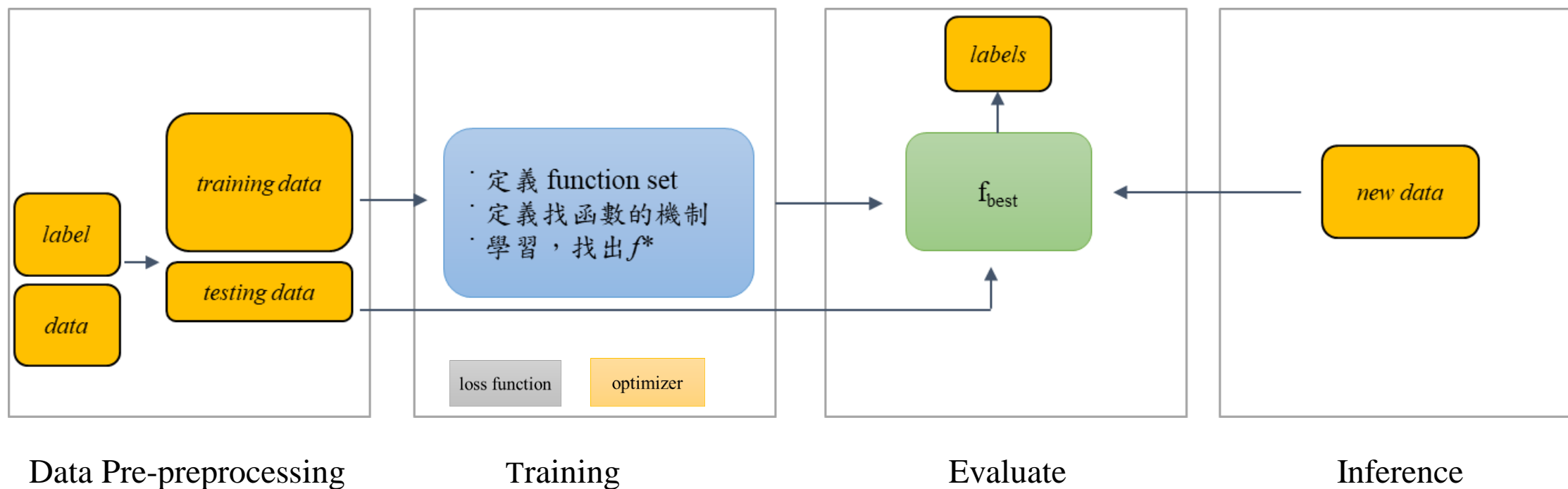


Inference 推論

使用新的資料餵入 f_{best} 得到推論結果

$$f_{\text{best}}(\text{cat}) = \text{'cat'}$$

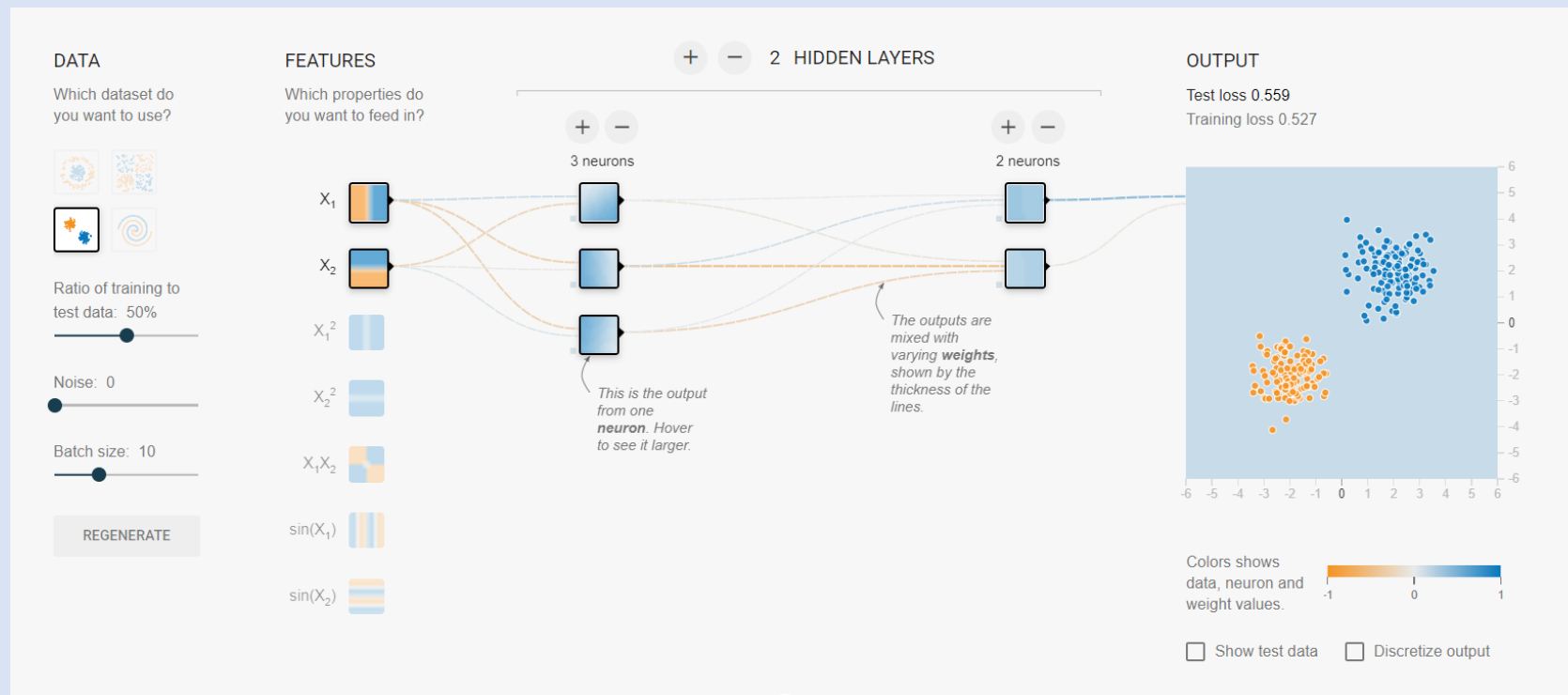
基本開發流程



AI Neural Network Playground

網址：

<https://playground.tensorflow.org/#activation=sigmoid&batchSize=10&dataset=gauss®Dataset=reg-plane&learningRate=0.03®ularizationRate=0&noise=0&networkShape=3,2&seed=0.67748&showTestData=false&discretize=false&percTrainData=50&x=true&y=true&xTimesY=false&xSquared=false&ySquared=false&cosX=false&sinX=false&cosY=false&sinY=false&collectStats=false&problem=classification&initZero=false&hideText=false>



機器學習-學習方式

① 監督式學習：

- 機器透過有標籤(正確答案)的資料學習



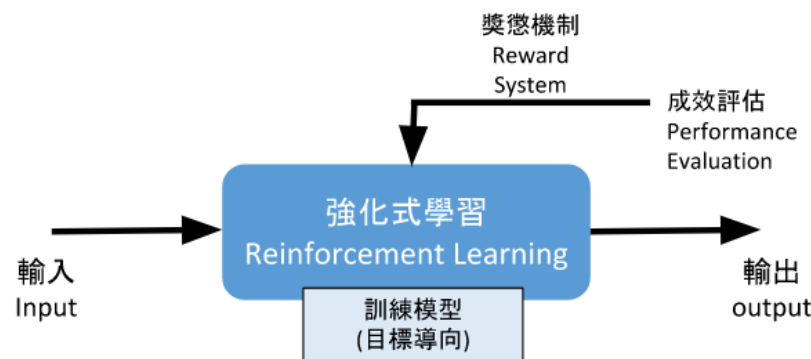
② 非監督式學習：

- 機器透過沒標籤的資料(資料特徵分群)



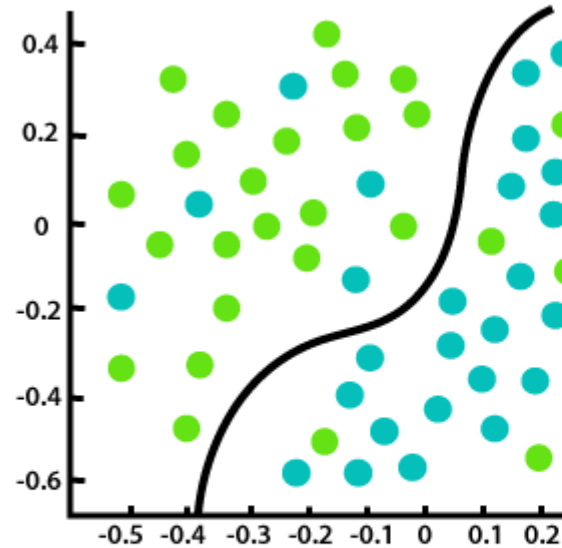
③ 強化式學習：

- 在環境給予的獎勵或懲罰的刺激下，逐步形成對刺激的預期，產生能獲得最大利益的習慣性行為

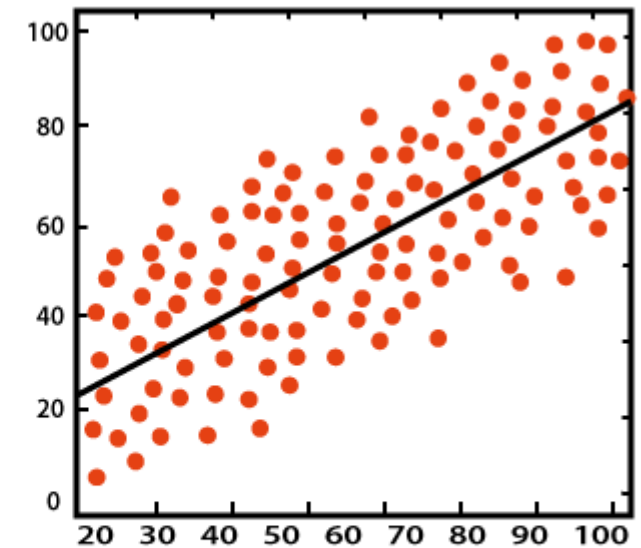


監督式學習

- 迴歸
 - 輸出是純量
 - 算法：線性、非線性迴歸
 - 例如：溫度預測、股票預測
- 分類
 - 輸出是類別
 - 算法：決策樹、SVM、邏輯迴歸
 - 例如：影像分類、語音辨識



Classification

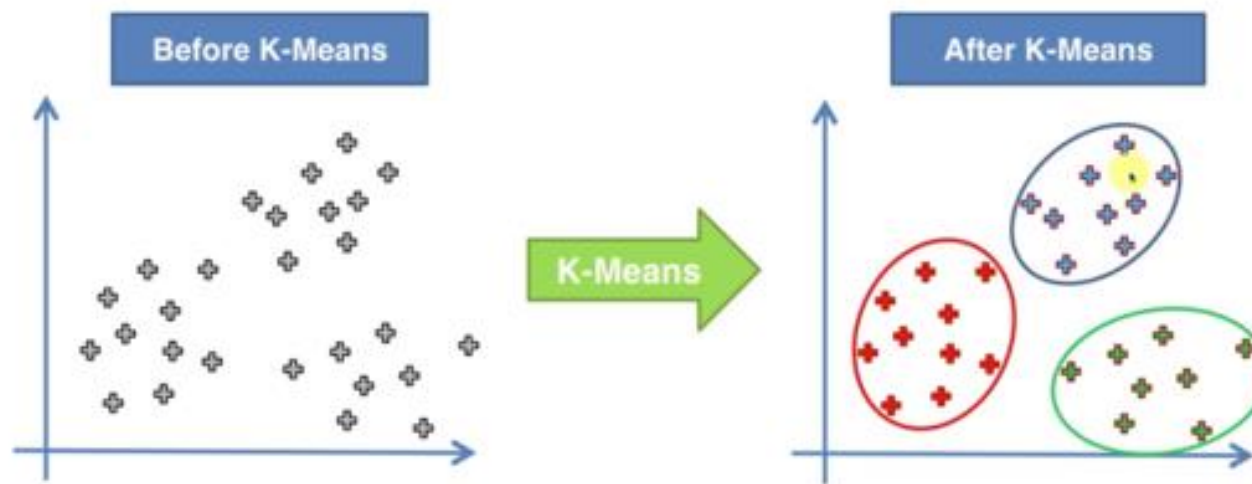


Regression

非監督式學習

- 分群

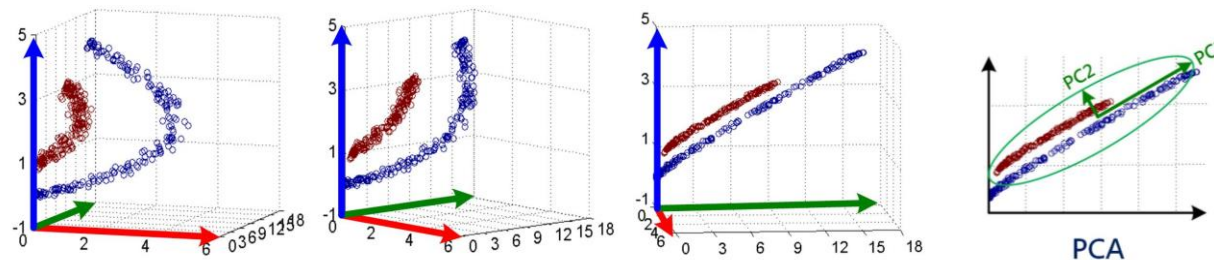
- 找出相似特徵的中心點
- 算法：K-means



<https://www.gushiciku.cn/pl/2Baj/zh-hk>

- 降維

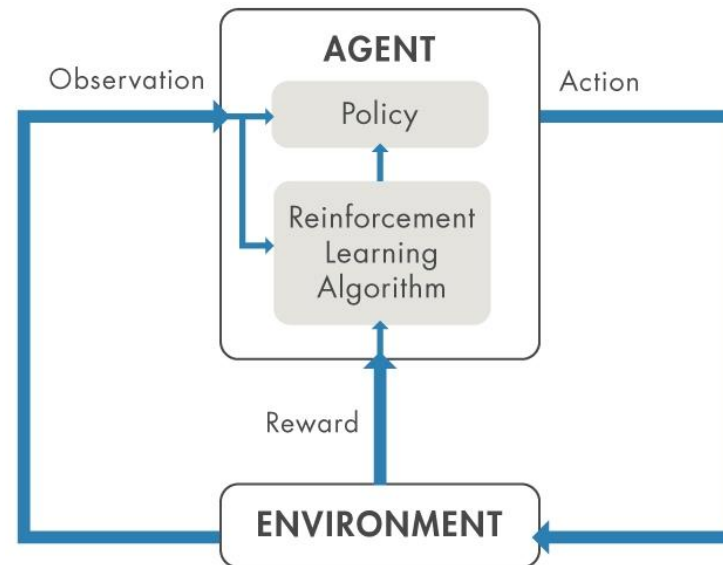
- 降低後續特徵提取演算
- 算法：PCA、LDA



<https://aihub.org.tw/platform/specialist/article/3e34f5fc-ddcb-11ea-a936-0242ac120002?page=>

強化式學習

- 機器(agent)透過環境的正、負向回饋 (positive / negative reward)，從中自我學習，並逐步形成對回饋刺激 (stimulus) 的預期，做出越來越有效率達成目標的行動 (action)，目的就是找到一個最好的Policy(策略)，可以讓reward最多。
- 算法：Q-learning、DQN
- 例如：AlphaGo、遊戲



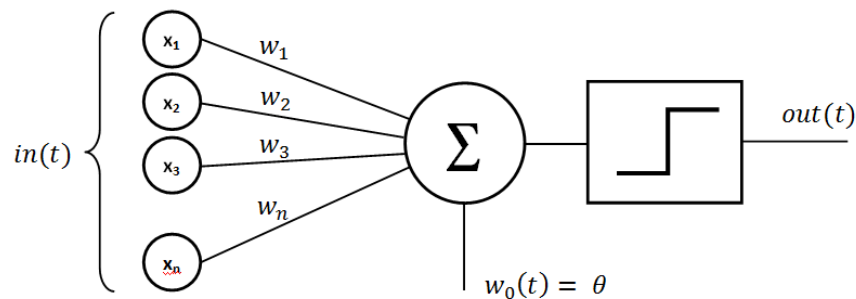
深度學習

- 深度學習又是機器學習的分支，深度學習是人工智慧中，成長最快的領域，深度學習模擬人類神經網絡的運作方式。
- 常見的神經網路類型：
 - 深度神經網路DNN(Deep Neural Network)
 - 卷積神經網路CNN(Convolutional Neural Network)
 - 遞迴神經網路RNN(Recurrent Neural Network)

演進

- Perceptron
- Multilayer Perceptron(MLP)
- Deep Neural Network(DNN)

$$y = \sigma \left(\sum wx \right)$$



w: weight & bias

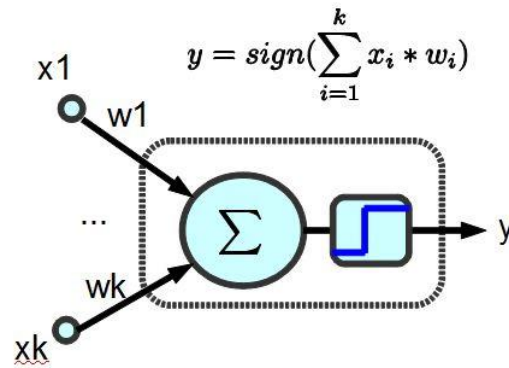
x: input

y: output

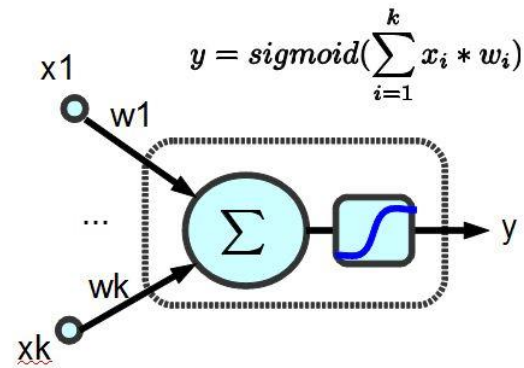
σ : activation function , 這裡的activation func.是sign函數

演進

- Perceptron
- Multilayer Perceptron(MLP)
- Deep Neural Network(DNN)



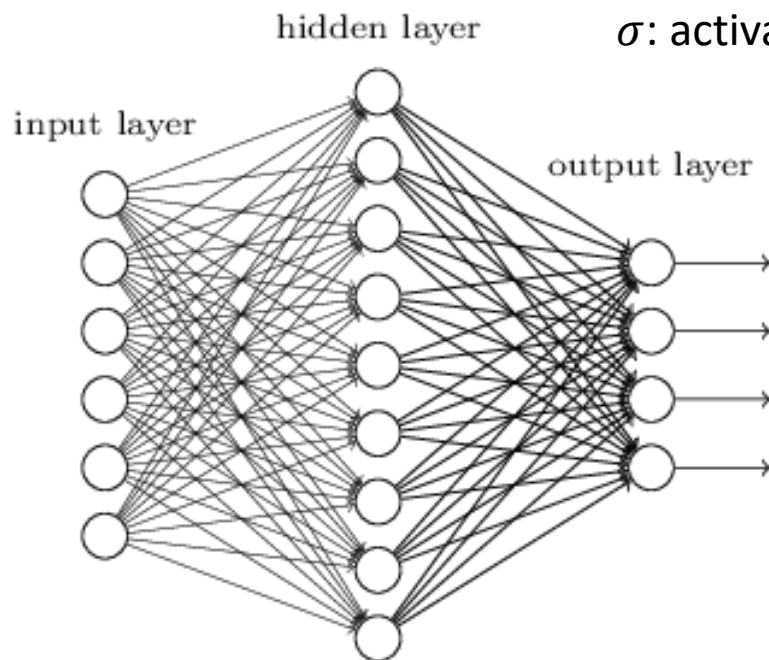
(a) 單層感知器的一個神經元



(b) 多層感知器的一個神經元

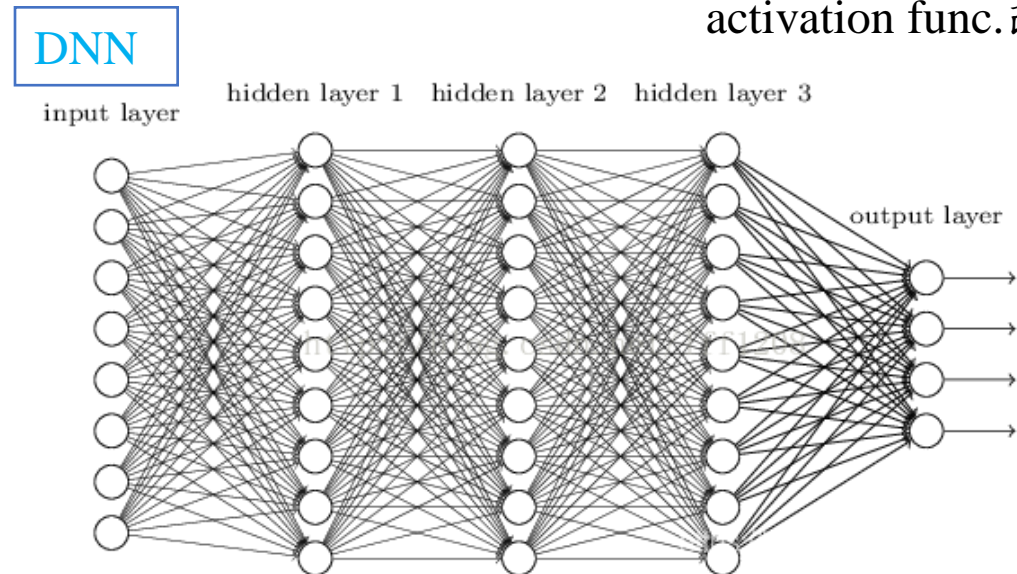
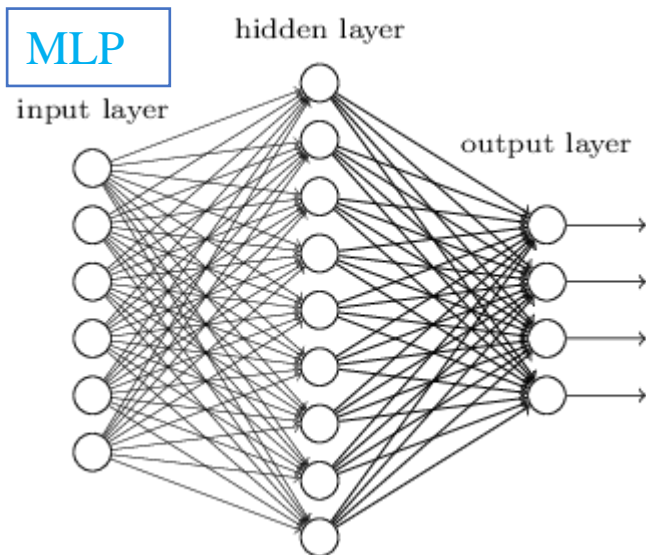
很多顆perceptron叫MLP

σ : activation function , sigmoid or tanh



演進

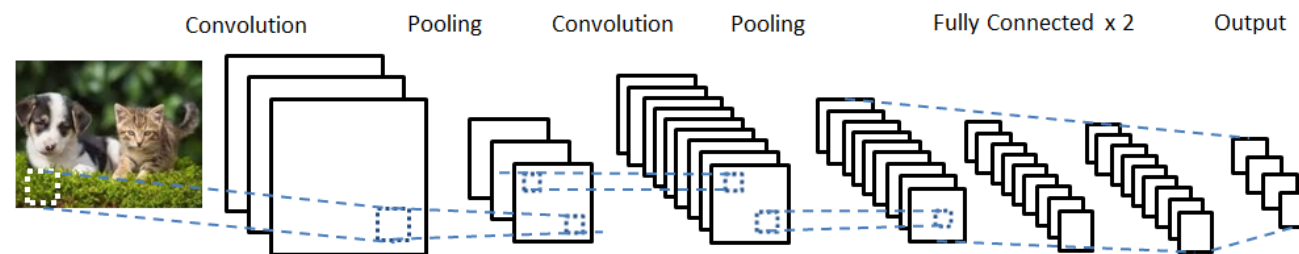
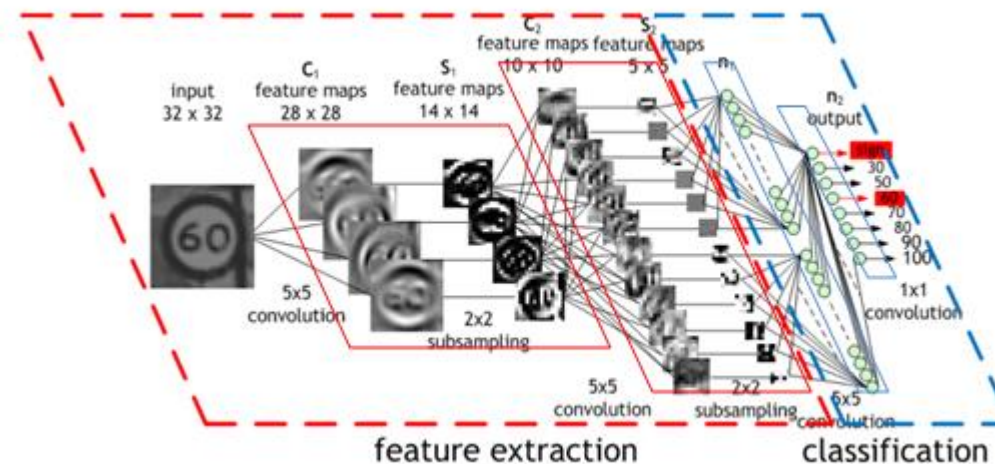
- Perceptron
- Multilayer Perceptron(MLP)
- Deep Neural Network(DNN)



很多層hidden layer叫DNN
activation func.改用relu

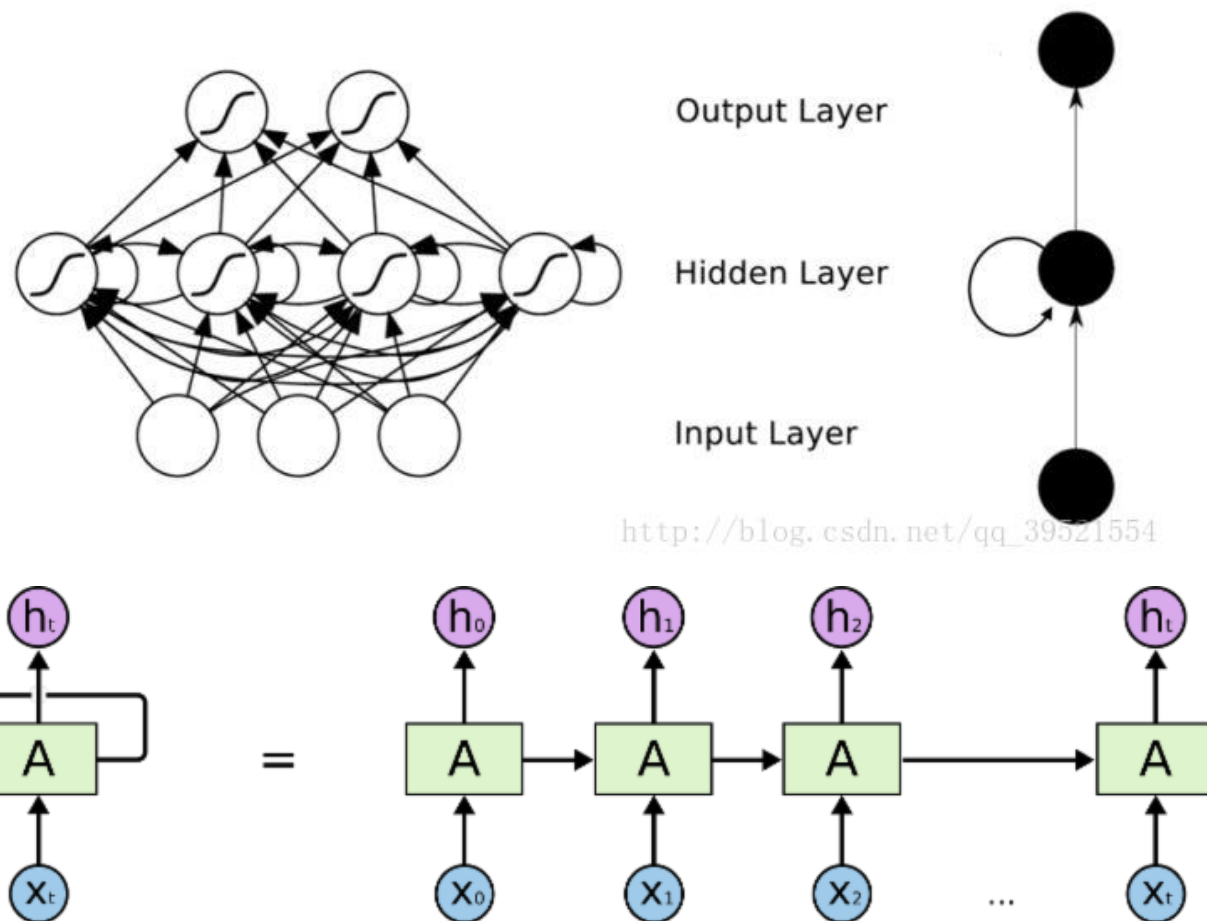
Convolutional Neural Network(CNN)

- 卷積層 (convolution layer)：利用隨機產生的遮罩進行特徵提取。
- 池化層 (pooling layer)：對不同位置的特徵進行統計，並取平均值或最大值作為最佳參考點，以減少資料特徵維度。
- 全連接層 (fully connected layer)：是將之前的卷積與池化後之結果進行平坦化，並接到最基本的神經網絡。



Recurrent Neural Network(RNN)

- 前面的神經元主要處理的輸入為，前一個輸入與後一個輸入是獨立沒有關係的。但是，某些任務前面的輸入會與後面的輸入有關係，具備時序的輸入，這時我們會需要RNN來幫忙。譬如自然語言處理。



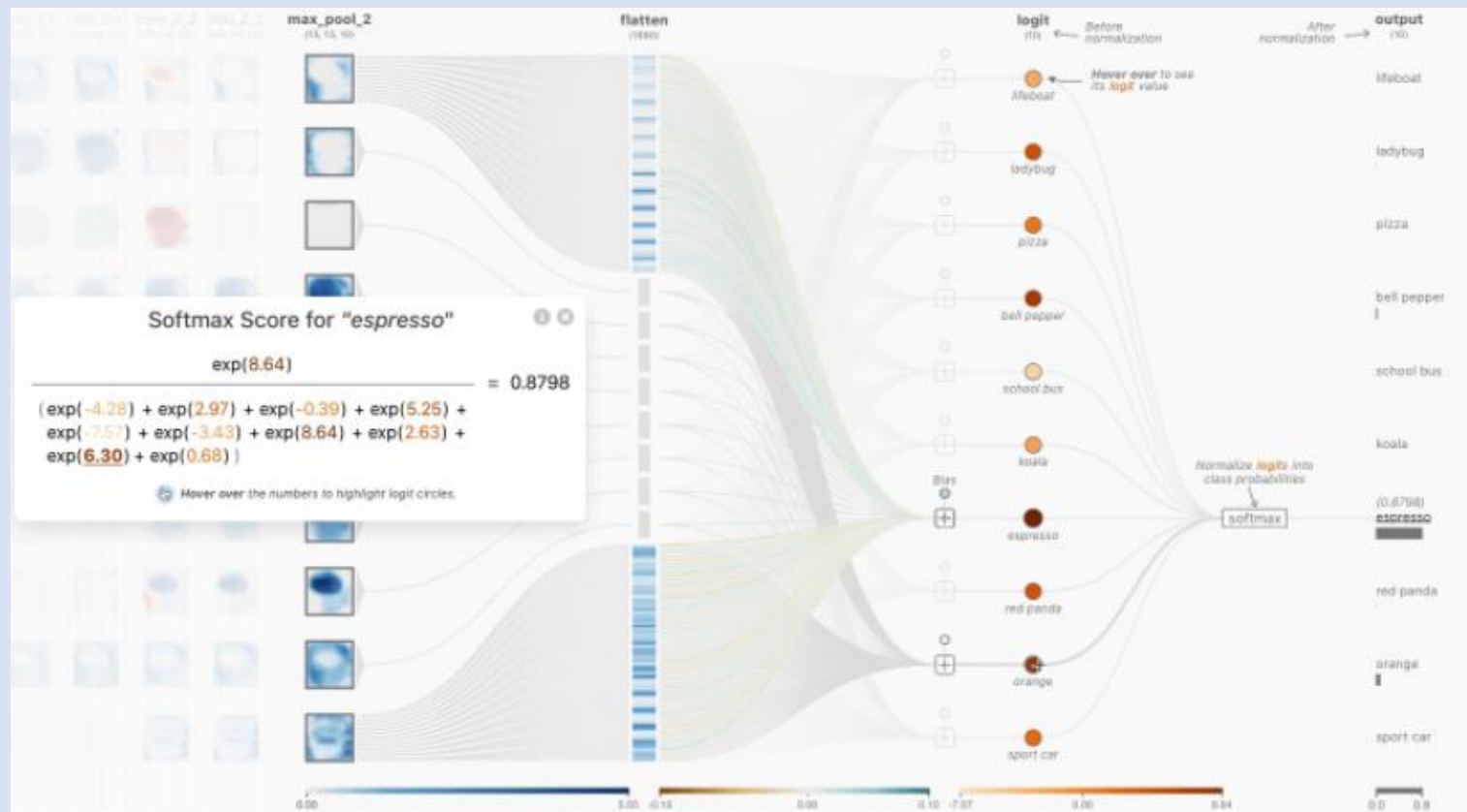
小結

- 機器學習為人工智慧的子領域，機器透過資料，尋找最佳的函數的方法
- 機器學習有三種學習方式，監督式學習、非監督式學習、強化式學習
- 其中監督式學習有迴歸與分類兩大任務
- 深度學習主要有三種神經網路，深度神經網路DNN、卷積神經網路CNN、循環神經網路RNN

CNN Explainer

網址：

<https://poloclub.github.io/cnn-explainer/>



實作1-感知神經網路

任務：使用iris資料集，利用perceptron做分類


Code前準備	二分類	三分類
envs 介紹 Colab操作	資料集 資料分析 資料預處理 訓練&評估 訓練可視化	訓練&評估 需自行修改： 資料集7:3 三類別

Code前準備

envs

- Using Colab 

- 框架：tensorflow 2.x 
TensorFlow

- 語言：python3.7~  python

- Package：sklearn, numpy, pandas, seaborn, matplotlib



Colab

- Google 帳號

二分類

任務：二分類，使用 petal length & sepal width 特徵

資料集：劃分8:2

資料集內容

- Labels
 - setosa: 山鳶尾
 - versicolor: 變色鳶尾
 - virginica: 維吉尼亞鳶尾
- Attributes(屬性)
 - Sepal length: 花萼長度 (cm)
 - Sepal width: 花萼寬度 (cm)
 - Petal length: 花瓣長度 (cm)
 - Petal width: 花瓣寬度 (cm)



二分類-評估指標

- 分類setosa與versicolor兩類
- (在這翻譯為是setosa與不是setosa)

實際標籤	setosa	TP	FN
	versicolor	FP	TN
		setosa	versicolor
		預測標籤	

$$Accuracy_i = \frac{TP + TN}{TP + FP + TN + FN}$$

模型判斷出的預測結果且實際分類正確的setosa、versicolor與整體之比？

紅色為不是setosa

黃色為實際上是setosa

TP (True Positive) :

預測為是，實際上為是

FP (False Positive) :

預測為是，實際上為不是

FN (False Negative) :

預測為不是，實際上為是

TN (True Negative) :

預測為不是，實際上為不是

三分類

任務：三分類，兩特徵petal length & petal width

資料集：指定劃分7:3

提示：

- 程式中的x[]輸入特徵
- sklearn專門切資料集的指令
- 部分二分類專用改三分類專用的註解拿掉



謝謝聆聽