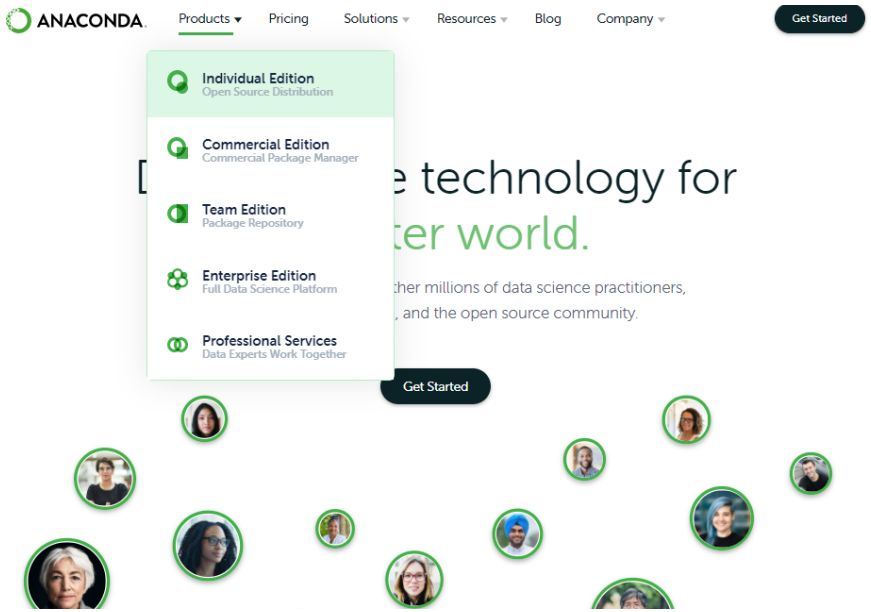
Lesson：Python 環境安裝

南分院 人工智慧應用部(0T200)

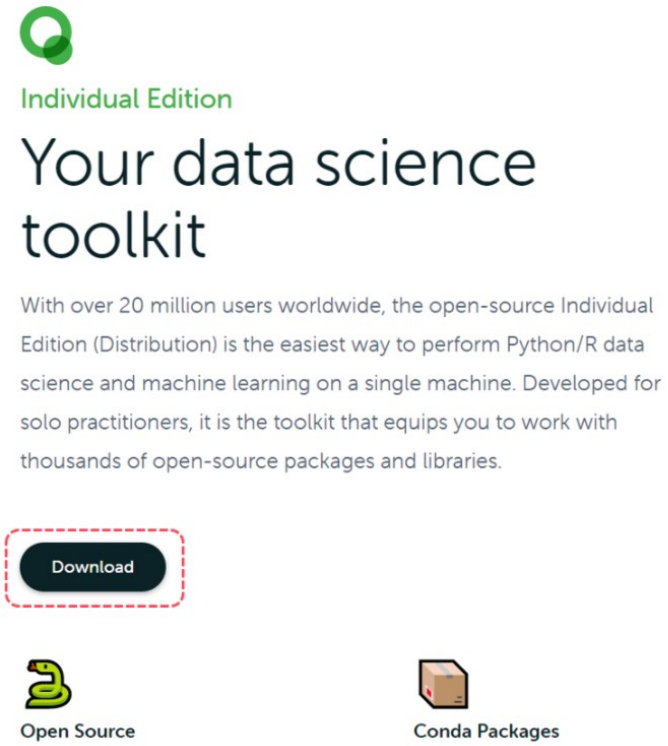
2021/04/12

python環境部屬

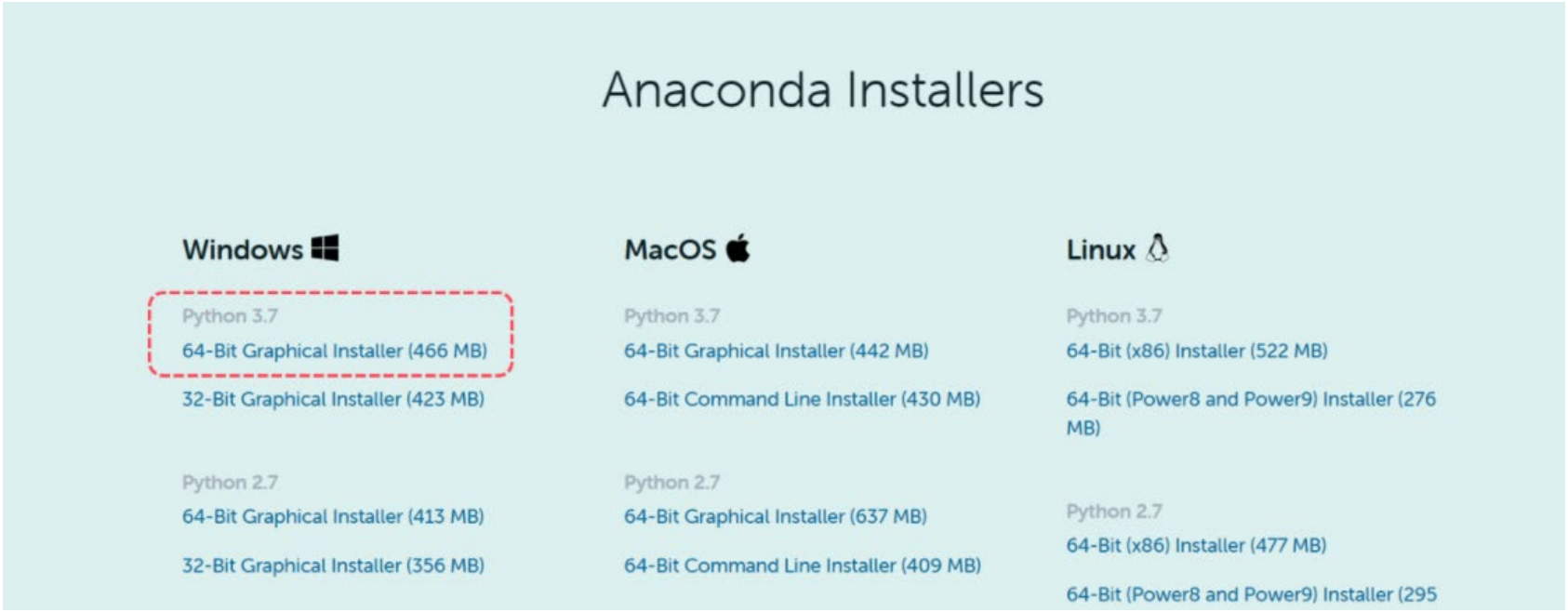
- 1. Anaconda官網(https://www.anaconda.com/)，點選上方 Products/Individual Edition



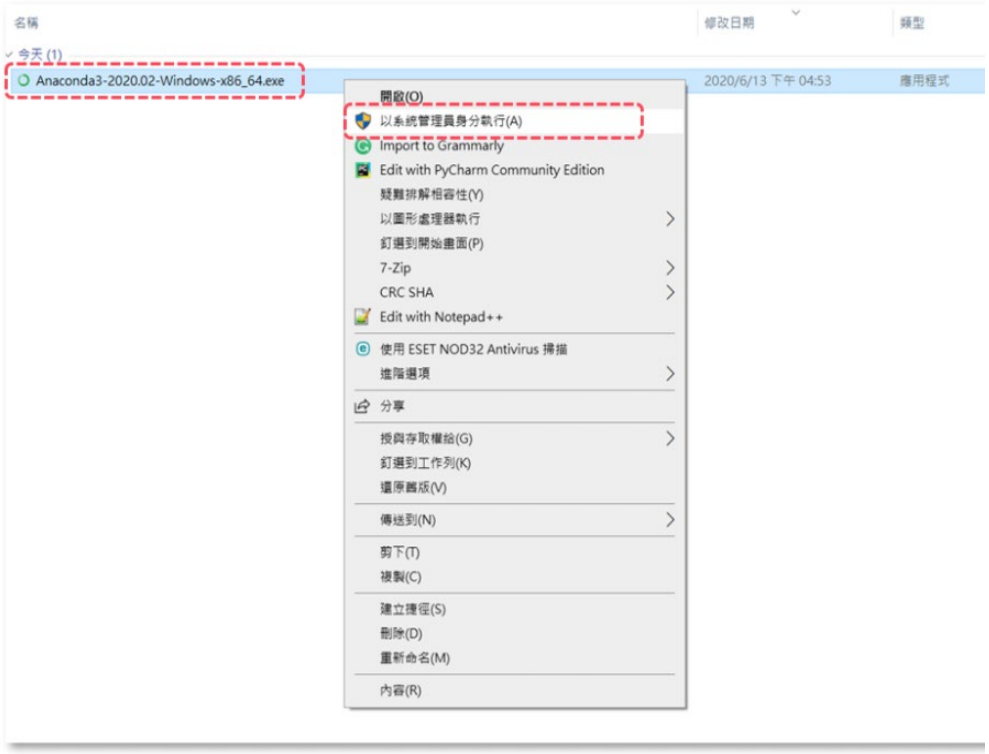
python環境部屬：點選「Download」



python環境部屬：選擇適合您的OS，點選anaconda installer

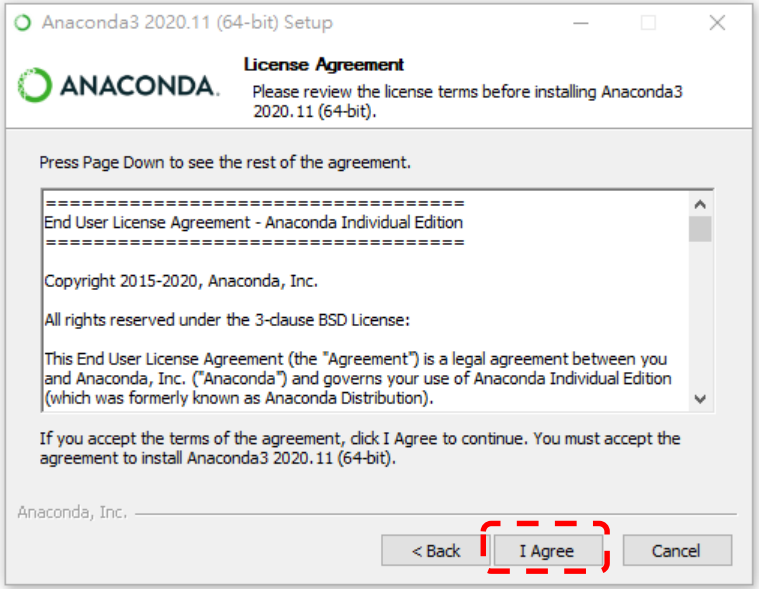


執行下載的 Anaconda installer

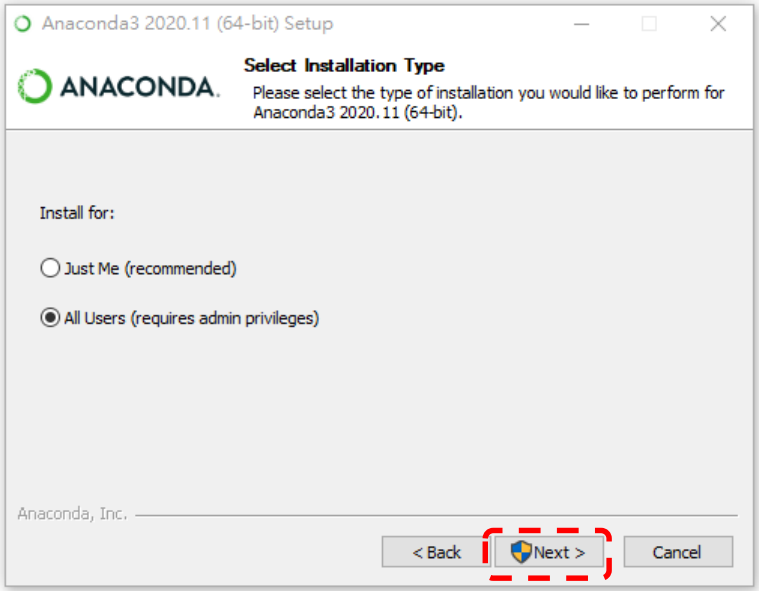


開始安裝點選「Next」

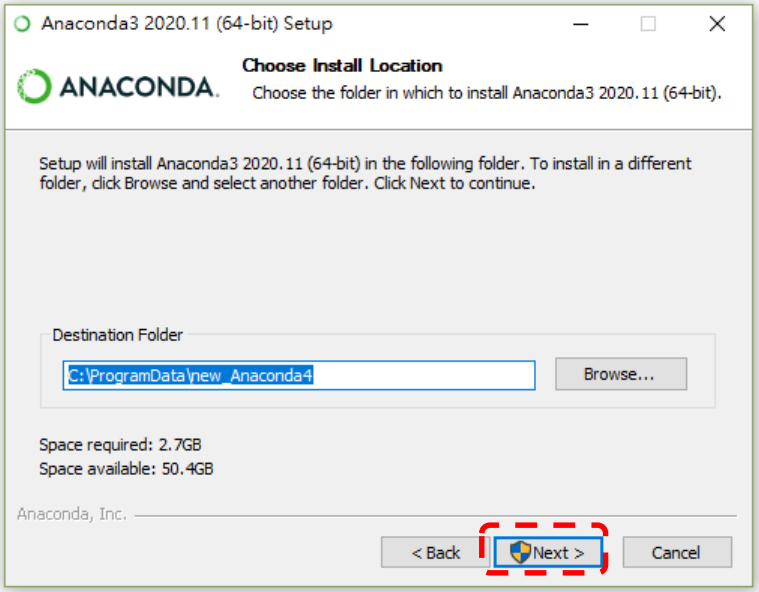
閱讀許可協議(License Agreement)後，點選「I Agree」



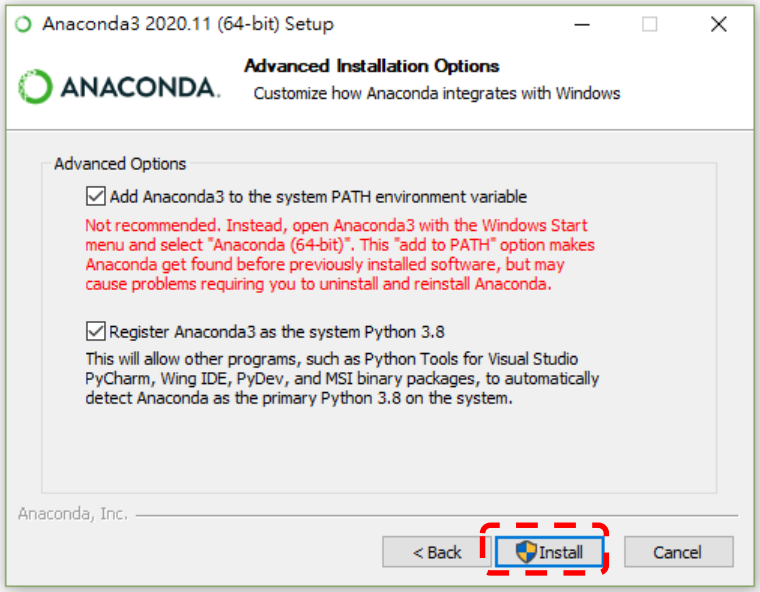
選擇安裝對象



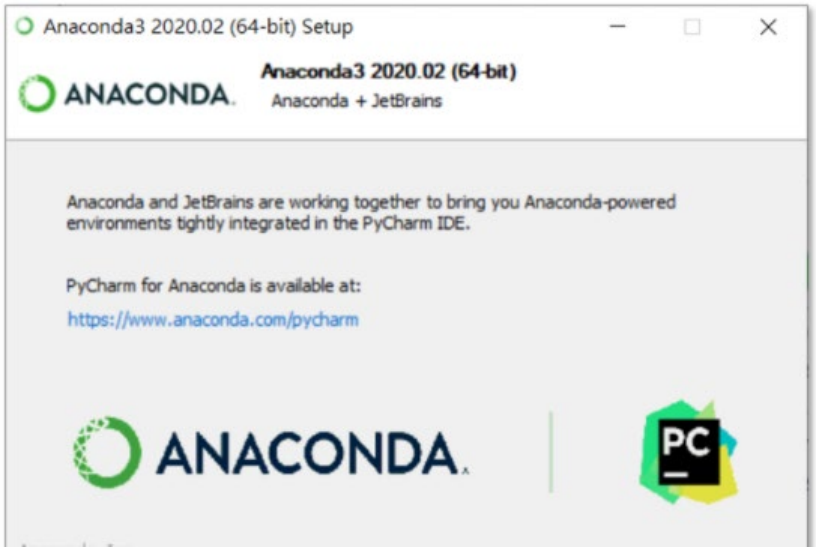
確認安裝路徑

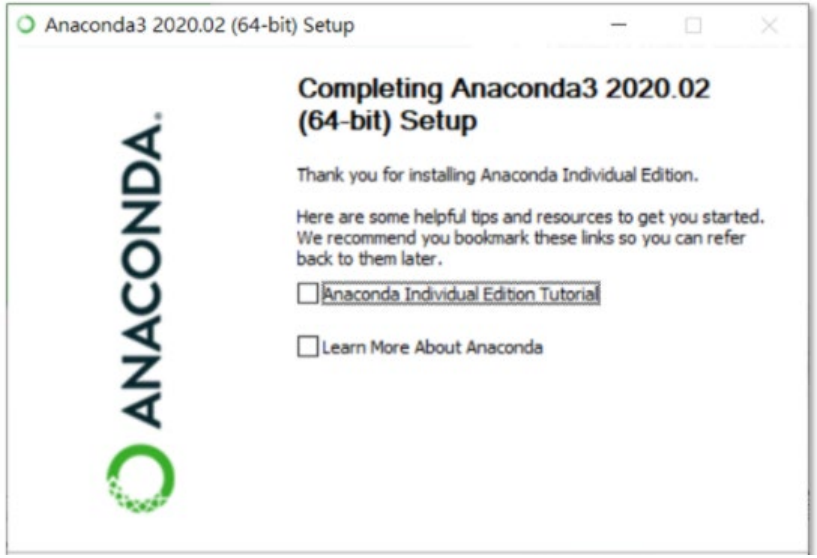


選擇是否要將 Anaconda 加到 PATH 環境變數中

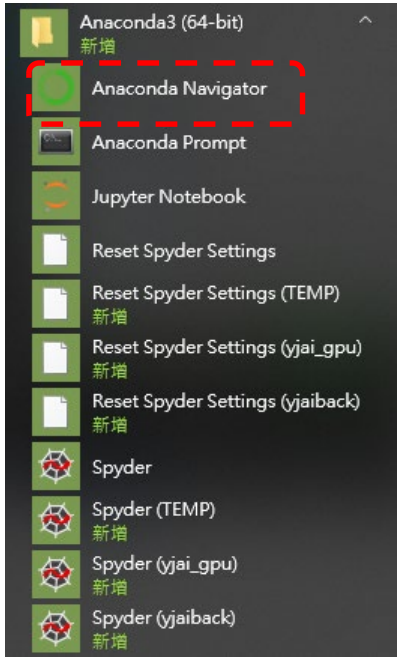


安裝完成

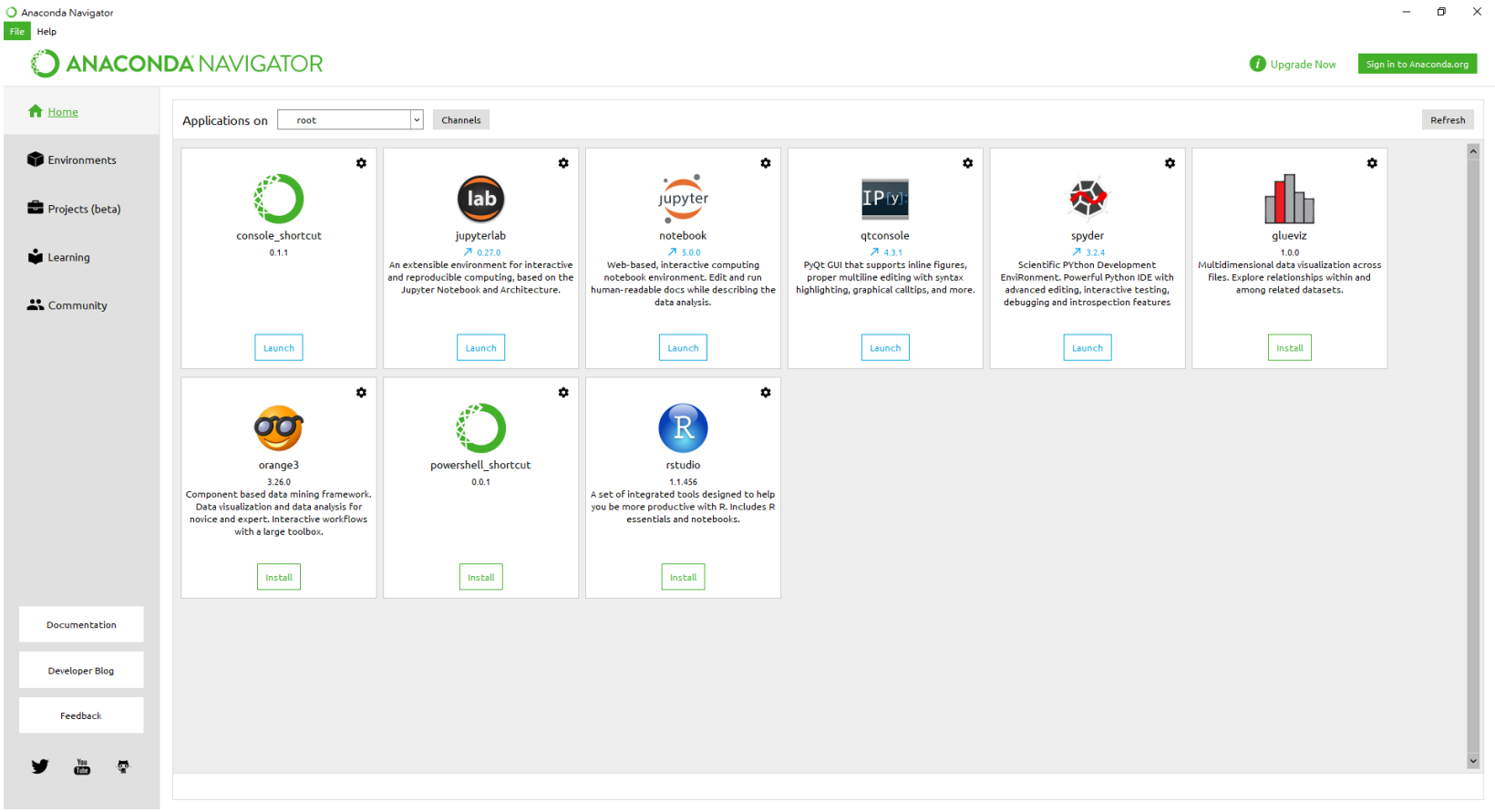




驗證安裝是否完成



安裝完成

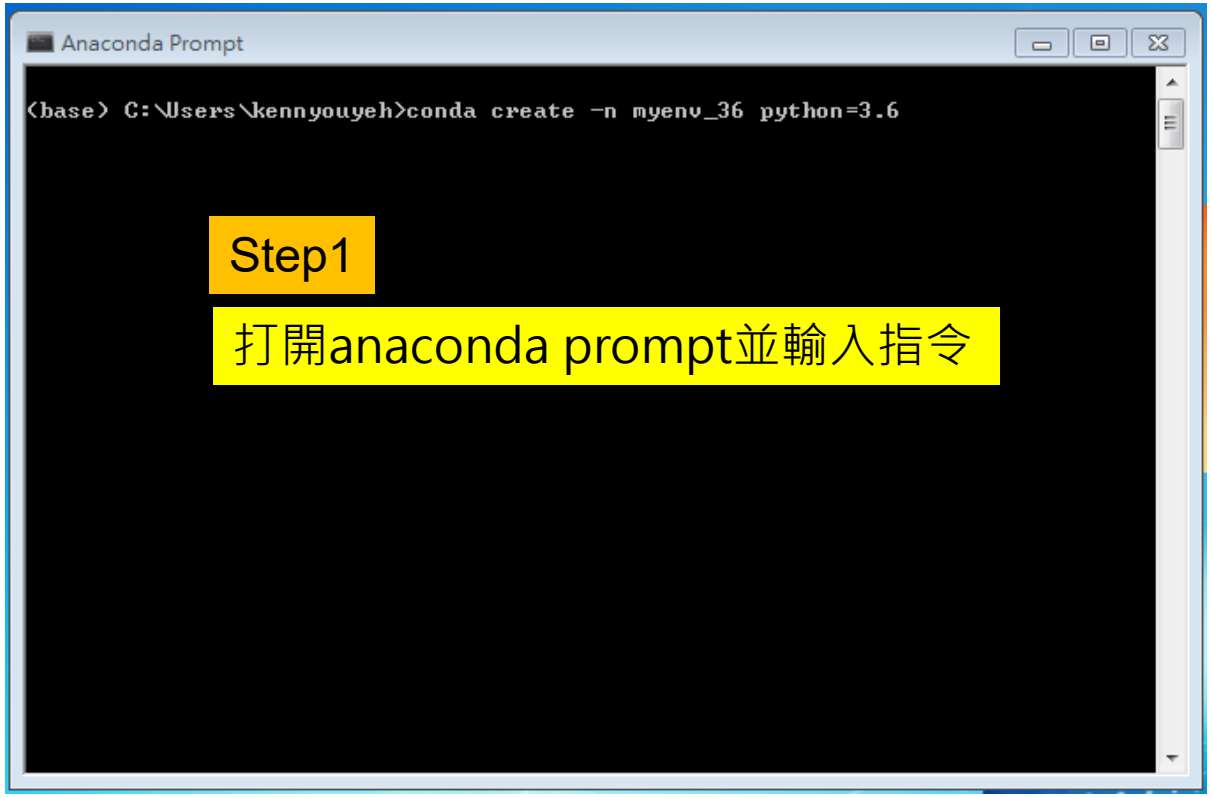

Anaconda Navigator
File Help
ANACONDA NAVIGATOR
Upgrade Now
Sign in to Anaconda.org
Home
Environments
Projects (beta)
Learning
Community
Documentation
Developer Blog
Feedback
Applications on root
Channels
Refresh
console_shortcut
0.1.1
Launch
jupyterlab
0.27.0
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch
notebook
5.0.0
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch
qtconsole
4.3.1
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch
spyder
3.2.4
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch
glueviz
1.0.0
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install
orange3
3.26.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install
powershell_shortcut
0.0.1
Install
rstudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.
Install

虛擬環境架設

- 建立新的虛擬環境
  - 指令：conda create -n myenv_36 python=3.6
- 備註
  - 一定要連網路

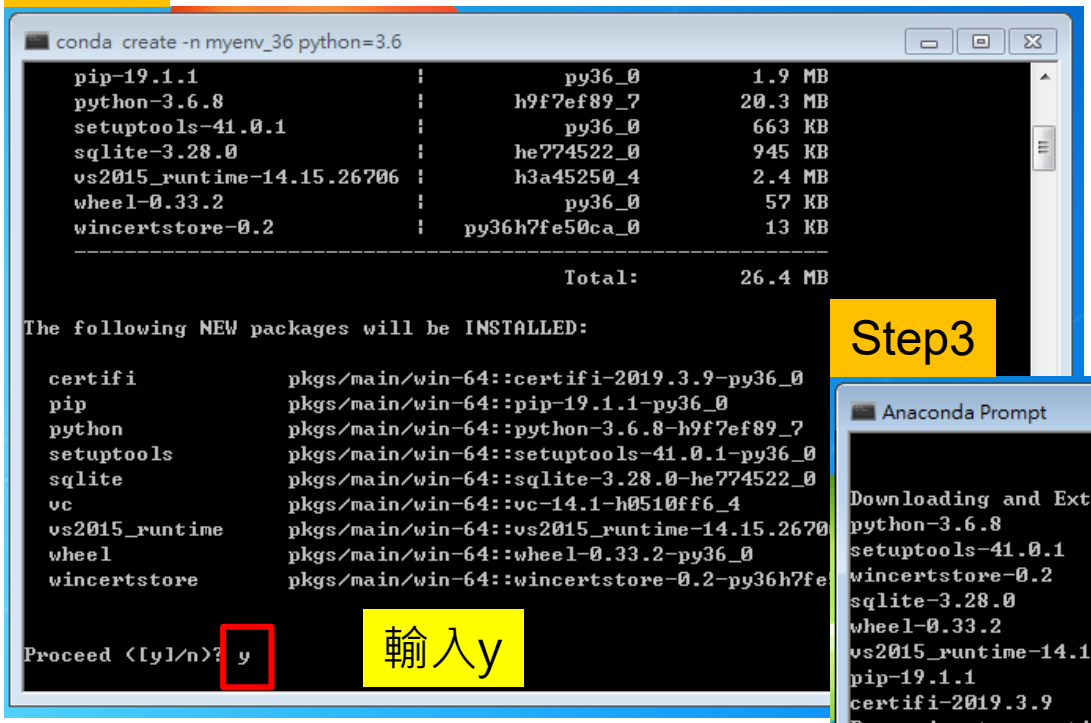
虛擬環境所使用的python版本

虛擬環境名稱

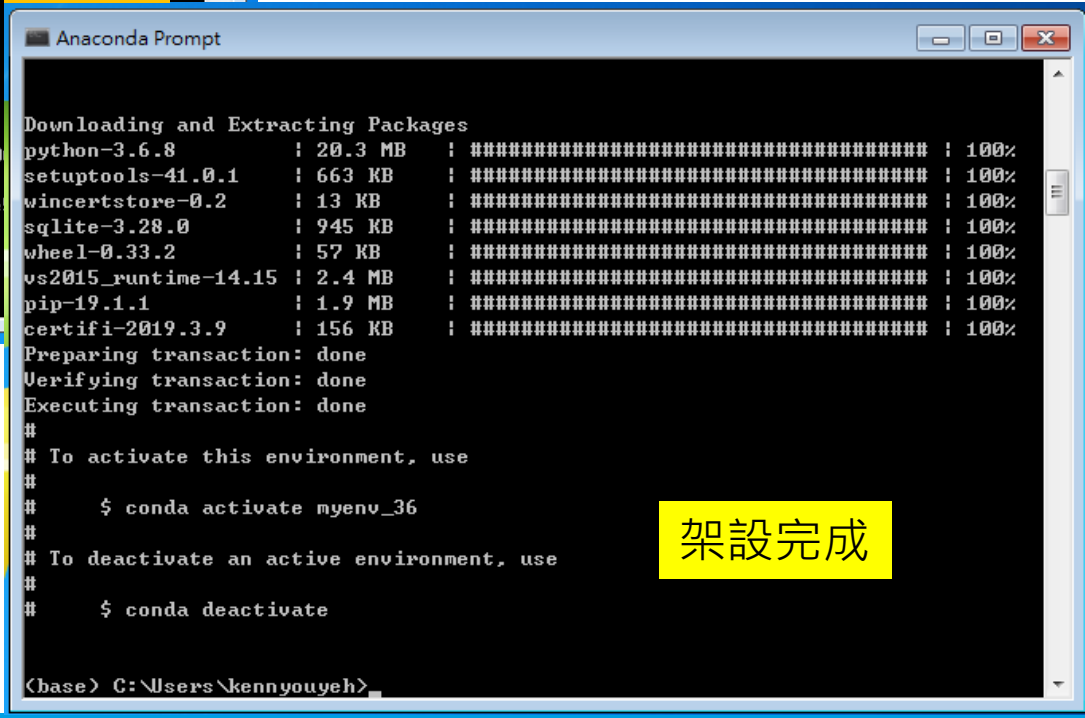


虛擬環境架設(2)

Step2

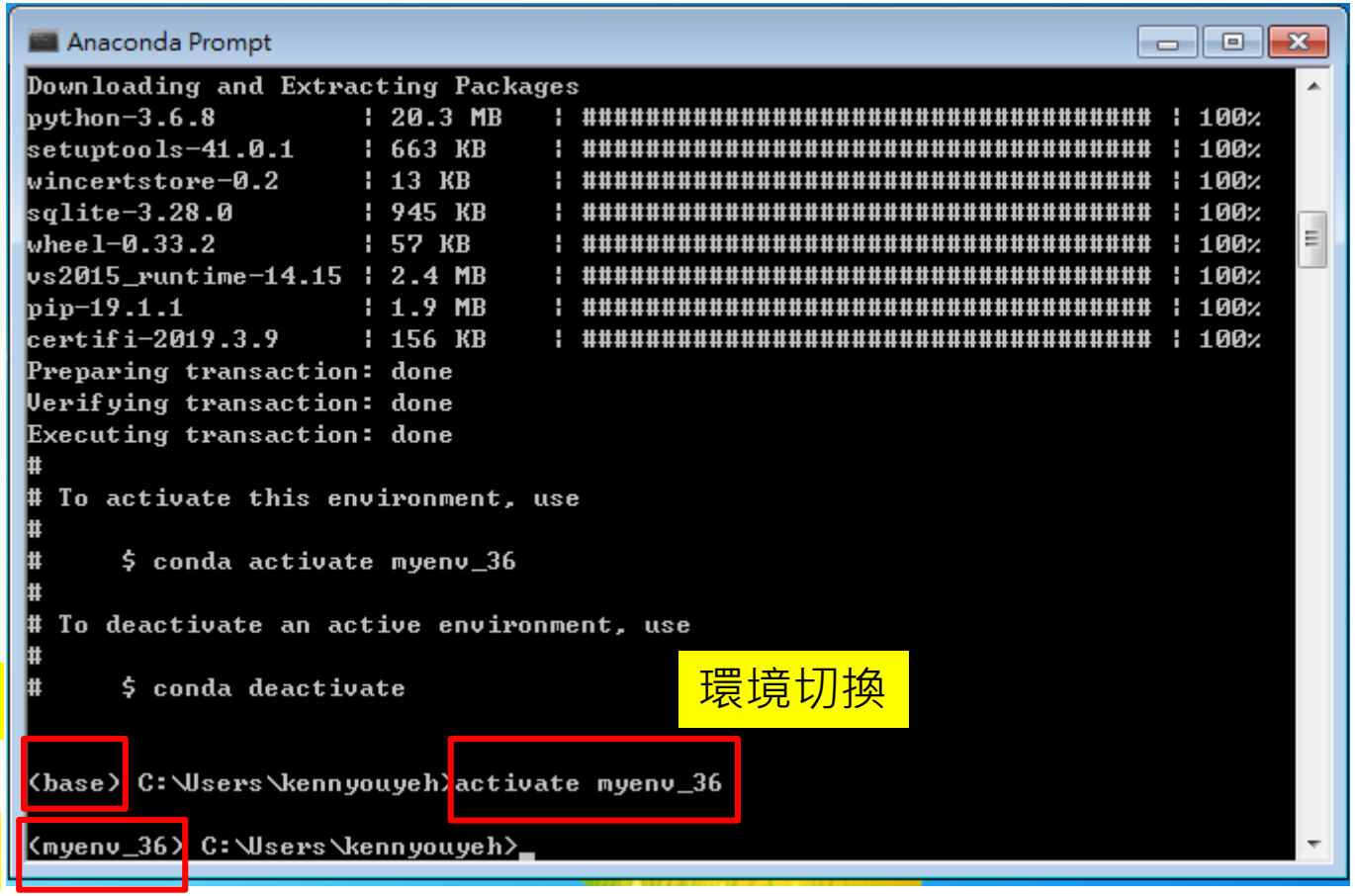


Step3



虛擬環境架設(3)

- 啟用虛擬環境
  - 指令：activate myenv_36



切換前在base底下

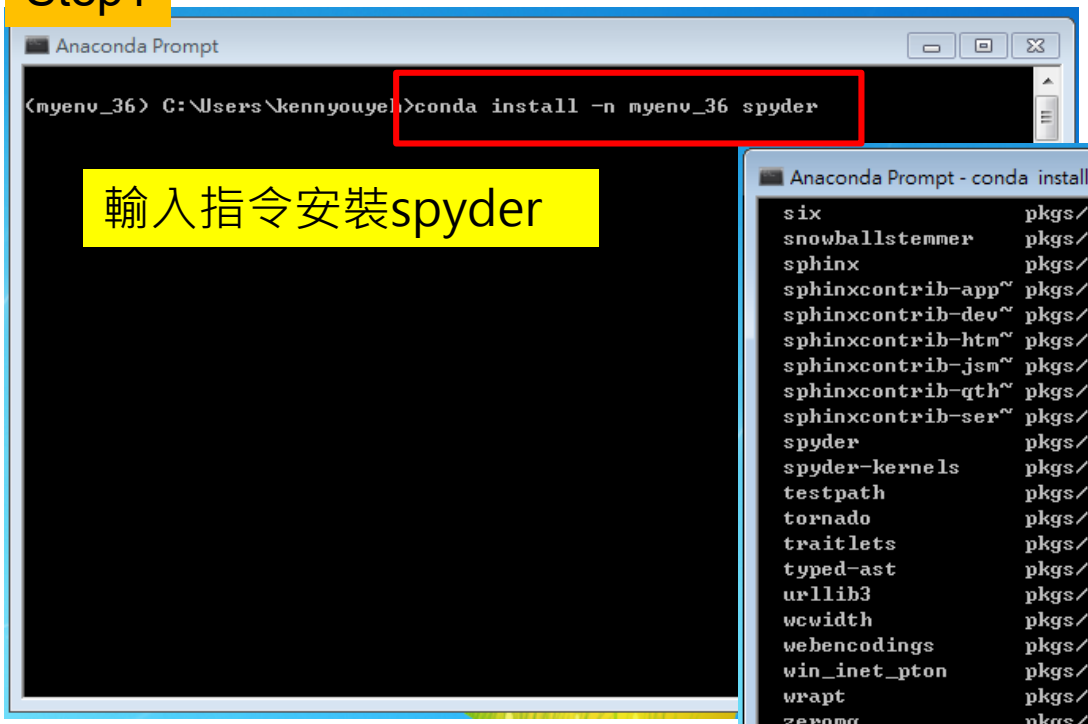
環境切換

切換後

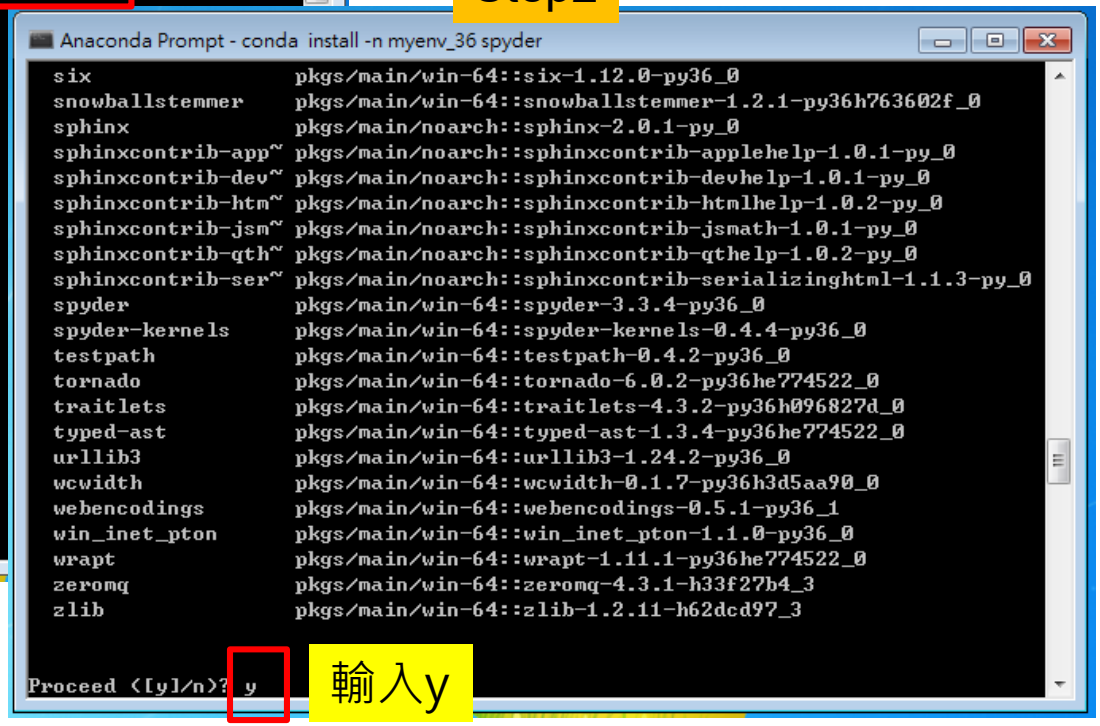
安裝IDE(1)

- 安裝IDE – spyder
  - 指令： conda install -n myenv_36 spyder

Step1

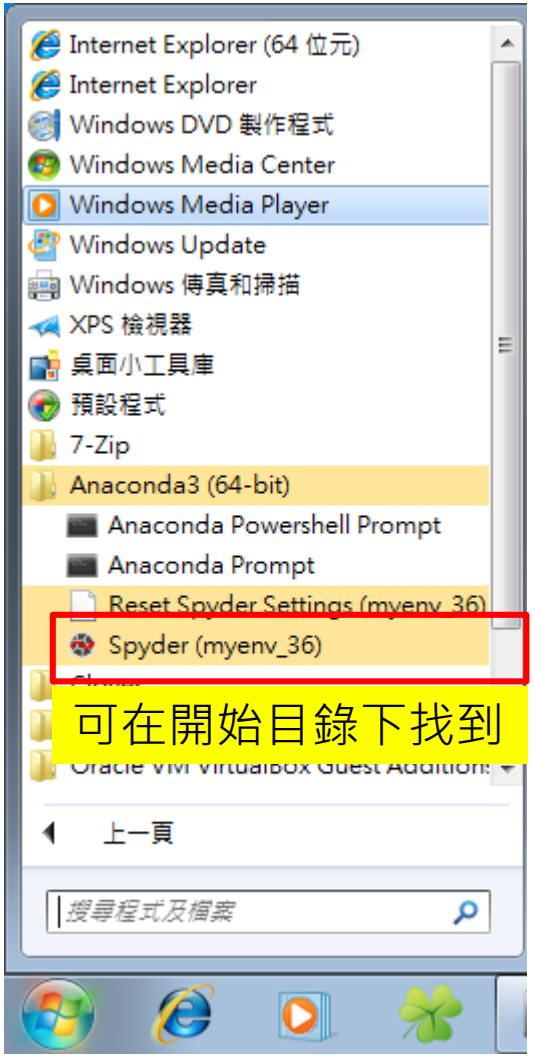


Step2



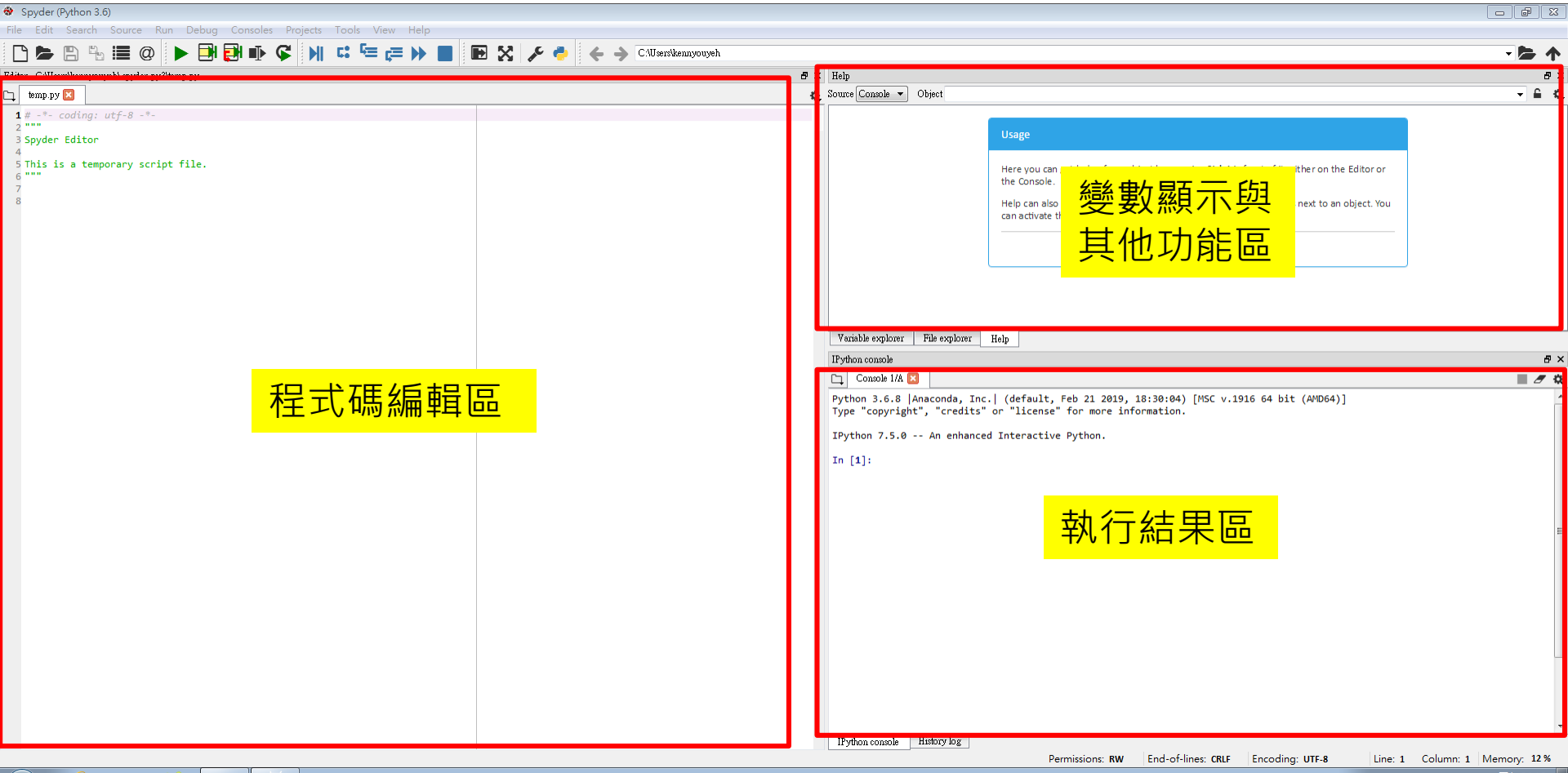
安裝IDE(2)


使用者帳戶控制
您是否要允許下列來自不明發行者的程式變更這部電腦?
程式名稱: python.exe
發行者: 不明
檔案來源: 此電腦上的硬碟
顯示詳細資料(D)
是(Y)
否(N)
變更這些通知顯示的時機
授權允許安裝


Internet Explorer (64 位元)
Internet Explorer
Windows DVD 製作程式
Windows Media Center
Windows Media Player
Windows Update
Windows 傳真和掃描
XPS 檢視器
桌面小工具庫
預設程式
7-Zip
Anaconda3 (64-bit)
Anaconda Powershell Prompt
Anaconda Prompt
Reset Spyder Settings (myenv_36)
Spyder (myenv_36)
上一頁
搜尋程式及檔案
可在開始目錄下找到

安裝IDE(3)

- Spyder打開後的畫面

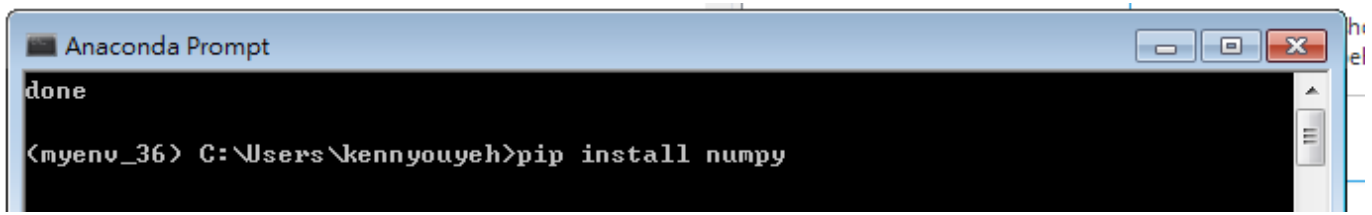


套件安裝

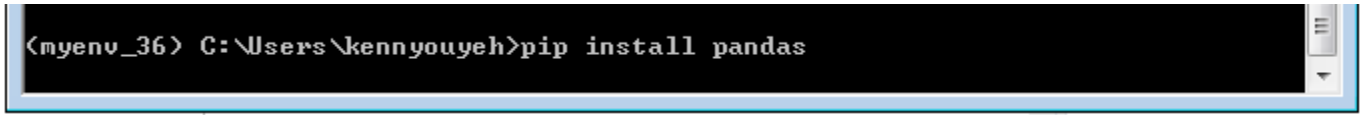
- 使用pip來管理套件
  - 範例指令：pip install numpy

套件名稱

安裝套件numpy



安裝套件pandas



其餘參考指令(1)

查詢目前建置的所有虛擬環境

- conda info –envs

切換虛擬環境

On Windows, in your Anaconda Prompt, run

- activate myenv_36

On macOS and Linux, in your Terminal Window, run

- source activate myenv_36

切換回root

- activate root

複製虛擬環境

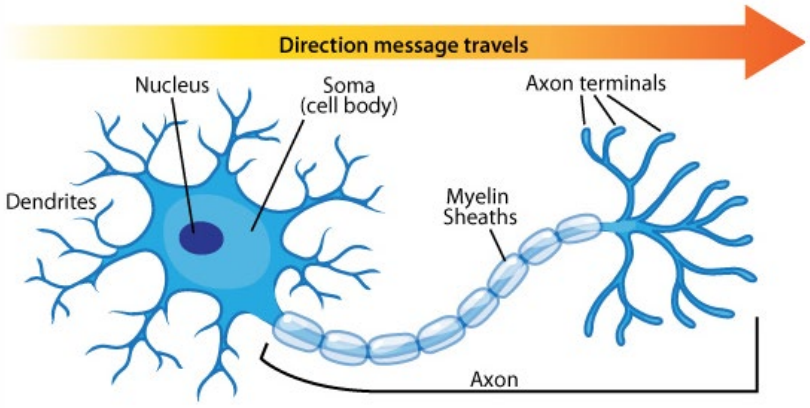
- conda create --name myenv_36_v2 --clone myenv_36

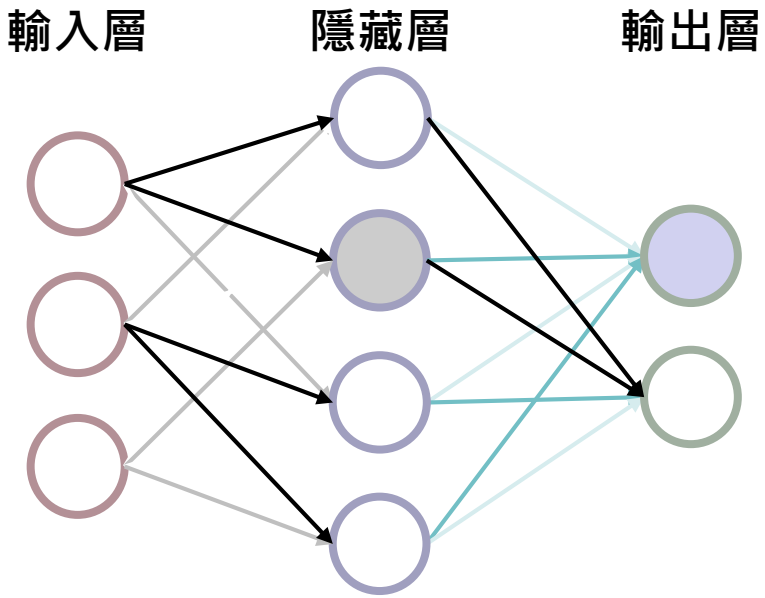
刪除虛擬環境

- conda env remove --name oc_36

神經網路模型

- 仿生物神經網路的結構與功能所產生的數學模型
- 接收脈波訊號後，神經細胞核對訊息的處理模式，可分：
  - 將收集到的訊號作加總
  - 非線性轉換
  - 產生一個新的脈波信號傳遞到下個神經細胞
- 由多個神經元 (neuron) 所組成，且神經元間彼此有不同強烈程度的連結
- 神經網路架構
  - 輸入層：解釋變數 X = (*t*, *s*)
  - 隱藏層：決定模型複雜度
  - 輸出層：反應變數 y
  - 活化函數：非線性轉換
  - 連結權重：模型參數





其餘參考指令(2)

安裝套件
latest version
- pip install SomePackage

specific version，安裝套件的指定版本，下方程式碼為安裝1.0.4版本
- pip install SomePackage==1.0.4

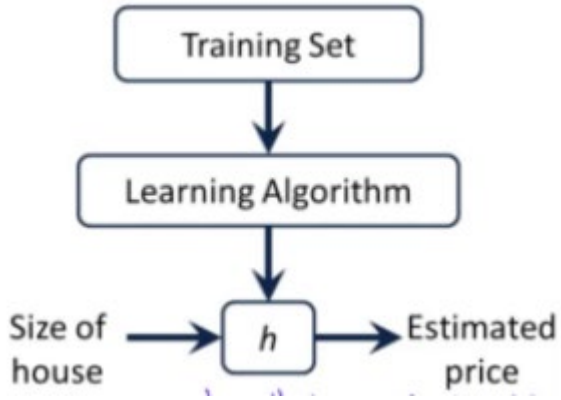
minimum version
- pip install 'SomePackage>=1.0.4'

透過Wheel安裝套件
wheel 本質上是一個 zip 包格式，它使用 .whl 擴展名，用於 python套件的安裝
- pip install SomePackage.whl

移除套件
- pip uninstall SomePackage

查詢套件版本與其他詳細資料
- pip show SomePackage

Linear Regression：房價預測問題



output

Input

price($) (y)	size in feet^2 (x)
463	2114
231	1405
325	1734
188	862

Hypothesis:

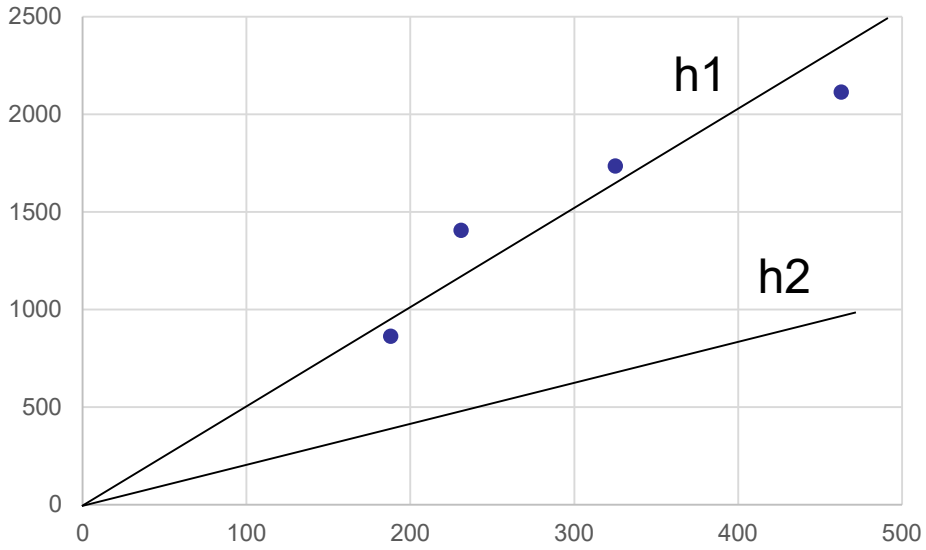
$$h_\theta(x) = \theta_0 + \theta_1 x$$

How to choose?

Parameters:

θ_0, θ_1

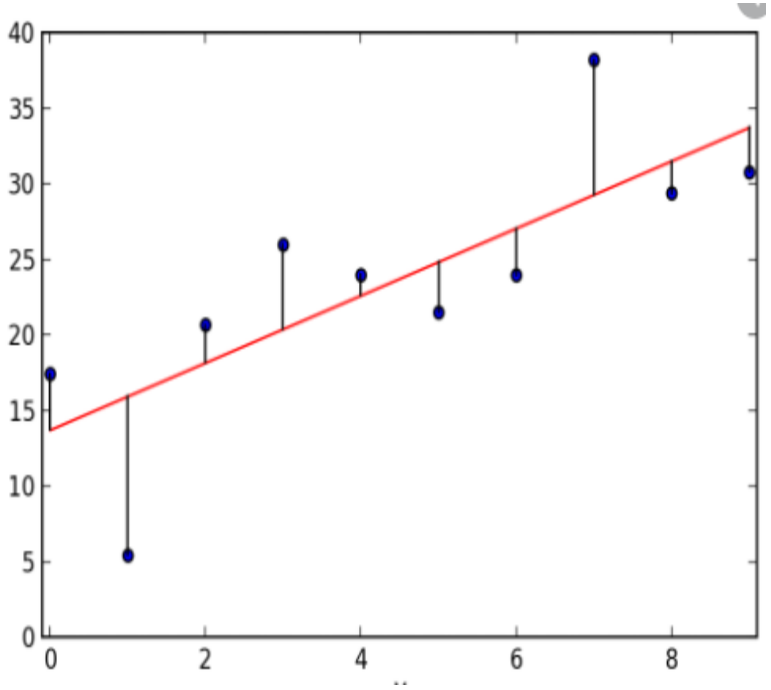
cost function

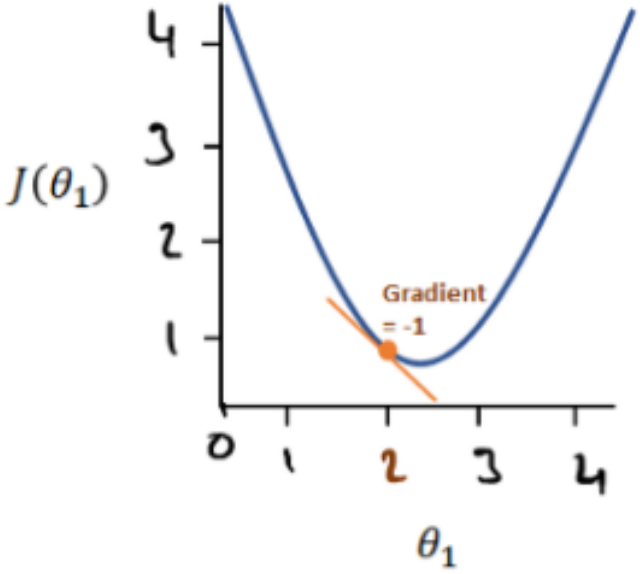


Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

cost function





$$h_\theta(x) = \theta_0 + \theta_1 x$$

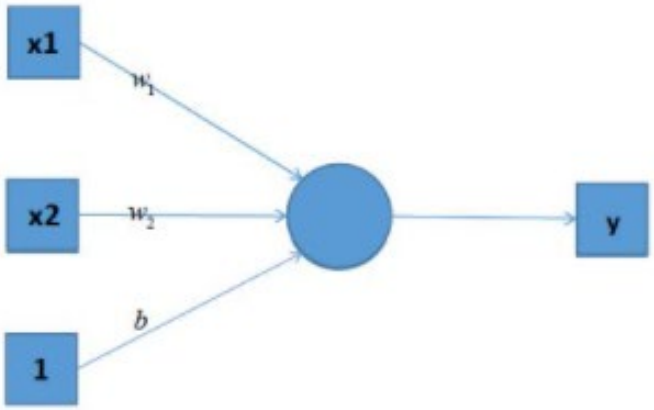
$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\theta_1 x_i - y_i)^2$$

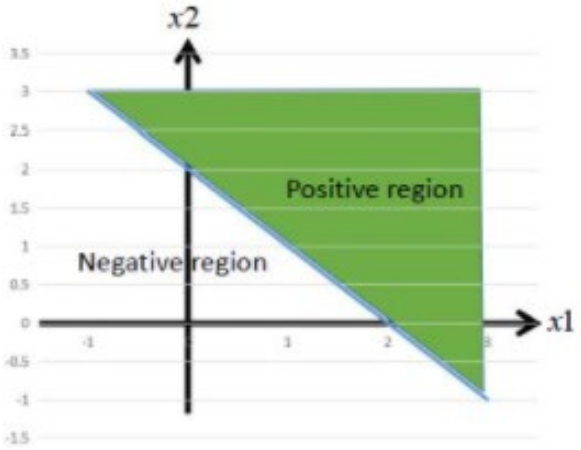
$$MSE = \frac{1}{N} \sum_{i=1}^{N} (\tilde{Y}_i - Y_i)^2$$

誤差越小越好

Activate function

Perceptron

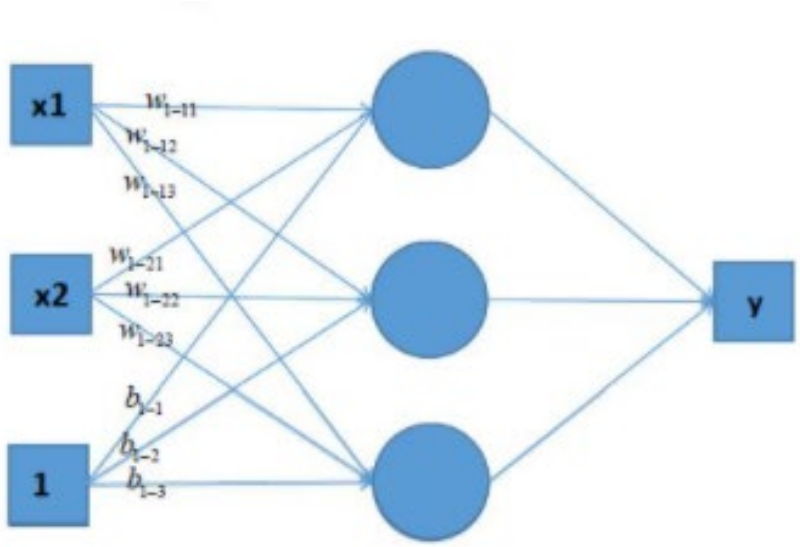




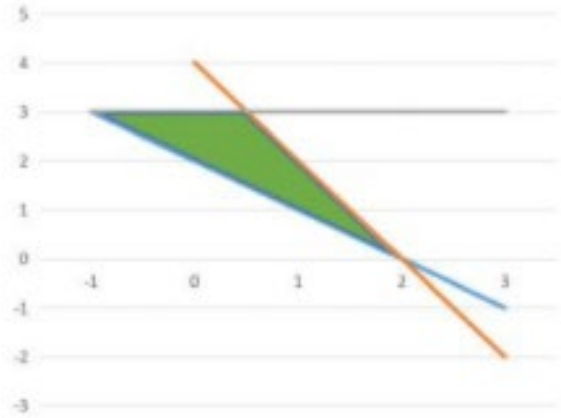
$w_1 = 1, w_2 = 1, b = -2$

$$y = w_1 x_1 + w_2 x_2 + b$$

Activate function



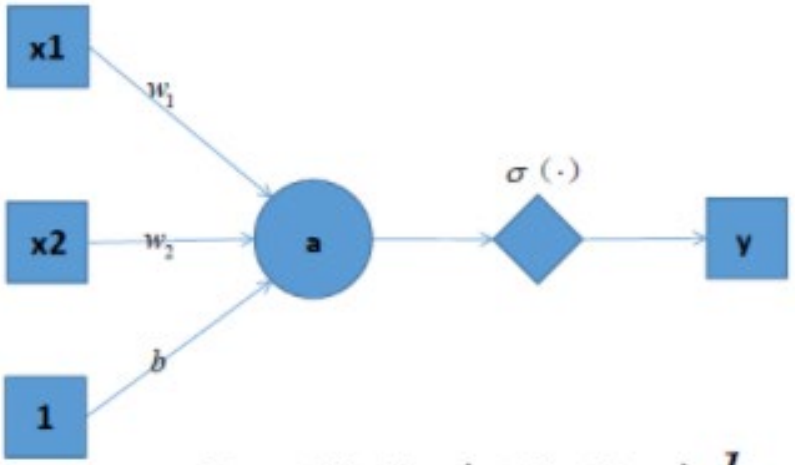
linear combination of three decision lines



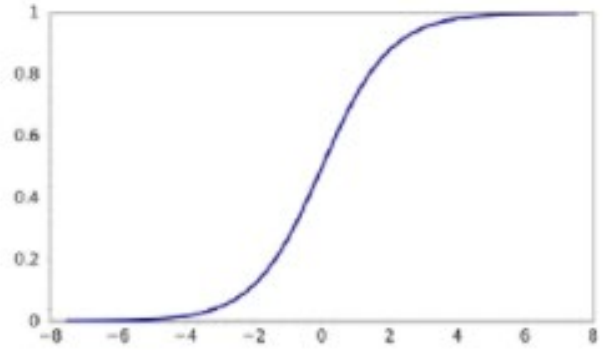
$$w_{1-11}=1, w_{1-12}=1, b_{1-1}=-2$$
$$w_{1-21}=2, w_{1-22}=1, b_{1-2}=4$$
$$w_{1-31}=0, w_{1-32}=1, b_{1-3}=3$$

$$y = w_{2-1}(w_{1-11}x_1 + w_{1-21}x_2 + b_{1-1}) + w_{2-2}(w_{1-12}x_1 + w_{1-22}x_2 + b_{1-2}) + w_{2-3}(w_{1-13}x_1 + w_{1-23}x_2 + b_{1-3})$$

Activate function -sigmoid



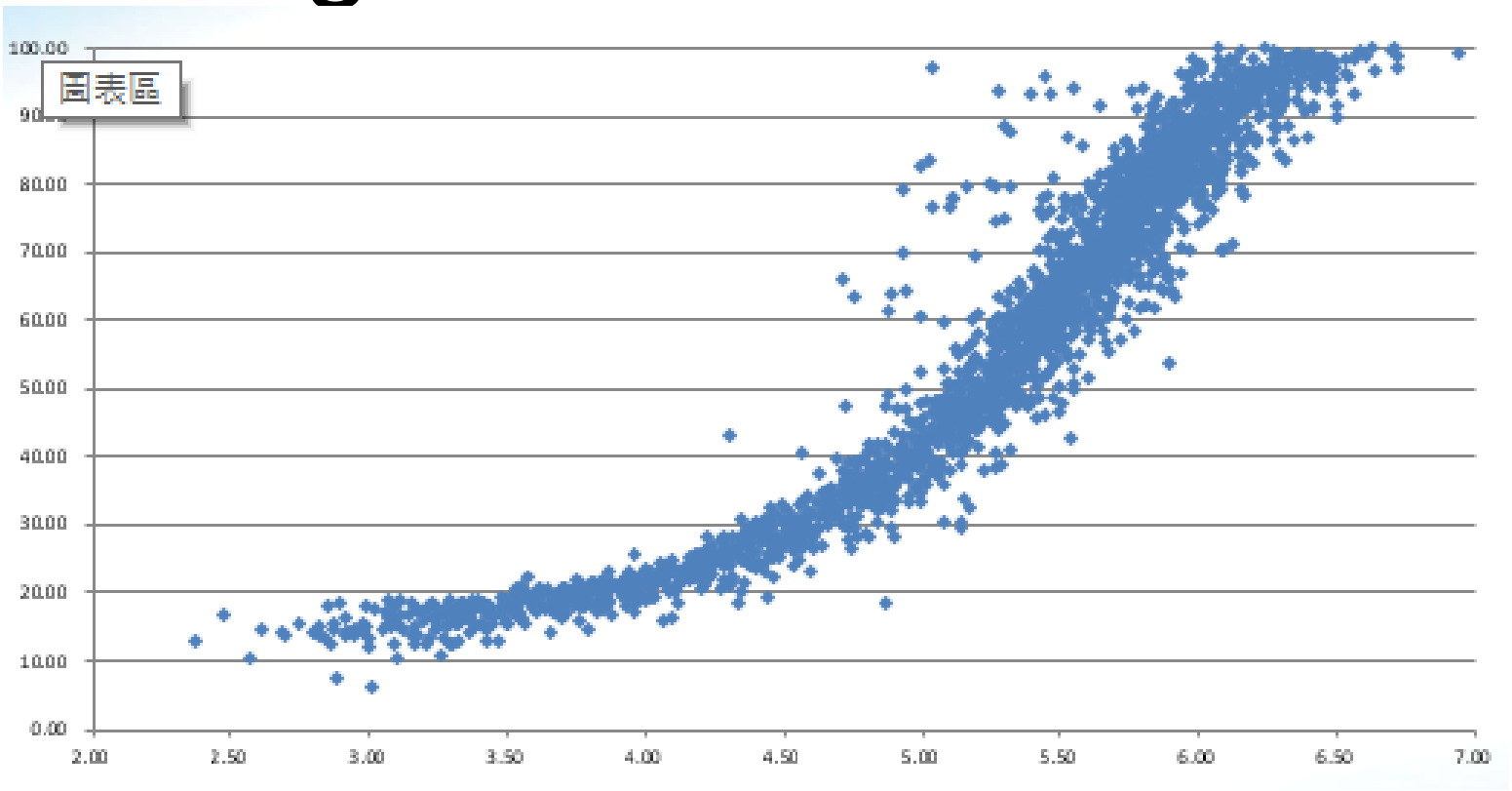
$$a = w_1 x_1 + w_2 x_2 + b$$
$$y = \sigma(a)$$



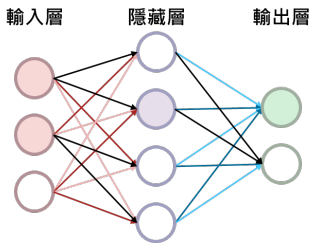
$\sigma(\cdot)$ is a non-linear activation function, sigmoid was the most popular one,

$$\sigma(y) = \frac{1}{1+e^{-y}}$$

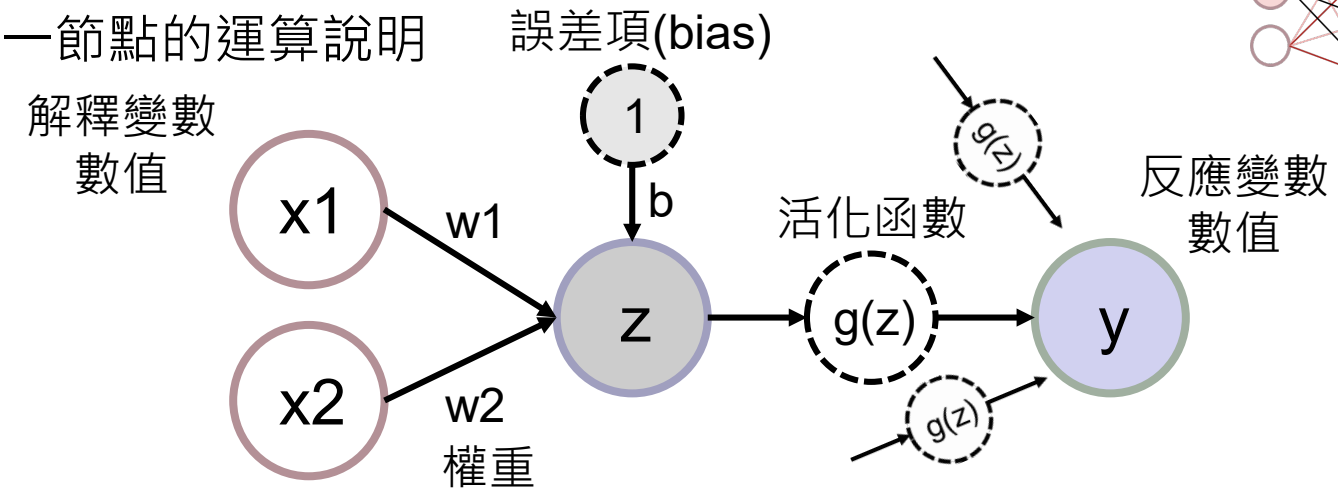
Activate function - sigmoid



神經網路模型 (2)



- 取單一節點的運算說明

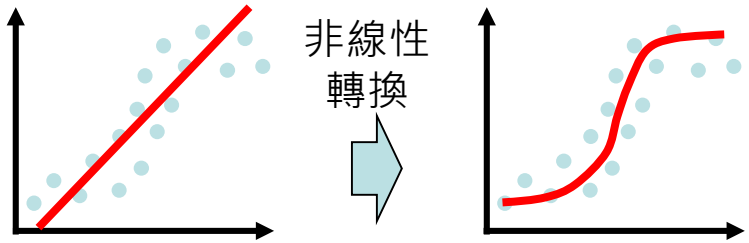


- 輸入數值的線性組合運算
- 類似於線性迴歸模型

- 是否所有前後關聯皆能夠以線性完整表示?

- 活化函數 activation function

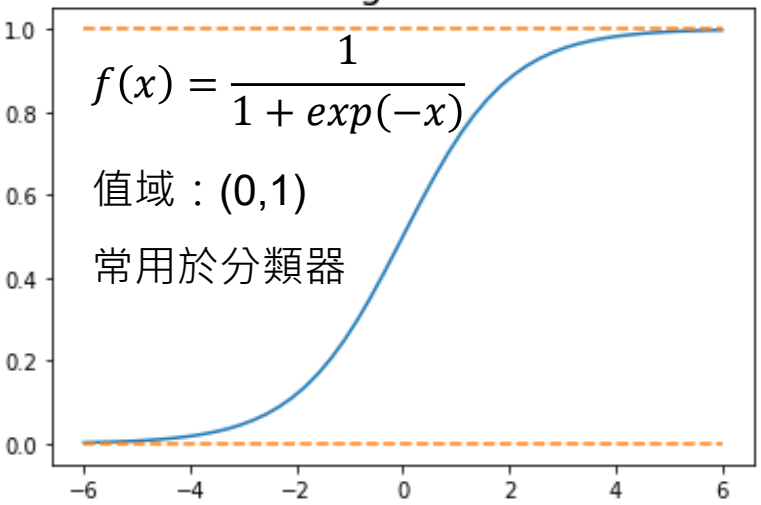
$y = g(z)$



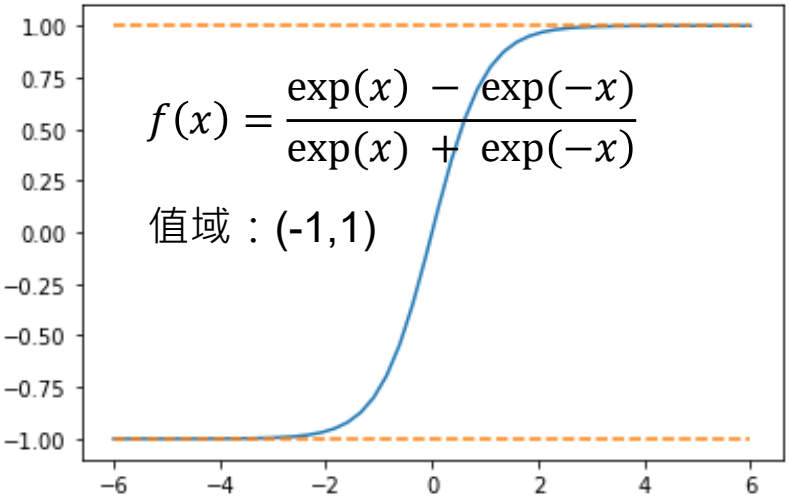
Activate function

- 活化函數的種類

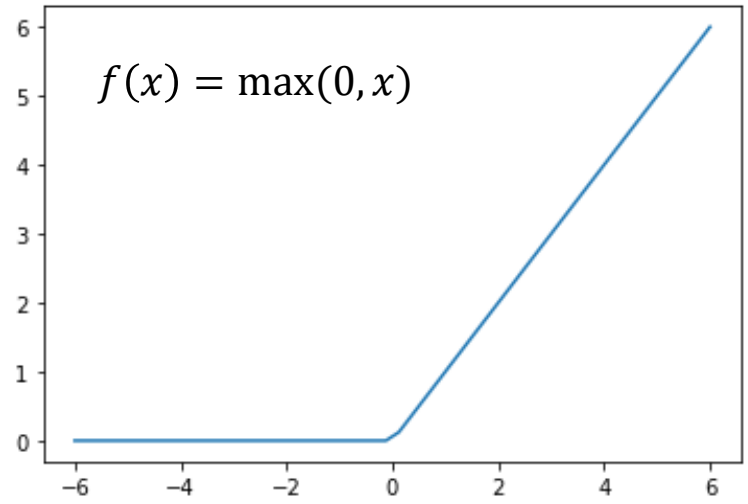
Sigmoid



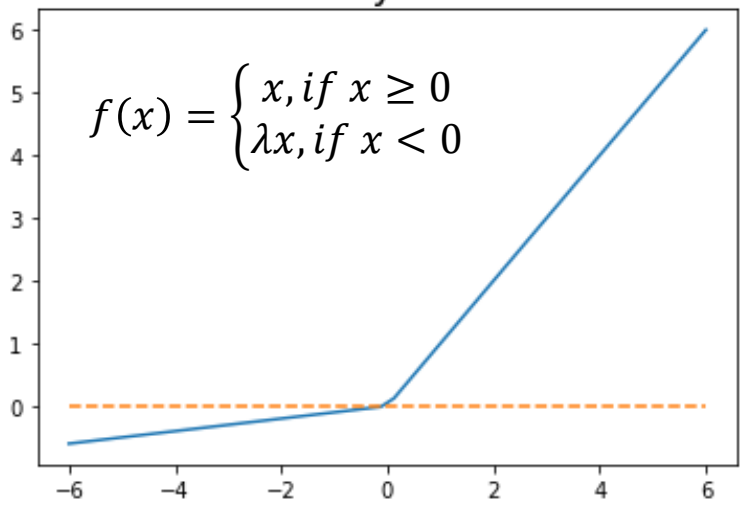
tanh



Relu



Leaky Relu

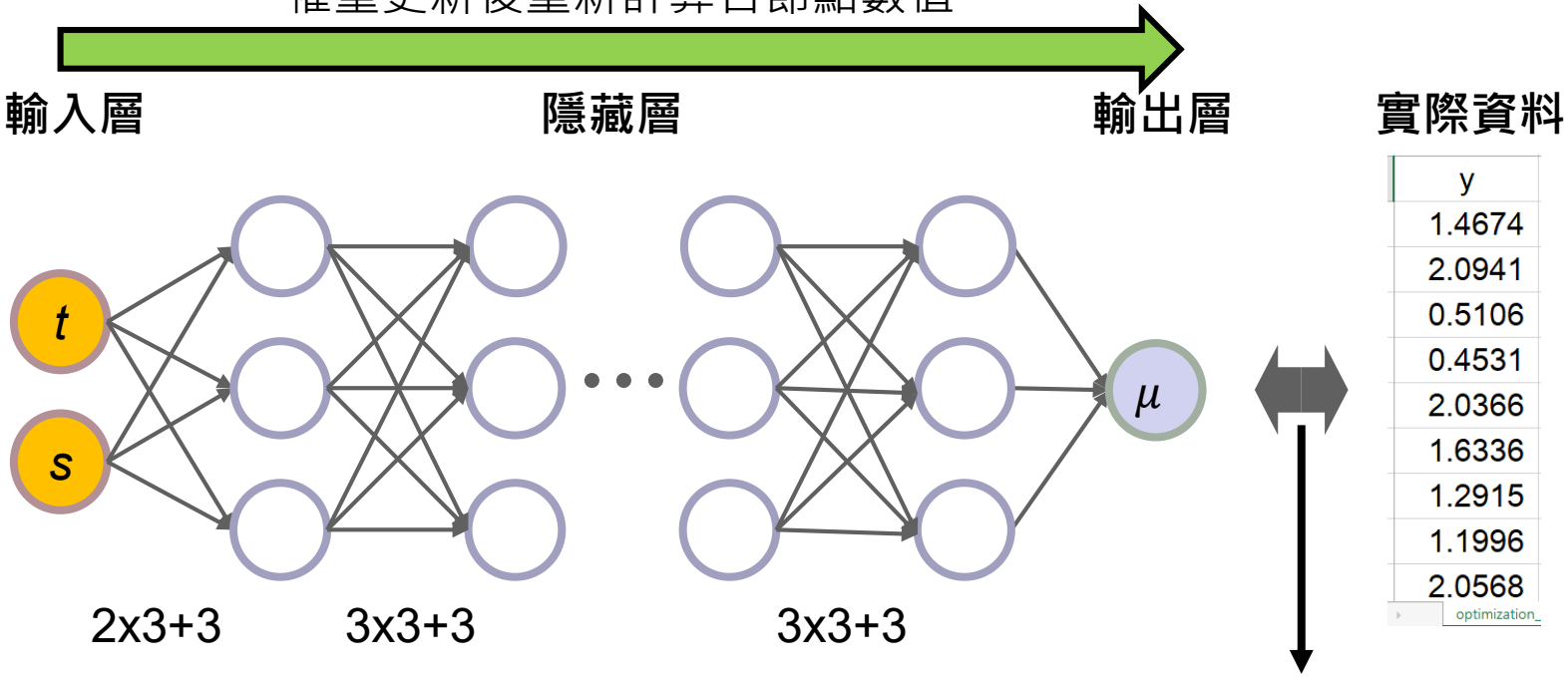


神經網路模型

權重　　偏差

- 參數最佳化量級：
  - 逐個隱藏層疊加 Σ前一層node數 x 當層 node 數 + 當層 node 數

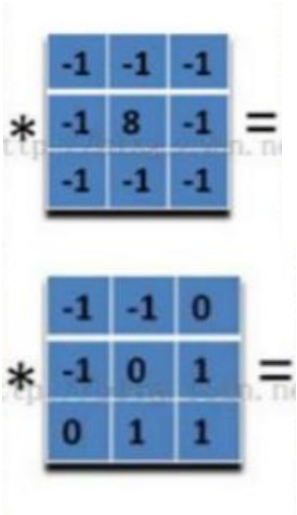
權重更新後重新計算各節點數值

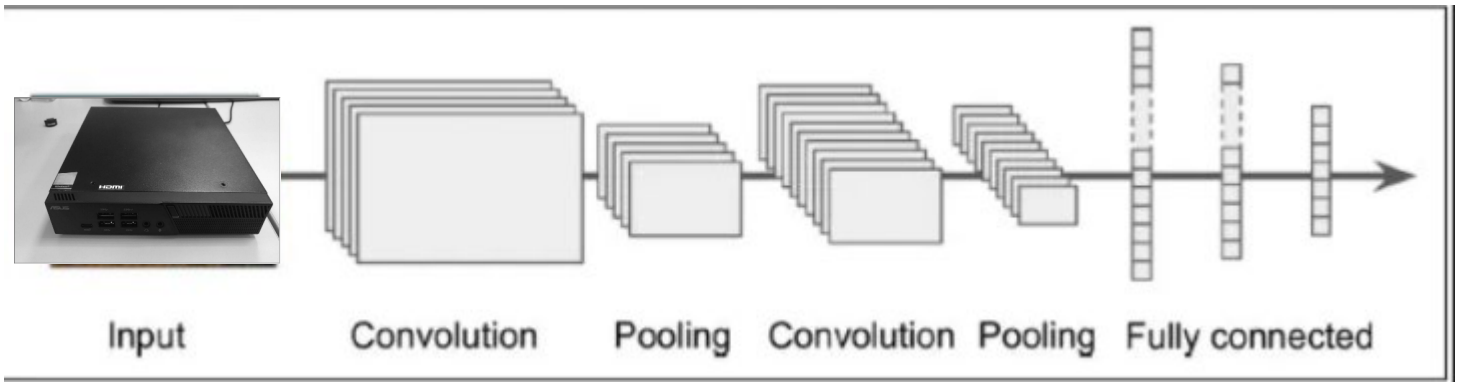


誤差　$\mathrm{MSE} = f(\boldsymbol{\beta}|y,t,s) = \sum_{i=1}^{n}[y_i - \mu(t_i,s_i|\boldsymbol{\beta})]^2$

使用倒傳遞 Back-propagation 演算法更新權重

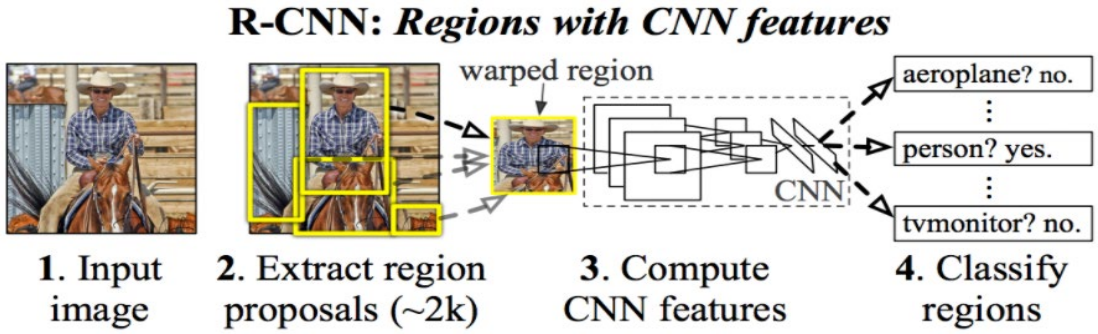
Convolutional Neural Networks
卷積神經網路





Object detection – 2 stage

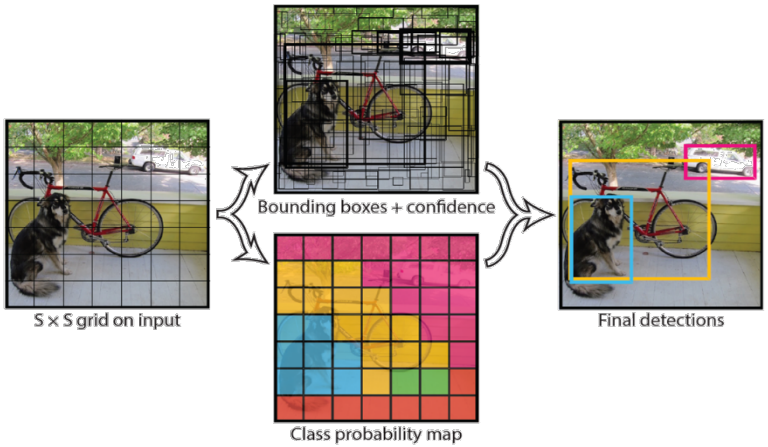
Sliding Windows，逐一掃瞄



R-CNN使用Selective Search找出2000－3000個region proposal，將取出的region proposal 壓縮成一樣大小之後再丟入CNN擷取特徵，利用SVM加以分類。

Object detection – YOLO

- Object Detection = Object Localization + Feature Extraction + Image Classification
- localization：框出bounding box（BB for short）



- YOLO 的概念是將一張圖片切割成 S x S 個方格，每個方格以自己為中心點各自去判斷 B 個 bounding boxes 中包含物體的 confidence score 跟種類。
- end-to-end 做物件偵測：基於影像的全域性資訊，利用CNN來同時預測多個**BB**並且針對每一個box來計算物體的機率，避免傳統object detection的必須分開訓練的缺點，並且大幅加快運算速度。

Object detection - YOLO

- 使用Keras進行yolo的預測
  - https://github.com/qqwweee/keras-yolo3
  - 下載source code
  - 下載yolo的原始weights檔案
  - 轉換為h5檔案 (kerase格式)

```
wget https://pjreddie.com/media/files/yolov3.weights
python convert.py yolov3.cfg yolov3.weights model_data/yolo.h5
```

```
_defaults = {
    #"model_path": 'model_data/trained_weights_stage_1.h5',
    "model_path": 'model_data/yolo.h5',

    "anchors_path": 'model_data/yolo_anchors.txt',
    "classes_path": 'model_data/coco_classes.txt',
    "score" : 0.3,
    "iou" : 0.45,
    "model_image_size" : (416, 416),
    "gpu_num" : 1,
}
```

影像偵測的範例

```
image_path = "D:\\YJ_AI_Test\\yolo_keras\\keras-yolo3-master\\human01.jpg"
image = Image.open(image_path)
yolo = YOLO()
r_image = yolo.detect_image(image=image)
r_image.show()
```

使用照片進行物件偵測

```
'''
yolo1 = YOLO()
sss=r'D:\YJ_AI_Test\yolo_keras\keras-yolo3-master\dog.mp4'
detect_video(yolo=yolo1,video_path=sss)
'''
```

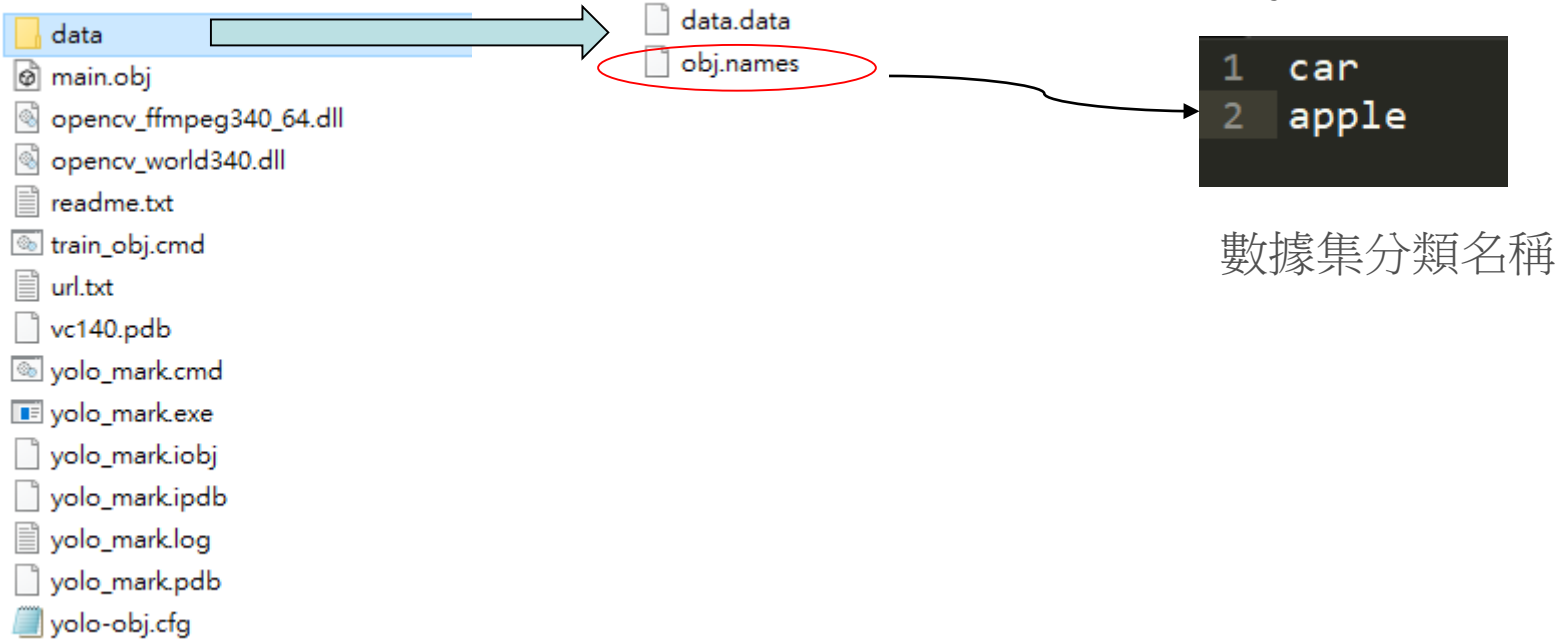
使用影音檔進行物件偵測

```
python yolo.py
```

在虛擬環境下啟動

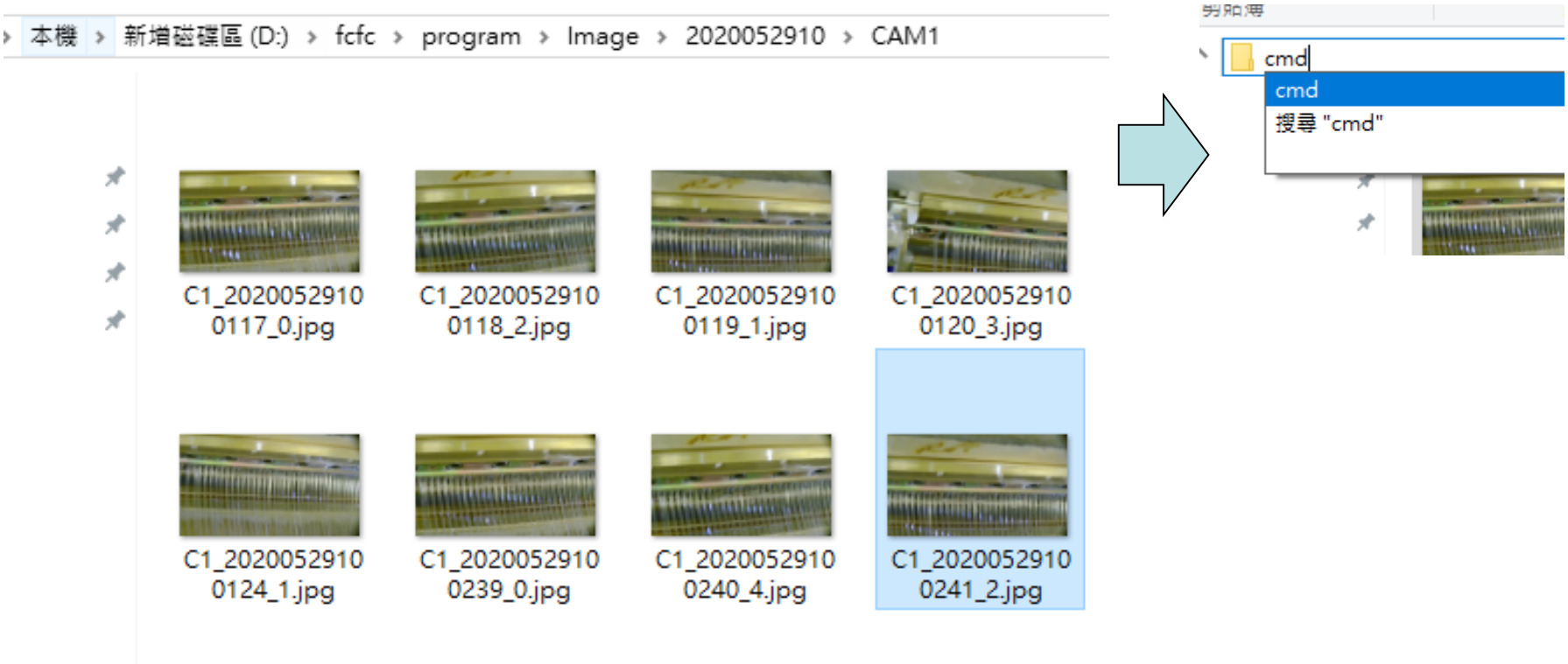
標記工具-YOLO Mark

- YOLO Mark是一種專為YOLO準備數據訓練的工具。此工具需依賴open cv。
- 資料夾內附幾個需要修改的檔案，yolo_mark.cmd為編寫相關資料的工作提示檔案，data資料夾中的obj.names是數據集所有分類的名稱。
- 以下介紹使用YOLO Mark的步驟。



YOLO Mark Step1

- Step1. 開始標記前，要先取得所有影像的路徑。因此先打開要標記的影像資料夾，在資料夾路徑打上cmd。

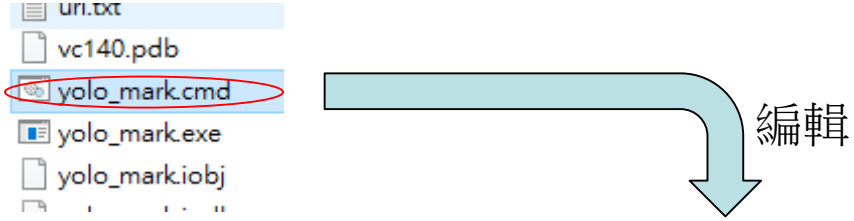


YOLO Mark Step2

- Step2. 當打完命令cmd後會出現命令列視窗，路徑就是目標資料夾的路徑。接續輸入以下命令 “ dir *.jpg /b/s > train.txt”。在該資料夾會出現一個新的train.txt檔案，此檔案內容即為資料夾內所有jpg檔案的列表。

YOLO Mark Step3

- Step3. 回到yolo mark資料夾，編輯yolo_mark.cmd的文件(或是新增一個.cmd的文件)，在新增如下格式內容：
  - “【yolo_mark.exe】【影像資料夾路徑】【影像列表檔案】【data/obj.names】 ”
  - 【yolo_mark.exe】 ：yolo mark 的執行檔，此範例為”yolo_mark.exe” 。
  - 【影像資料夾路徑】：要進行標記的影像資料夾路徑，如step2內所提及的資料夾，此範例為” D:\fcfc\program\Image\2020052910\CAM1” 。
  - 【影像列表檔案】 ：要進行標記的影像的檔案列表，就如step2內所提及的train.txt ，此範例為”D:\fcfc\program\fcfc_image_0624_c1_c4_train\1\train.txt”
  - 【data/obj.names】：data/obj.names是數據集所有分類的名稱 ，此範例為”data/obj.names”

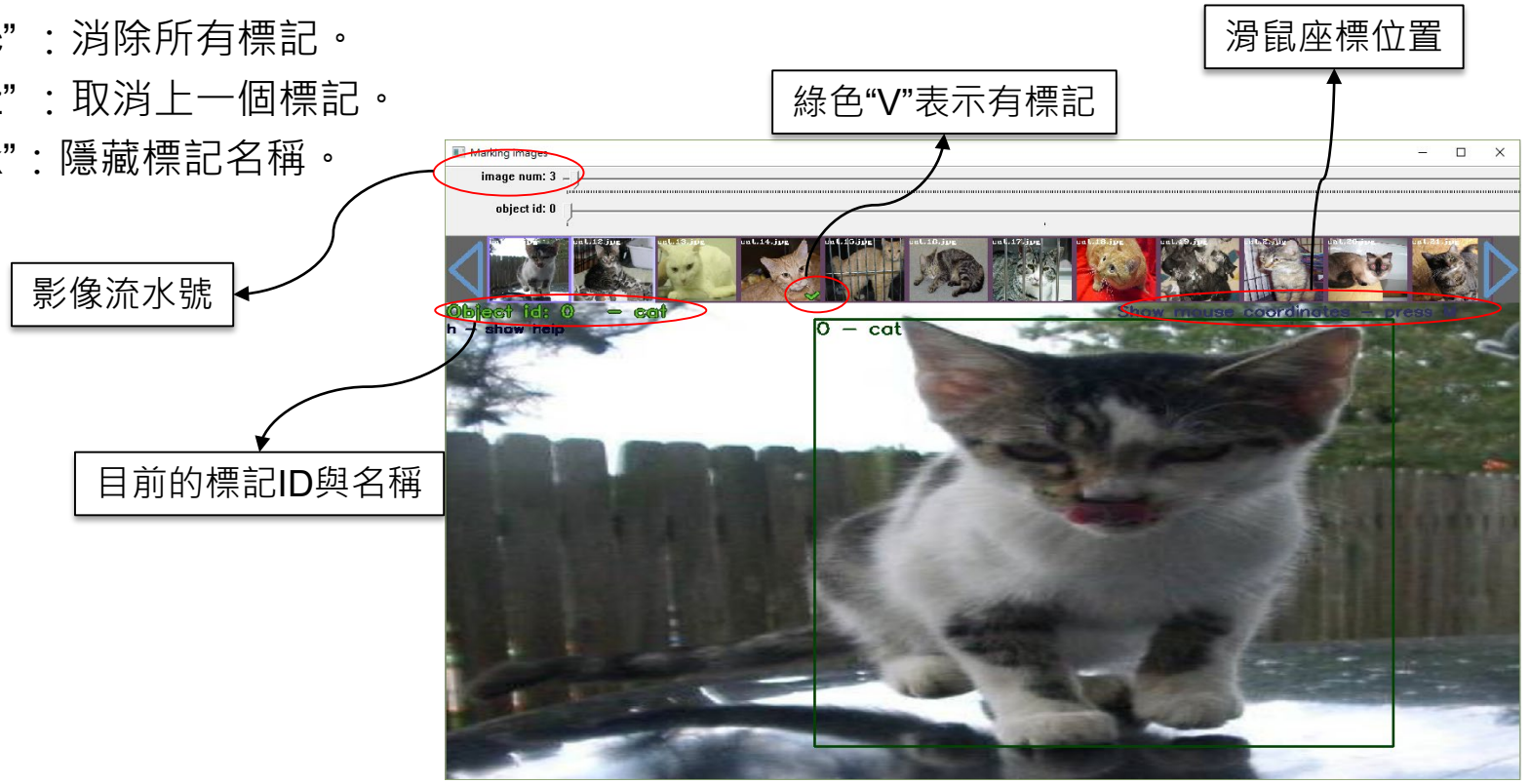



```
echo      Example how to start marking bouded boxes for training set Yolo v``

yolo_mark.exe D:\fcfc\program\Image\2020052910\CAM1 D:\fcfc\program\Image\2020052910\CAM1\train.txt data/obj.names
```

YOLO Mark Step4

- 點擊yolo_mark.cmd，用滑鼠直接進行拖拉放的動作，即可標示。按下ESC即可離開yolo_mark。 以下介紹幾個常用的功能(須注意大小寫)：
  - 方向鍵”←”與”→”：分別是上一張圖片，下一張圖片。
  - 數字鍵“0”~”9”：切換標記ID 。(需在obj.names 內編寫分類名稱)
  - “h”：可以顯示功能提示。
  - “M”：可以滑鼠座標位置。
  - “c” ：消除所有標記。
  - “z” ：取消上一個標記。
  - “k”：隱藏標記名稱。



YOLO Mark Step5

- 標記完成後，回到影像資料夾中，YOLO Mark會對每一張影像產生一個與影像相同檔名的txt檔案。此即為該影像的YOLO標記檔案。txt檔案中每一行代表一個標記。意義如下：

- 【標記ID】【標記框的中心點位置X】【標記框的中心點位置Y】【標記框的寬度w】【標記框的長度h】
- 注意：此格式中位置或大小都以影像的長或寬百分比做為紀錄方式。

YOLO training

```
set path=%path%;C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\bin
set LASTEST_MDL=PCBweightsTmp\yolo-model_50000.weights
set BIN=darknet_no_gpu.exe
set CFG=cfg/yolo_190226.cfg
rem set INIT_WEIGHTS=weights/yolov2.weights
set INIT_WEIGHTS=weights/darknet53.conv.74

rem Kfold 0

set DATA=obj\obj.data

:: train cmd
%BIN% detector train %DATA% %CFG% %INIT_WEIGHTS%
```

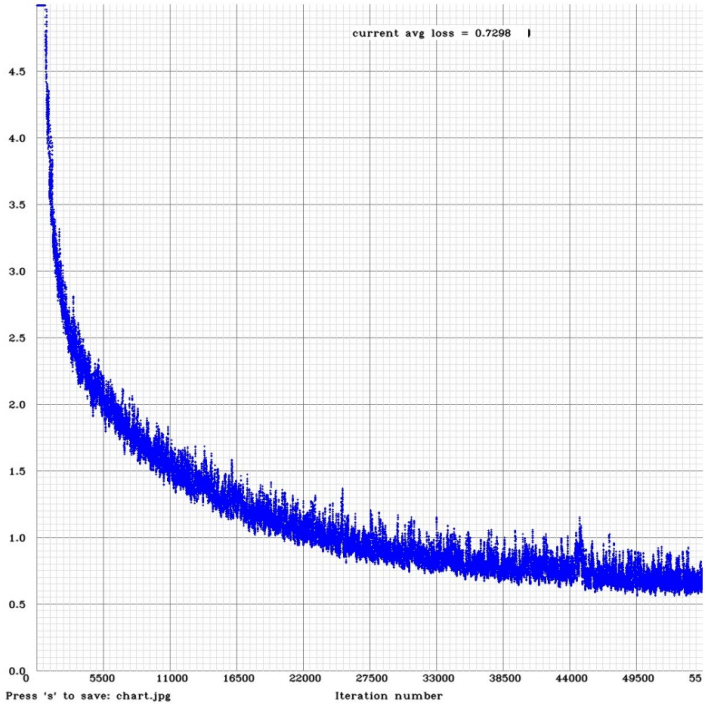
obj.data

```
classes= 1
train  = obj\train.txt
valid  = obj\valid.txt
names = ../yolo/obj/obj.names
backup = ../yolo/backup
```

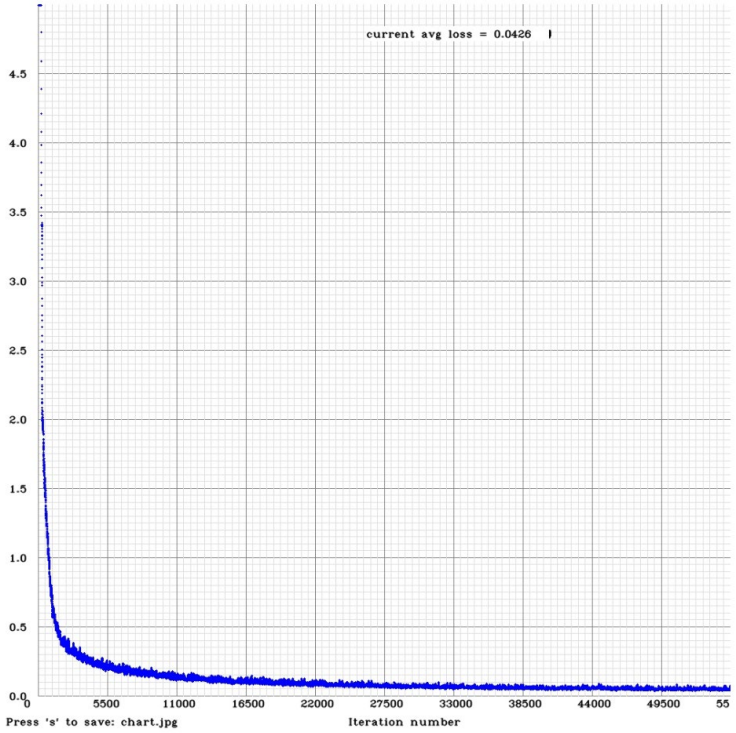
必要的設定檔案

1. 初始Weight
2. Configure
3. 訓練集
4. 執行程式
5. 驗證集

Model - 收斂曲線



較差的模型收斂



較佳的模型收斂

參考網站

- https://github.com/AlexeyAB/darknet
- https://pjreddie.com/darknet/yolo/