

實驗1：Python開發環境安裝

吳庭育

tyw429@gmail.com

Outline

- Anaconda介紹
- 安裝Anaconda套件組
- Anaconda Prompt模組管理
- 安裝TensorFlow和Keras
- Spyder 整合開發環境介紹
- Jupyter Notebook介紹
- Google Colaboratory介紹

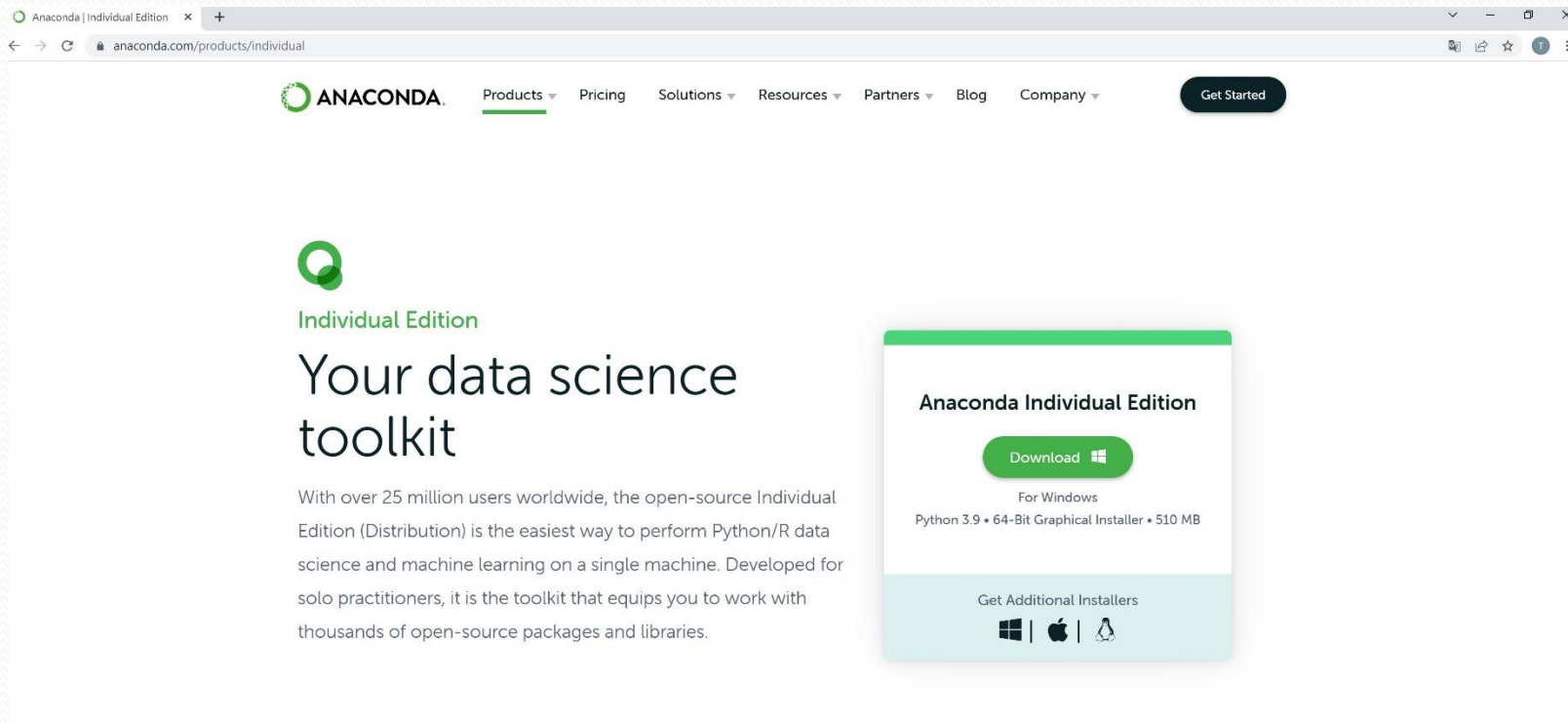
安裝 Anaconda 套件組

Anaconda介紹

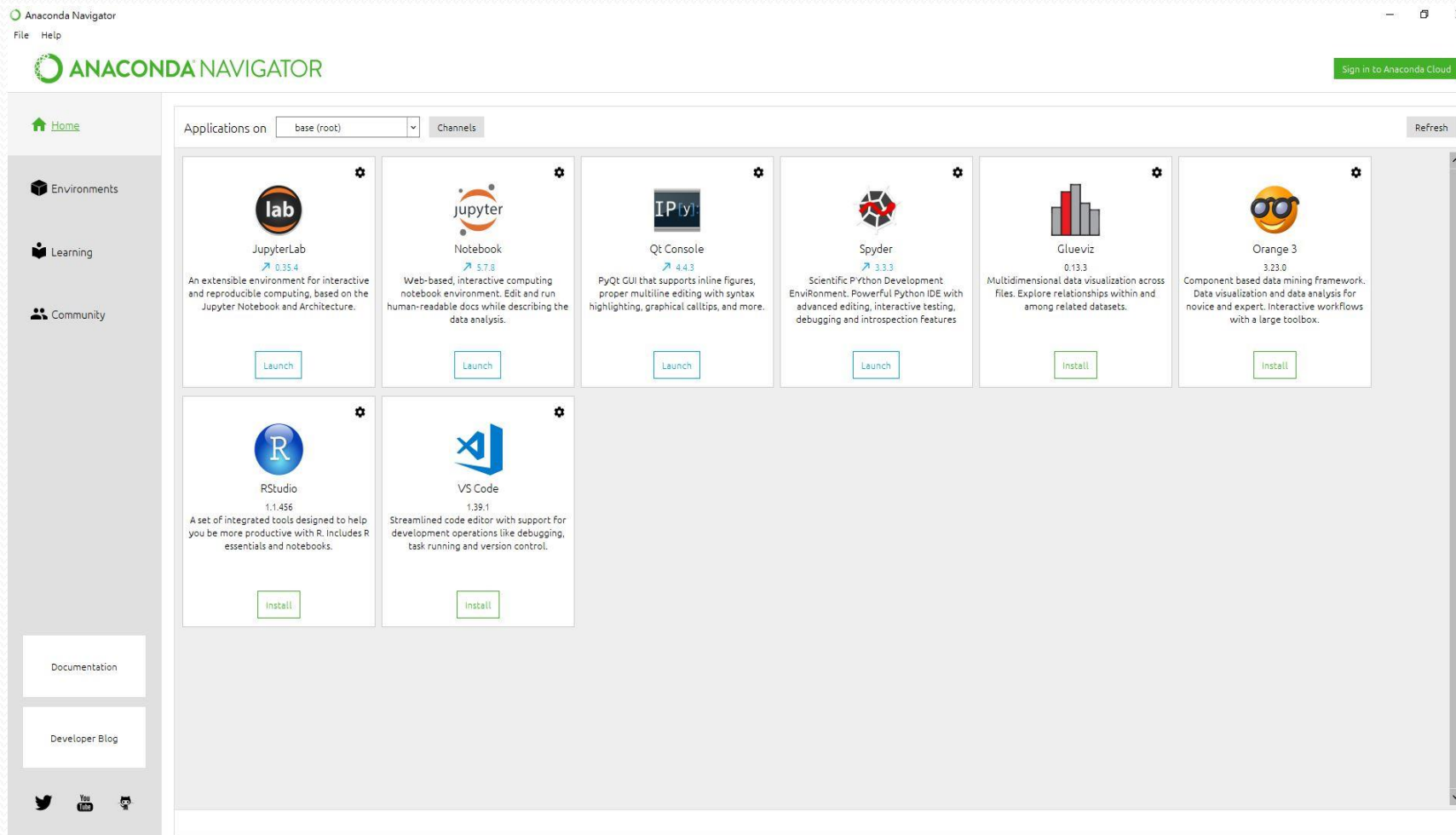
- Anaconda 是Python語言著名的整合安裝套件，內建 **Spyder** 整合開發環境和 **Jupyter Notebook**，除了標準模組外,還包含Scipy、NumPy、Pandas和 Matplotlib 等資料科學和機器學習的相關套件。

下載與安裝 Anaconda

<https://www.anaconda.com/distribution/>



Anaconda操作畫面

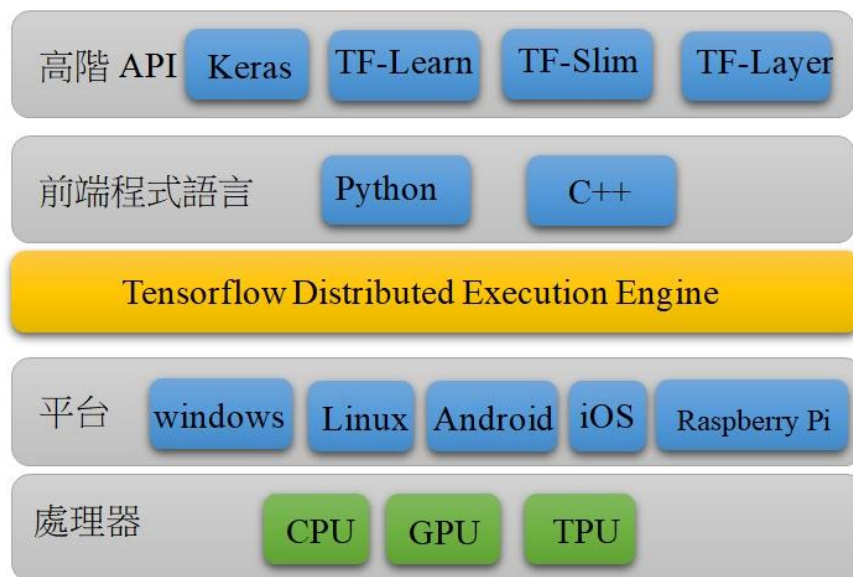


安裝 TensorFlow 和 Keras

認識 TensorFlow 與 Keras

- 目前支援 Python 語言的深度學習函式庫有很多種,最著名的就是Google的TensorFlow。而Keras 是架構在TensorFlow上的高階函式庫,可以更容易使用Python語言來實作深度學習的神經網路。

Tensorflow 架構圖說明:



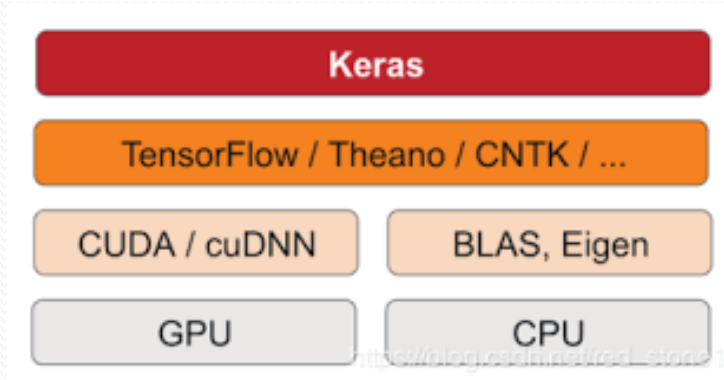
Google 的 TensorFlow

- 安裝 TensorFlow 和 Keras

- TensorFlow 是一套開放原始碼和高效能的數值計算函式庫，一個建立機器學習的框架。
- TensorFlow 是機器學習完整的學習平台，提供大量工具和社群資源，可以幫助開發者加速機器學習的研究與開發，和輕鬆部署機器學習的大型應用程式。
- TensorFlow 是 Google Brain Team 開發，在 2005 年底開放專案後，2017 年推出第一個正式版本。
- TensorFlow 因為其輸入/輸出的運算資料是向量、矩陣等多維度的數值資料稱為張量(Tensor)。
- 建立的機器學習模型需要使用流程圖來描述訓練過程的所有數值運算操作，稱為計算圖(Computational Graphs)
- Tensor 張量就是經過這些流程 Flow 的數值運算來產生輸出結果，稱為:Tensor + Flow=TensorFlow

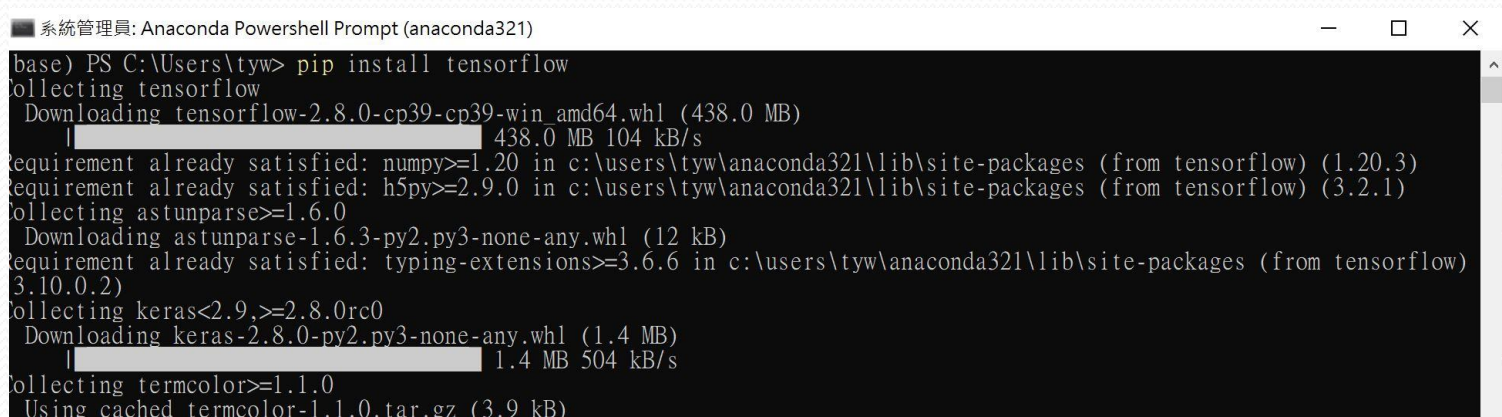
Keras 介紹

- Keras是Google工程師Francois Chollet 使用Python開發的一套開放原始碼的高階神經網路函式庫，支援多種後台(Backend)的神經網路計算引擎，其預設引擎是TensorFlow。
- Keras的特色如下：
 - Keras能夠使用相同的Python程式碼在CPU或GPU上執行。
 - Keras提供高階APIs來快速建構深度學習模型的神經網路。
 - Keras預建全連接、卷積、池化、RNN、LSTM和GRU等多種神經層。



TensorFlow和Keras套件安裝

- 完成 Anaconda 整合安裝套件後，可以開始安裝 TensorFlow 和 Keras 套件。
- 請執行「開始 /Anaconda3 (64-bits)/Anaconda Prompt」 令開啟「Anaconda Prompt」命令提示字元視窗後。
- 使用 pip install 指令安裝 TensorFlow 套件。
 - (base) C:\Users\JOE>pip install tensorflow <Enter>



```
系統管理員: Anaconda Powershell Prompt (anaconda321)
(base) PS C:\Users\tyw> pip install tensorflow
Collecting tensorflow
  Downloading tensorflow-2.8.0-cp39-cp39-win_amd64.whl (438.0 MB)
    |████████████████████████████████████████| 438.0 MB 104 kB/s
Requirement already satisfied: numpy>=1.20 in c:\users\tyw\anaconda321\lib\site-packages (from tensorflow) (1.20.3)
Requirement already satisfied: h5py>=2.9.0 in c:\users\tyw\anaconda321\lib\site-packages (from tensorflow) (3.2.1)
Collecting astunparse>=1.6.0
  Downloading astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\tyw\anaconda321\lib\site-packages (from tensorflow) (3.10.0.2)
Collecting keras<2.9,>=2.8.0rc0
  Downloading keras-2.8.0-py2.py3-none-any.whl (1.4 MB)
    |████████████████████████████████████████| 1.4 MB 504 kB/s
Collecting termcolor>=1.1.0
  Using cached termcolor-1.1.0.tar.gz (3.9 kB)
```

TensorFlow和Keras套件安裝

- 使用 pip install 指令安裝 TensorFlow 套件。
 - (base) C:\Users\JOE>pip install keras <Enter>
 - 如果出現安裝套件錯誤 為Numpy的版本不相容，使用下列指令更新版本
 - (base) C:\Users\JOE>pip install --upgrade numpy<Enter>

pip 指令介紹

- pip是一個以Python電腦程式語言寫成的軟體包管理系統，他可以安裝和管理軟體包，另外不少的軟體包也可以在「Python軟體包索引」
- pip的其中一個主要特點就是其方便使用的命令列介面，這讓使用者可以透過以下的一句文字命令來輕易地安裝Python軟體包：
 - `pip install some-package-name`
- 使用者也可以輕易地透過以下的命令來移除軟體包：
 - `pip uninstall some-package-name`
- pip也擁有一個透過「需求」檔案來管理軟體包和其相應版本數目的完整列表之功能，這容許一個完整軟體包組合可以在另一個環境（如另一部電腦）或虛擬化環境中進行有效率的重新創造。這個功能可以透過一個已正確進行格式化的文字檔案和以下的命令來完成：
 - `pip install -r requirements.txt`

Spyder 整合開發環境介紹

Spyder 整合開發環境介紹

- Spyder 是一套開放原始碼且跨平台的 Python 整合開發環境 (Integrated Development Environment, IDE)，這是功能強大的互動開發環境，支援程式碼編輯、互動測試、偵錯和執行 Python 程式。
- 啟動 Spyder 可以從 Anaconda Navigator 啟動 Spyder，也可以直接從開始功能表來啟動 Spyder。
 1. 請執行 <開始/Anaconda3 (64-bit)/Spyder> 命令。
 2. 如果初次使用 Python，就會顯示警告視窗。
 3. 請按允許存取鈕繼續。
 4. 訊息指出如果使用 Anaconda 內建的 Spyder，請不要自行升級。

Spyder開發環境

The screenshot displays the Spyder Python IDE (version 3.7) interface. The main editor window shows a Python script named `Ch4_2_1.py` containing a `Perceptron` class and its usage. The class implements an `__init__` method for initializing weights and bias, a static `activation_function` method, and a `__call__` method for processing input data. The script also creates instances of the `Perceptron` class and processes input data to produce output weights.

```
1 import numpy as np
2
3 class Perceptron:
4     def __init__(self, input_length, weights=None, bias=None):
5         if weights is None:
6             self.weights = np.ones(input_length) * 1
7         else:
8             self.weights = weights
9         if bias is None:
10            self.bias = -1
11        else:
12            self.bias = bias
13
14        @staticmethod
15        def activation_function(x):
16            if x > 0:
17                return 1
18            return 0
19
20        def __call__(self, input_data):
21            w_input = self.weights * input_data
22            w_sum = w_input.sum() + self.bias
23            return Perceptron.activation_function(w_sum), w_sum
24
25 weights = np.array([1, 1])
26 bias = -0.5
27 h1 = Perceptron(2, weights, bias)
28
29 weights = np.array([1, 1])
30 bias = -1.5
31 h2 = Perceptron(2, weights, bias)
32
33 weights = np.array([1, -2])
34 bias = -0.5
35 o1 = Perceptron(2, weights, bias)
36
37 input_data = [np.array([0, 0]), np.array([0, 1]),
38               np.array([1, 0]), np.array([1, 1])]
39 for x in input_data:
40     out1, w1 = h1(np.array(x))
41     out2, w2 = h2(np.array(x))
```

The IPython console on the right shows the execution of the script, displaying the output of the `Perceptron` class for each input data point. The console output is as follows:

```
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.4.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/tyw/Desktop/tyw/Ch4_2_2.py', wdir='C:/Users/tyw/Desktop/tyw')
[0 0] [-0.5  1.5] [0 1] 0
[0 1] [0.5 0.5] [1 1] 1
[1 0] [0.5 0.5] [1 1] 1
[1 1] [ 1.5 -0.5] [1 0] 0

In [2]: runfile('C:/Users/tyw/Desktop/tyw/Ch4_4_2.py', wdir='C:/Users/tyw/Desktop/tyw')
1.42

In [3]: runfile('C:/Users/tyw/Desktop/tyw/Ch4_2_1.py', wdir='C:/Users/tyw/Desktop/tyw')
[0 0] [-0.5 -1.5] [0 0] 0
[0 1] [ 0.5 -0.5] [1 0] 1
[1 0] [ 0.5 -0.5] [1 0] 1
[1 1] [ 1.5 0.5] [1 1] 0

In [4]:
```

The bottom status bar indicates the current file is "New file", permissions are "RW", end-of-lines are "CRLF", encoding is "ASCII", and the current line and column are 1 and 1, respectively. The memory usage is 32%.

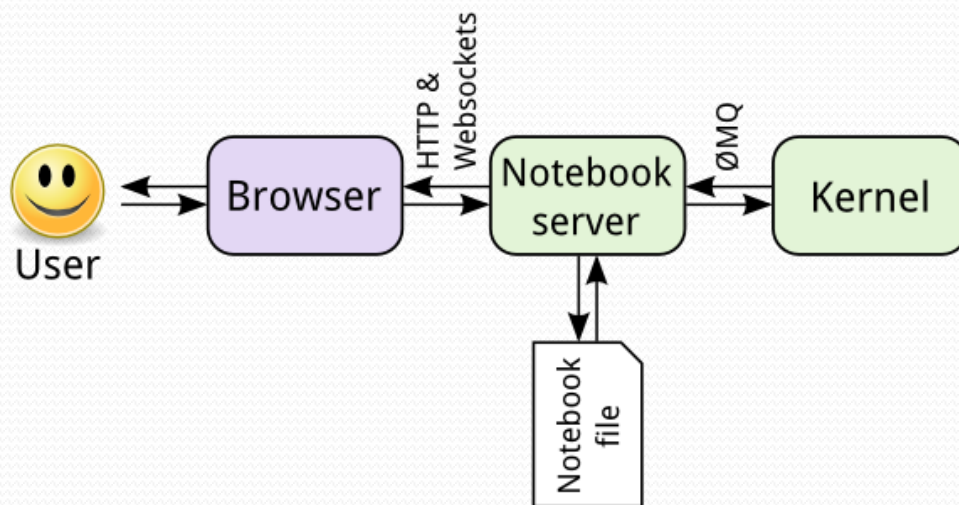
Jupyter Notebook介紹

Jupyter Notebook 介紹 (1/3)

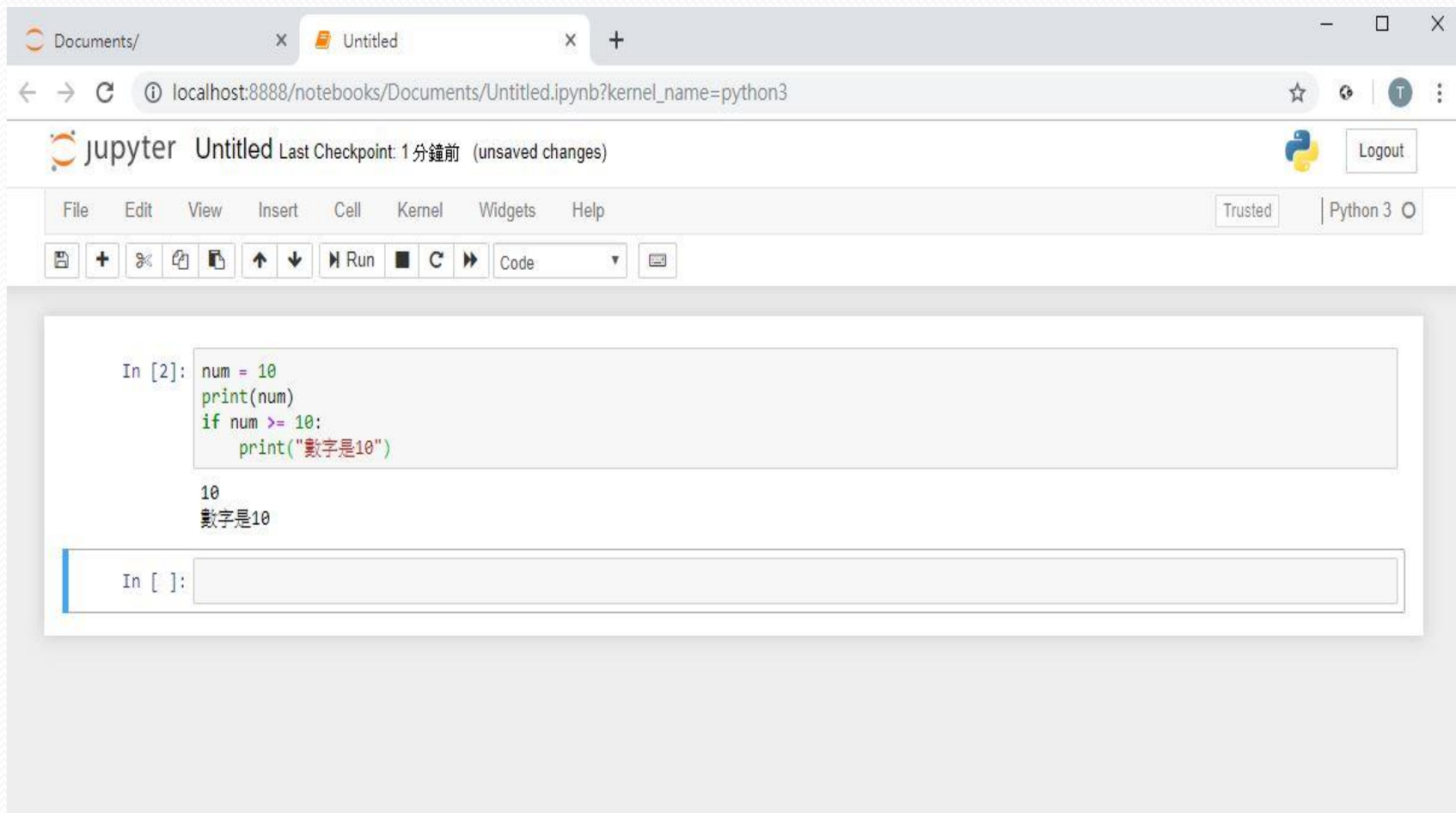
- Notebook是將console-based的方式，延伸到一個可互動式計算的新方向，並且提供一個web-based的方式來處理整個計算過程，包括開發、編輯、文件化及執行程式，並且可立即傳遞結果。
- IPython是一個提供互動式運算架構的系統，具有強大的交互式shell及數據運算視覺化的特點，可方便且彈性的整合不同編輯器，擁有高效能及容易使用的平行運算功能，同時也是Jupyter Notebook其中的一個kernel。
- Jupyter則是從IPython發展演變而來的，後來持續發展成多語言不再只支援Python。
- Jupyter這個名字主要是由Julia、Python及R所構成，當然除了Julia、的Python及R之外，還有提供許多的kernels(例如Java、C#、Go、Ruby、JavaScript等超過50多種Jupyter kernels)。

Jupyter Notebook介紹 (2/3)

- Jupyter Notebook 包含了兩個組成： Web Application 和 Notebook Documents。
 - 網頁應用(Web Application)：基於瀏覽器(web-based)的互動創作及應用工具，包括可以計算、數學、文檔創作及豐富的多媒體輸出。
 - 文檔顯示(Notebook Documents)：顯示所有在上述Web Application當中的內容，包括計算的輸入/輸出、文件說明/解釋、數學運算及式子、圖片及所有豐富多媒體內容。



Jupyter Notebook介紹 (3/3)



The screenshot displays a web browser window with a Jupyter Notebook interface. The browser's address bar shows the URL `localhost:8888/notebooks/Documents/Untitled.ipynb?kernel_name=python3`. The notebook's title bar indicates the file is named "Untitled" and shows the last checkpoint was saved 1 minute ago with unsaved changes. A "Logout" button is visible in the top right corner. The main menu includes "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". The toolbar contains icons for saving, creating new cells, deleting cells, running code, and other standard Jupyter actions. The current cell is a code cell containing the following Python code:

```
In [2]: num = 10
        print(num)
        if num >= 10:
            print("數字是10")
```

The output of the code is displayed below the code cell:

```
10
數字是10
```

Below the output, there is an empty input prompt for the next cell:

```
In [ ]:
```

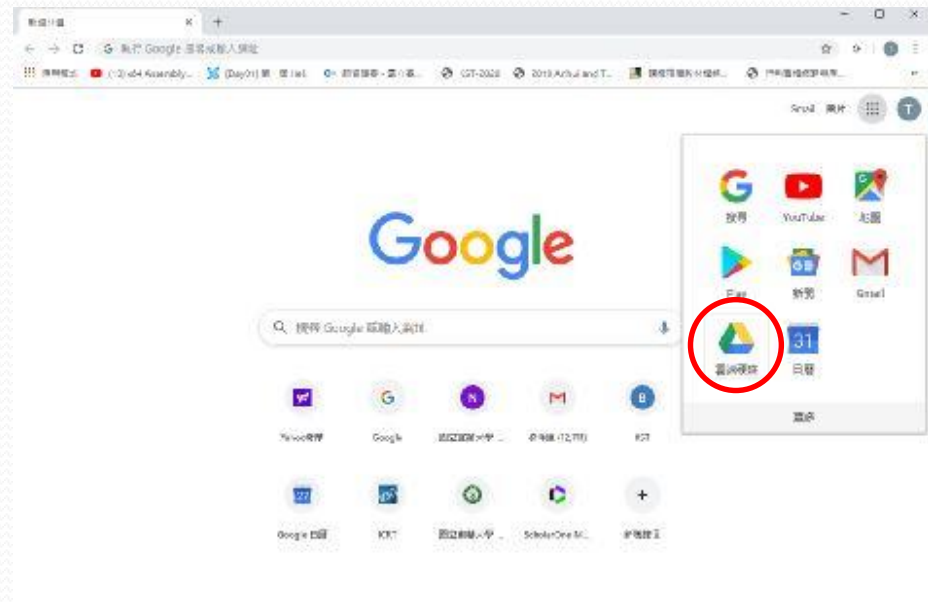
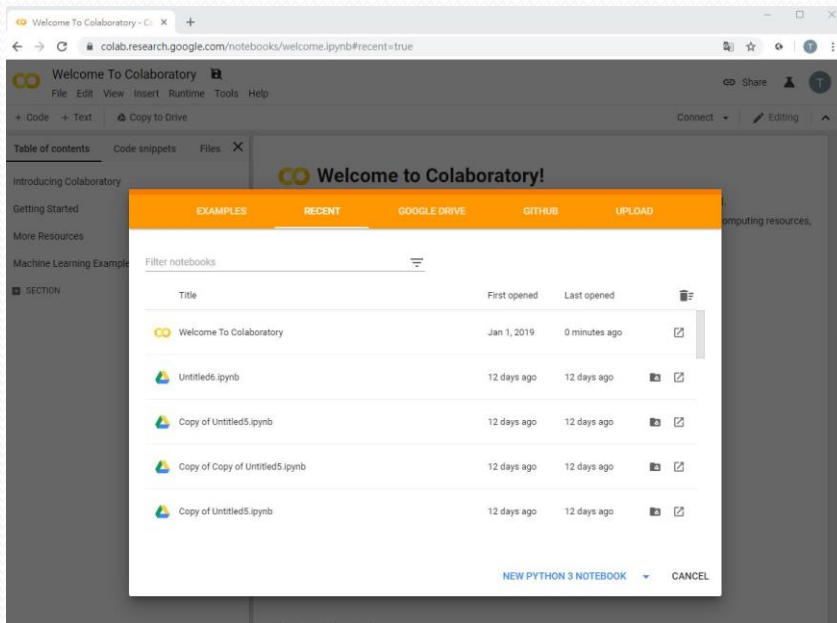
Google Colaboratory介紹

Google Colaboratory 介紹

- Google Colaboratory 簡稱 Google Colab，是一個基於 Jupyter Notebook 的免費服務(須註冊一個 Google 帳號)，所以如果曾經用過 Jupyter Notebook 來學習 Python 的話，那應該不必特別學習就能立即上手這個服務了。
- 使用 Google Colaboratory 不需要安裝 NumPy、Pandas、Matplotlib、TensorFlow 和 Keras 等資料科學和深度學習套件，就可以馬上使用 Colaboratory 來開發 Python 深度學習專案或應用，而且所有操作完全是在雲端進行。
- Colaboratory 提供強大計算能力且支援 GPU/TPU 加速運算，就算你沒有足夠效能的工作電腦，一樣可以使用 Colaboratory 雲端服務來入門 Python 深度學習，而且目前來說這是完全免費的服務。

Google Colaboratory 介紹

- 如果你已經有了 Google 帳號，打開瀏覽器到 <https://colab.research.google.com/> 就可以準備開始使用 Google Colab 了，一開始會要你建立一本新的 notebook 開始，從提示畫面中你也可以看到有多種方法能建立 notebook：



Untitled1.ipynb - Colaboratory

colab.research.google.com/drive/15ySEOVgIMN0E4M...

File Edit View Insert Runtime Tools Help

+ Code + Text

RAM Disk

Editing

```
import numpy as np

class Perceptron:
    def __init__(self, input_length, weights=None, bias=None):
        if weights is None:
            self.weights = np.ones(input_length) * 1
        else:
            self.weights = weights
        if bias is None:
            self.bias = -1
        else:
            self.bias = bias

    @staticmethod
    def activation_function(x):
        if x > 0:
            return 1
        return 0

    def __call__(self, input_data):
        w_input = self.weights * input_data
        w_sum = w_input.sum() + self.bias
        return Perceptron.activation_function(w_sum), w_sum

weights = np.array([1, 1])
bias = -0.5
h1 = Perceptron(2, weights, bias)

weights = np.array([1, 1])
bias = -1.5
h2 = Perceptron(2, weights, bias)

weights = np.array([1, -2])
bias = -0.5
o1 = Perceptron(2, weights, bias)

input_data = [np.array([0, 0]), np.array([0, 1]),
               np.array([1, 0]), np.array([1, 1])]
for x in input_data:
    out1, w1 = h1(np.array(x))
    out2, w2 = h2(np.array(x))
    new_point = np.array([w1, w2])
    new_input = np.array([out1, out2])
    out, w = o1(new_input)
    print(x, new_point, new_input, out)
```

```
[0 0] [-0.5 -1.5] [0 0] 0
[0 1] [ 0.5 -0.5] [1 0] 1
[1 0] [ 0.5 -0.5] [1 0] 1
[1 1] [1.5 0.5] [1 1] 0
```

Untitled1.ipynb - Colaboratory

colab.research.google.com/drive/15ySEOVgIMN0E4M...

File Edit View Insert Runtime Tools Help

+ Code + Text

Runtime

- Run all Ctrl+F9
- Run before Ctrl+F8
- Run the focused cell Ctrl+Enter
- Run selection Ctrl+Shift+Enter
- Run after Ctrl+F10
- Interrupt execution Ctrl+M I
- Restart runtime... Ctrl+M .
- Restart and run all...
- Reset all runtimes...
- Change runtime type**
- Manage sessions
- View runtime logs

Notebook settings

Runtime type
Python 3

Hardware accelerator
GPU

☐ Omit code cell output

None
GPU
TPU

Cancel this notebook

CANCEL SAVE

```
data = [np.array([0, 0]), np.array([0, 1]),
```


動腦時間

- 請說明TensorFlow與Keras的關係為何？
- 試比較Spyder與Jupyter Notebook的差異？
- 如果個人的主機無支援GPU時，要使用深度學習是否使用Google Colaboratory可以解決運算量不足的問題？為什麼？

