

第九章

Keras CIFAR-10 影像辨識資料集介紹



CIFAR-10

CIFAR-10由Alex Krizhevsky、Vinod Nair 與 Geoffrey Hinton收集的一個用於影像辨識資料集，共有10個分類：飛機、汽車、鳥、貓、鹿、狗、青蛙、船、卡車。

CIFAR-10資料集與之前MNIST資料集相比，彩色、雜訊較多，同一分類如卡車：大小不一、角度不同、顏色不同。所以CIFAR-10影像辨識的難度，比起MNIST資料集高很多。

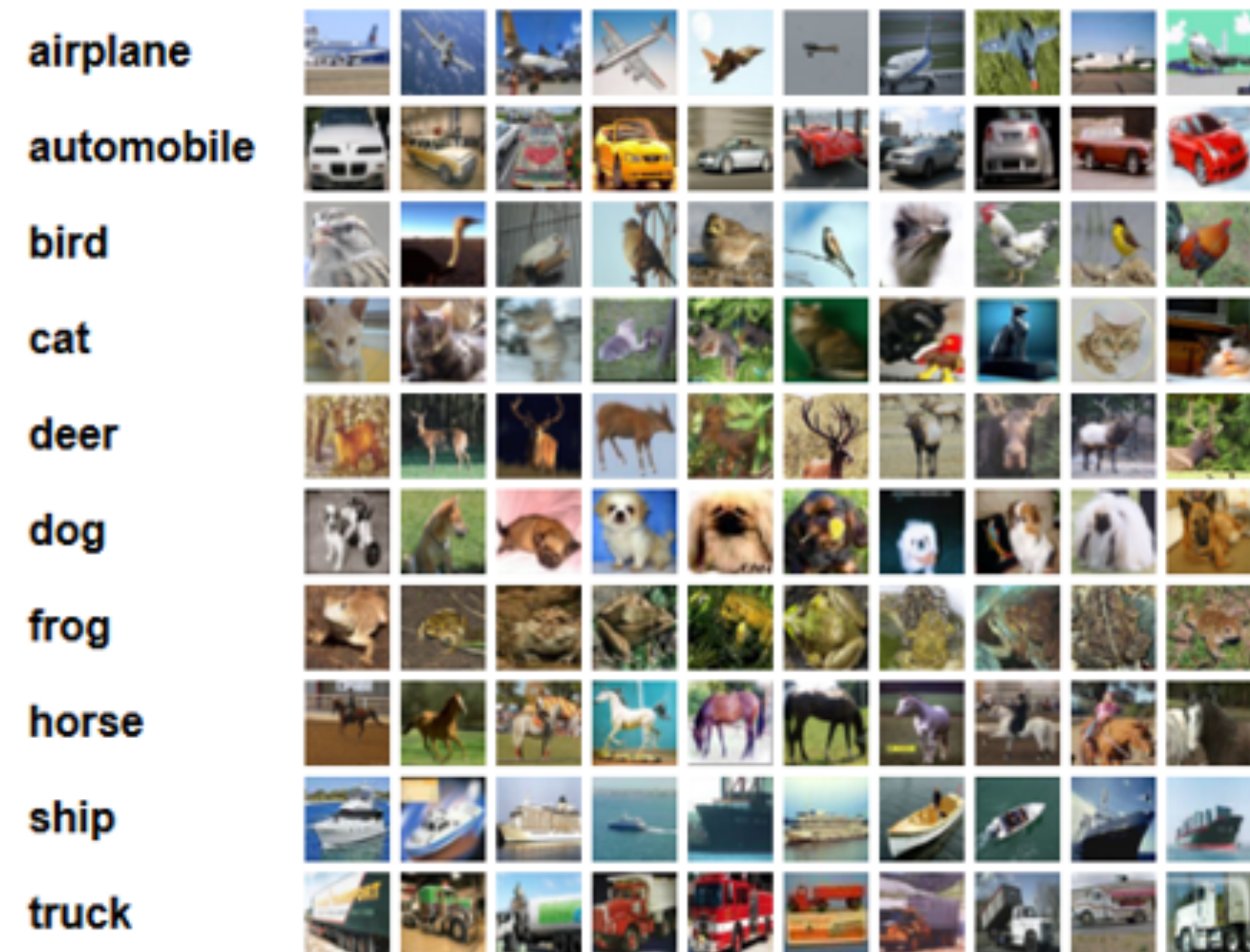
CIFAR-10

The CIFAR-10 dataset

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

Here are the classes in the dataset, as well as 10 random images from each:



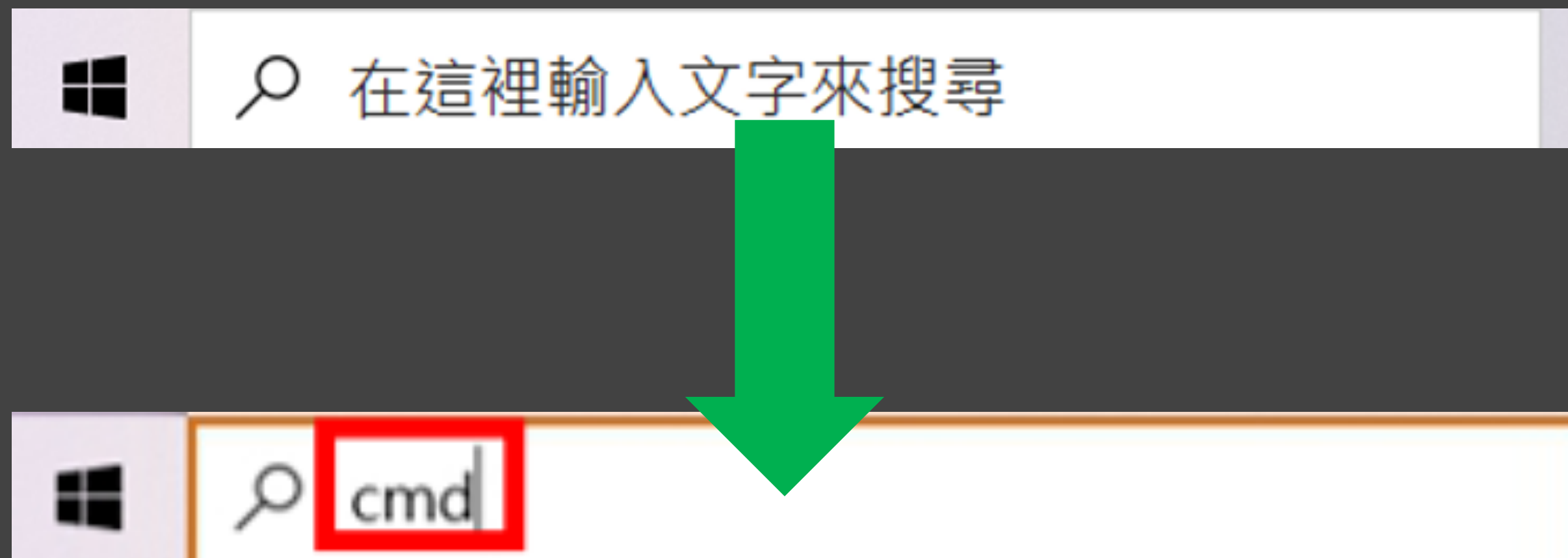
- CIFAR-10資料集共60000個32X32彩色圖像，有50000個訓練圖像和10000個測試圖像。
- 共有10個分類：飛機、汽車、鳥、貓、鹿、狗、青蛙、船、卡車。每個分類有6000個圖像。

9.1 下載CIFAR-10資料

9.1 下載CIFAR-10資料

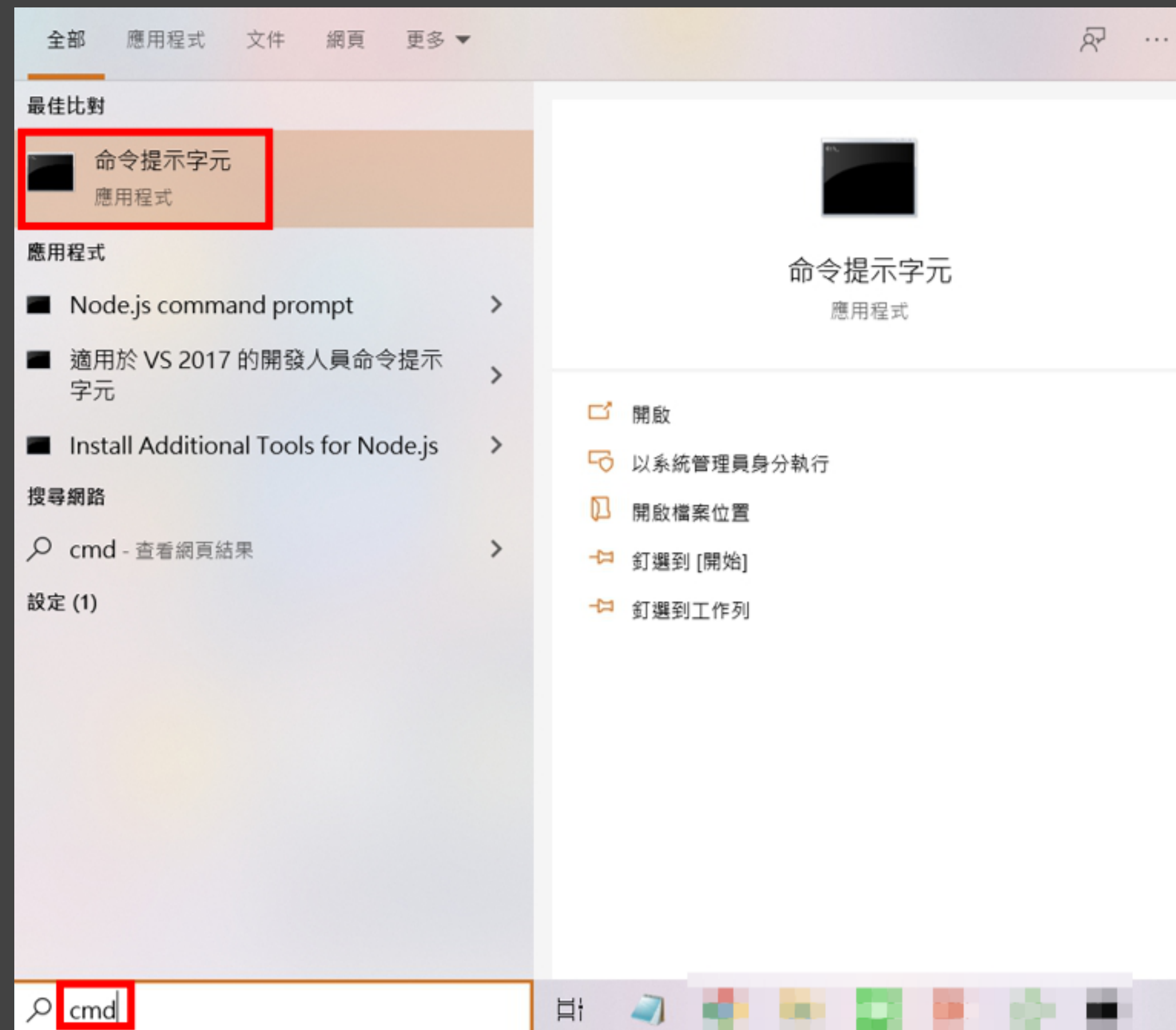
- 開啟命令提示字元

1. 左鍵單擊電腦工具列的搜尋欄位，並在當中輸入cmd。



9.1 下載CIFAR-10資料

2. 以左鍵單擊的方式開啟搜尋結果中的命令提示字元。



9.1 下載CIFAR-10資料

- 開啟Jupyter Notebook

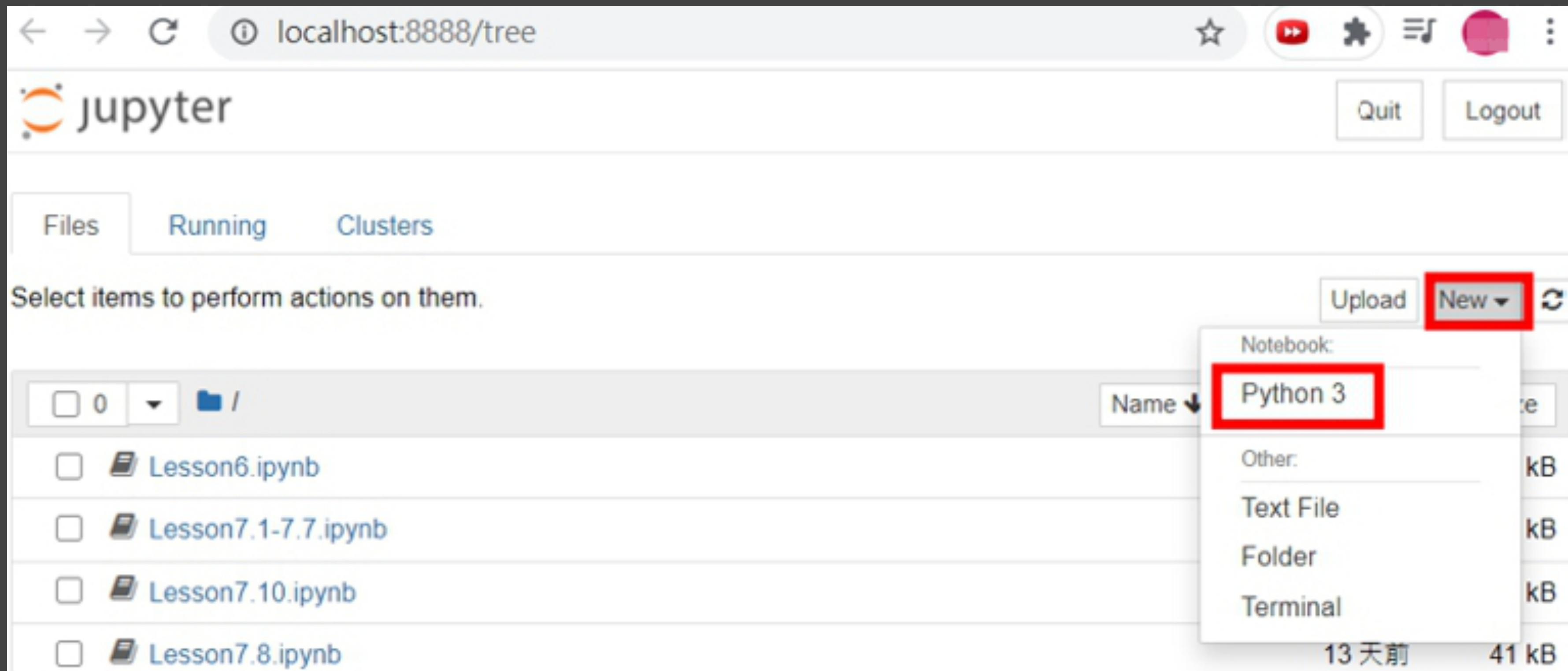
3. 在命令列輸入「jupyter notebook」接著按下Enter鍵。

```
C:\pythonwork>jupyter notebook
[I 18:43:38.686 NotebookApp] JupyterLab extension loaded from C:\Users\nan\anaconda3\lib\site-packages\jupyterlab
[I 18:43:38.686 NotebookApp] JupyterLab application directory is C:\Users\nan\anaconda3\share\jupyter\lab
[I 18:43:38.696 NotebookApp] Serving notebooks from local directory: C:\pythonwork
[I 18:43:38.697 NotebookApp] Jupyter Notebook 6.1.4 is running at:
[I 18:43:38.699 NotebookApp] http://localhost:8888/?token=cbc96ad8661f78ed3dc3073404088a3affce32ed4c84242e
[I 18:43:38.702 NotebookApp] or http://127.0.0.1:8888/?token=cbc96ad8661f78ed3dc3073404088a3affce32ed4c84242e
[I 18:43:38.702 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 18:43:38.911 NotebookApp]
```

9.1 下載CIFAR-10資料

- 創建新的Python 3檔案

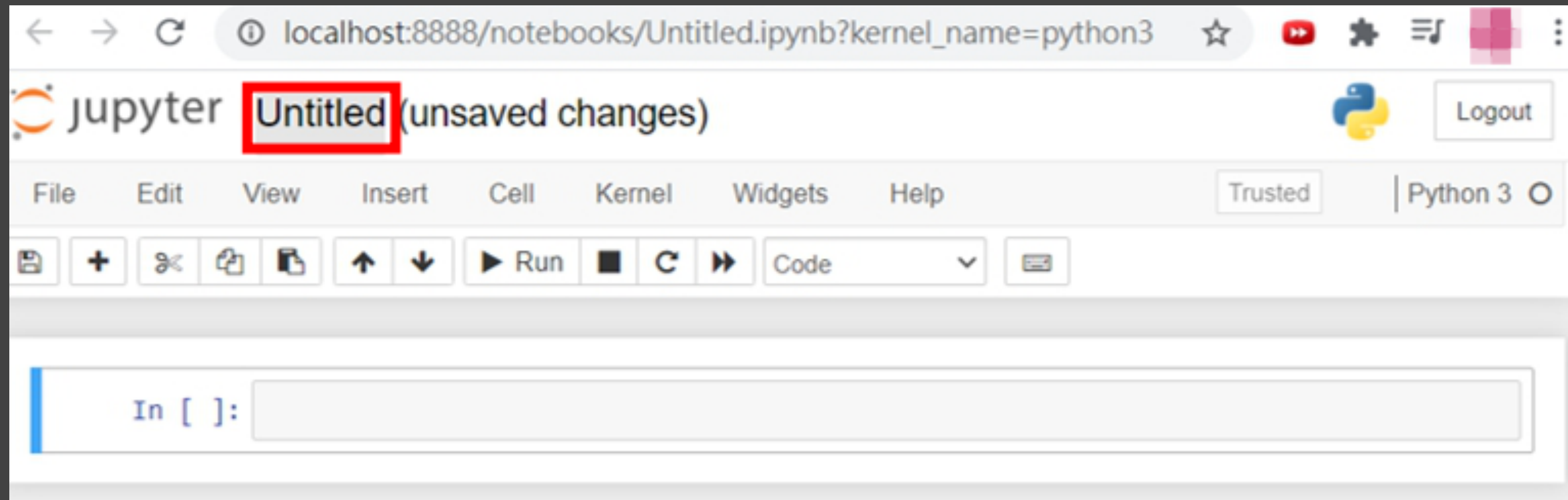
4. 左鍵單擊畫面右上方的New按鈕，並在出現的選單中左鍵單擊Python3。



9.1 下載CIFAR-10資料

- 更改新檔案的名稱

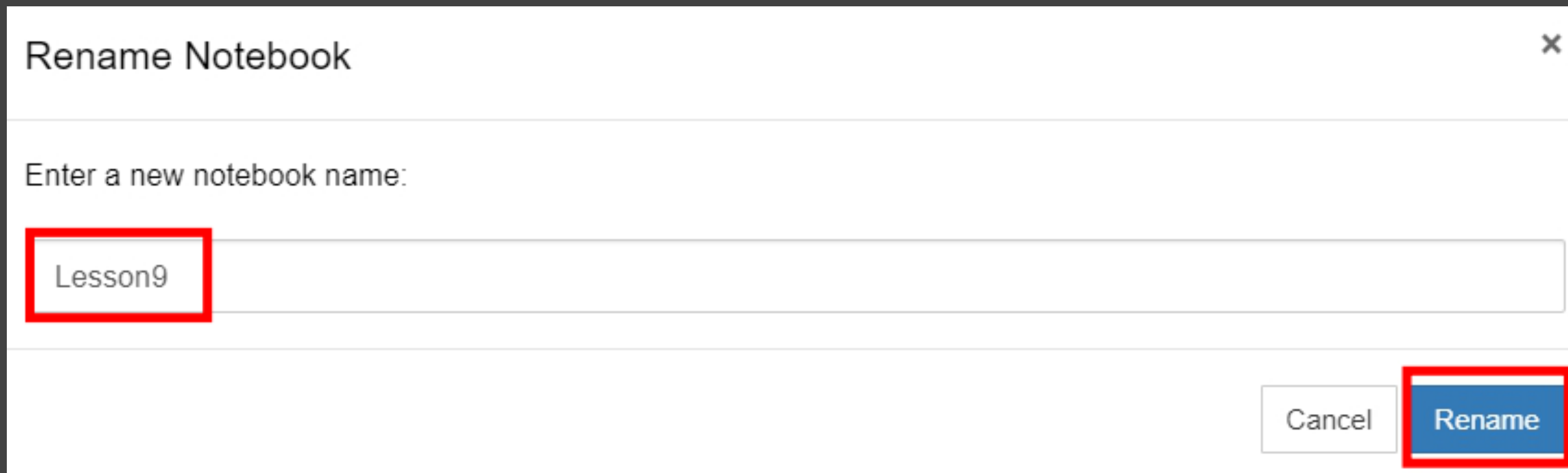
5. 左鍵單擊Untitled(圖中紅框處)。



9.1 下載CIFAR-10資料

- 開啟Jupyter Notebook

6. 在輸入框輸入Lesson 9，並且左鍵單擊Rename按鈕。



Rename Notebook

Enter a new notebook name:

Lesson9

Cancel Rename

9.1 下載CIFAR-10資料

- 匯入所需模組

7. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
from keras.datasets import cifar10
import numpy as np
np.random.seed(10)
```

從keras.datasets匯入cifar10資料集

匯入numpy模組，NumPy是Python語言的擴充程式庫。支援維度陣列與矩陣運算。

設定seed可以讓每次需要隨機產生的資料，都有相同的輸出。

9.1 下載CIFAR-10資料

- 下載並且解壓縮cifar10檔案

8. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

以下是第一次執行下載檔案的畫面。

```
(x_img_train,y_label_train),\  
(x_img_test,y_label_test)=cifar10.load_data()
```

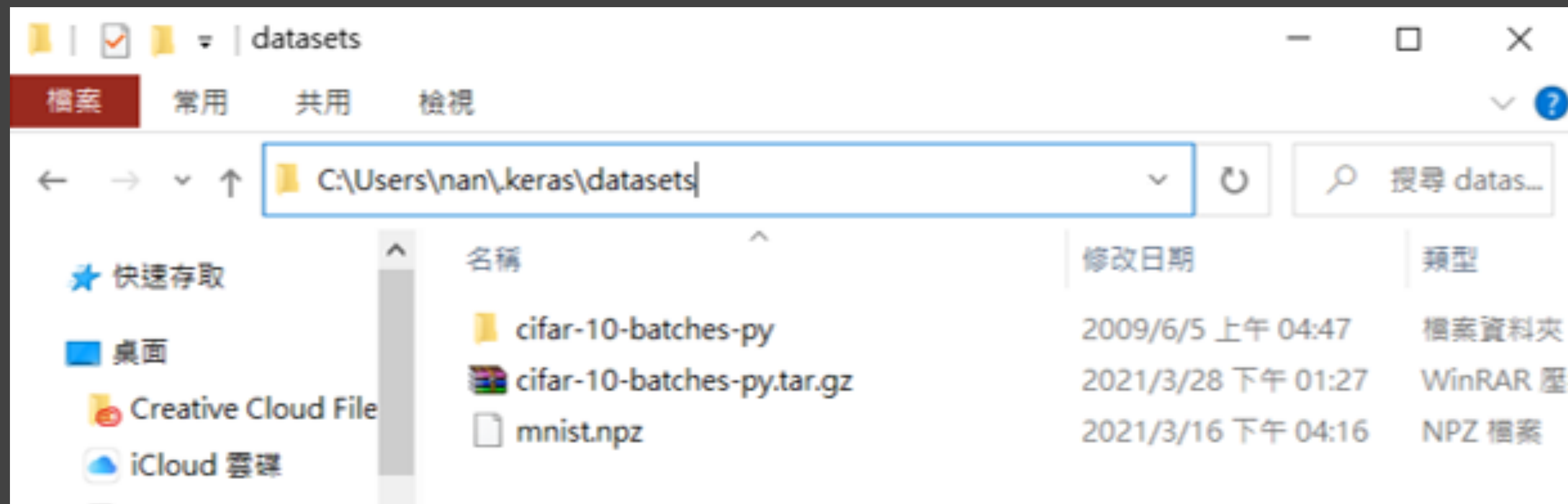
```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz  
170500096/170498071 [=====] - 224s 1us/step
```

Keras提供cifar10.load_data可以下載或讀取cifar10資料第一次執行cifar10.load_data()方法時,程式會檢查是否有cifar-10-batches-py.tar檔案,如果還沒有就會下載檔案,並且解壓縮檔案。因為必須要下載檔案所以執行時間會比較久。

9.1 下載CIFAR-10資料

- 查看cifar10資料檔案

9. 在下列目錄中可以找到cifar10資料檔案。



在Windows作業系統下查看cifar10資料檔案，因為筆者的使用者名稱是nan,所以下載後會存在下列目錄
C:\Users\nan\keras\datasets，檔案名稱是cifar-10-batches-py.tar

9.1 下載CIFAR-10資料

- 讀取cifar10資料

10. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
(x_img_train,y_label_train),\  
(x_img_test,y_label_test)=cifar10.load_data()
```

當您下次再次執行
`cifar10.load_data()`時，由於
之前已經下載檔案，所以就不
需要再執行下載，只需要讀取
檔案，執行時間就不會太久。

9.1 下載CIFAR-10資料

- 查看cifar10資料

11. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
print('train:',len(x_img_train))  
print('test:',len(x_img_test))
```

```
train: 50000  
test: 10000
```

以上執行結果，你可以看到第1維度是筆數，資料可分為2部分：

- train 訓練資料50000筆。
- test測試資料10000筆。

9.2 查看訓練資料

9.2 查看訓練資料

- 訓練資料是由images與labels所組成

`y_label_train`是圖形資料的真實值，每一個數字代表一種圖形類別名稱，共有10個類別：

0: airplane(飛機)

1: automobile(汽車)

2: bird(鳥)

3: cat (貓)

4: deer(鹿)

5: dog(狗)

6: frog(青蛙)

7: horse(馬)

8: ship(船)

9: truck(貨車)

9.2 查看訓練資料

- Images的shape 形狀：使用.shape方法查看x_img_train的shape

3. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
x_img_train.shape
```

```
(50000, 32, 32, 3)
```

第1維度是筆數

第2,3維度影像
大小是32X32

第4維度是RGB三
原色，所以是3

9.2 查看訓練資料

- 第0筆images影像的內容

3. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
x_img_test[0]
array([[158, 112, 49],
       [159, 111, 47],
       [165, 116, 51],
       ...,
       [137, 95, 36],
       [126, 91, 36],
       [116, 85, 33]],

       [[152, 112, 51],
       [151, 110, 40],
       [159, 114, 45],
       ...,
       [136, 95, 31],
       [125, 91, 32],
       [119, 88, 34]],

       [[151, 110, 47],
       [151, 109, 33],
       [158, 111, 36],
       ...,
```

3個數字是RGB的三原色

每一點都是由是RGB三原色所組成，RGB共有3個數字，數字的範圍由0至255，代表圖形RGB的三原色。

9.2 查看訓練資料

- `y_label_train`的shape形狀

3. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
y_label_train.shape
```

```
(50000, 1)
```


9.3 查看多筆images與label

9.3 查看多筆images與label

- 定義label_dict字典：先以python字典dict定義每一個數字所代表的圖形類別名稱

3. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
label_dict = {0:"airplane",1:"automobile",2:"bird",3:"cat",4:"deer",  
              5:"dog",6:"frog",7:"horse",8:"ship",9:"truck"}
```

9.3 查看多筆images與label

- 修改plot_images_labels_prediction()函數

3. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
import matplotlib.pyplot as plt
def plot_images_labels_prediction(images, labels, prediction, idx, num=10):
    fig = plt.gcf()
    fig.set_size_inches(12, 14)
    if num>25:num=25
    for i in range(0, num):
        ax = plt.subplot(5, 5, 1+i)
        ax.imshow(images[idx], cmap='binary')
        title = str(i) + ', ' + label_dict[labels[i][0]]
        if len(prediction) > 0:
            title += '=>' + label_dict[prediction[i]]
        ax.set_title(title, fontsize=10)
        ax.set_xticks([]);ax.set_yticks([])
        idx+=1
    plt.show()
```

為了方便我們查看多筆資料images與labels，我們將修改第6章所建立的plot_images_labels_prediction()，並且使用label_dict字典，將label與prediction的0~9數字轉換為圖形類別名稱。

9.3 查看多筆images與label

- 查看訓練資料前10筆資料：

3. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
plot_images_labels_prediction(x_img_train,y_label_train,[],0)
```



使用plot_images_labels_prediction()顯示訓練資料前10筆資料。因為還沒有預測資料，所以predictio參數輸入空的list[]。

9.4 將image進行預處理

9.4 將image進行預處理

- 查看訓練資料第1個影像的第1個點

3. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
x_img_train[0][0][0]  
array([59, 62, 63], dtype=uint8)
```

你可以看到每一點，共有3個數字，分別表
RGB[59,62,63]。

9.4 將image進行預處理

- 將照片影像image的數字標準化

3. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
x_img_train_normalize = x_img_train.astype('float32') / 255.0  
x_img_test_normalize = x_img_test.astype('float32') / 255.0
```

image的數字標準化，可以提高模型的準確率，因為image的數字是0到255，所以最簡單標準化的方式是除以255。

9.4 將image進行預處理

- 查看照片影像image的數字標準化後的結果

3. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
x_img_train_normalize[0][0][0]  
array([0.23137255, 0.24313726, 0.24705882], dtype=float32)
```

全部的數值都在0與1之間。

9.5

將label進行資料預處理

使用cifar-10資料集，我們希望預測的是影像的類型。

例如:「船」的影像的label是8，經過One-hot Encoding轉換為0000000010，而這10個數字正好對應到輸出層的10個神經元。

9.5 將label進行資料預處理

- 查看label原本的shape形狀

3. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
y_label_train.shape
```

```
(50000, 1)
```

從以上執行結果可以看到，
資料共計50000筆，每一筆
是1個0~9的數字。

9.5 將label進行資料預處理

- 查看前5筆資料

3. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
y_label_train[:5]
```

```
array([[6],  
       [9],  
       [9],  
       [4],  
       [1]], dtype=uint8)
```

0~9的數字，代表
影像的分類

9.5 將label進行資料預處理

- 將label標籤欄位轉換為Onehot encoding

3. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
from keras.utils import np_utils  
y_label_train_OneHot = np_utils.to_categorical(y_label_train)  
y_label_test_OneHot = np_utils.to_categorical(y_label_test)
```

- keras提供np_utils.to_categorical法，可以進行One-hot encoding轉換。
- 將訓練資料與測試資料的label標籤欄位，進行One-hot encoding 轉換。

9.5 將label進行資料預處理

- One-hot encoding 轉換之後的label標籤欄位

3. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
y_label_train_OneHot.shape
```

```
(50000, 10)
```

共計50000筆，每一筆10個0或1的組合。

9.5 將label進行資料預處理

- 查看轉換為One-hot encoding之後的結果

3. 在命令列輸入下列程式碼，並且按下Shift+Enter執行。

```
y_label_train_OneHot[:5]  
array([[0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],  
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 1.],  
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 1.],  
       [0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],  
       [0., 1., 0., 0., 0., 0., 0., 0., 0., 0.]], dtype=float32)
```

以第1筆資料說明：

原來的真實值是6，也就是frog，執行One-hot encoding轉換後，變成0或1的組合，只有第6個數字(由0算起)是1,其餘都是0。

9.6

結論

9.5 將label進行資料預處理

- 介紹如何下載並且讀取CIFAR-10資料集
- 介紹CIFAR-10資料集的特色
- 完成資料的預處理
- 下一章：使用Keras建立卷積神經網路(CNN)，
建立模型、訓練模型、進行預測