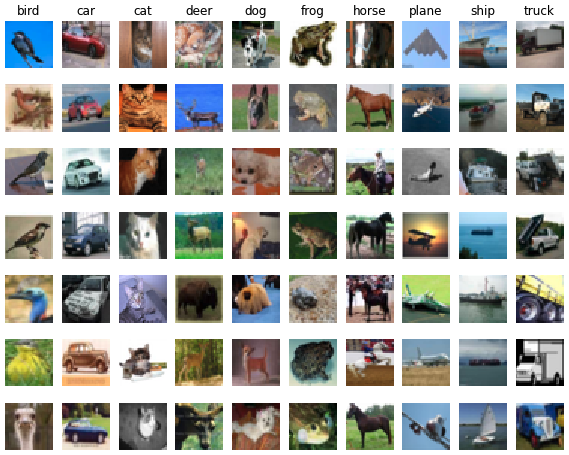
HW3: CIFAR10 and DataLoader

B083040029 邱品諺

1. CIFAR10-image-dataset

CIFAR10為包含了50000張32x32的RGB圖片並分類成10個class的dataset。下圖為以每類7張圖片示意：



1. Dataset Download

宣告變數dataset作為class ImageFolderDataset的object，並從[此網站](https://i2dl.vc.in.tum.de/static/data/cifar10.zip)下載CIFAR10 dataset至指定的google drive路徑，其中ImageFolderDataset繼承了class Dataset，且Dataset繼承了Abstract Base Method (ABC)，使Dataset作為Abstract class。在Dataset中指定了兩個abstract method: \_\_getitem\_\_(), \_\_len\_\_() 使其child class自行定義函式內容。在class ImageFolderDataset中定義了四個變數：classes, class\_to\_idx, images, labels。其中\_find\_classes()回傳變數classes及class\_to\_idx，前者為以10個種類名稱所形成的list，並以字母順序排序，後者為每個種類所對應的label為何，函式中以數字0-9為代表，並以dictionary儲存，像是 {‘bird’: 0, ‘car’: 1, …}。而make\_dataset()回傳的變數images為dataset中各張圖片的路徑，而labels為各張圖片所對應的label。

1. ImageFolderDataset Implementation

\_\_len\_\_(): 此函式回傳dataset的長度，而CIFAR10的dataset為50000筆資料，因此長度為50000。此函式可使ImageFolderDataset的Instance可直接使用len()獲得dataset長度。

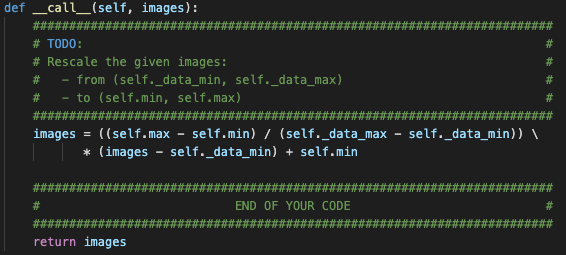
\_\_getitem\_\_(): 此函數回傳dataset中的指定項目，首先利用load\_image\_as\_numpy將路徑的圖片轉換成長度為32x32x3的NumPy array，並將該圖片對應的label以一個pair的方式存入dictionary並回傳。

1. Transform and Image Preprocessing

此部分實作了三種調整數據比例的方式：

* 1. Rescale

以圖片為例，每個pixel的亮暗程度以0-255表示，若想rescale至0-1之間，以下為計算公式：



* 1. Normalization

此部分須對每個channel執行，也就是分別對RGB三個軸normalize，計算為後數值範圍將落在[0, 1]，以下為計算公式：

* 1. Standardization

此部分也須對每個channel執行，計算完畢後每個channel的平均值為0、標準差為1，以下為計算公式：

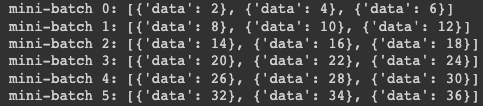
最後，利用ComposeTransform傳入欲對資料集依序進行的前處理方式以進行資料前處理，再轉換成長度為32x32x3的NumPy array及其對應的label以dictionary輸出。

1. DataLoader

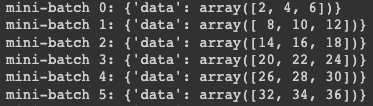
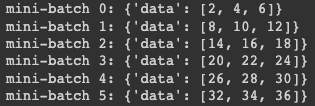
由於深度學習在資料載入及模型訓練上，資料通常是以batch為單位，就像是有好多個一樣大小的子集所組合成的資料集，因此此部分我們將自己實作DataLoader以改變資料讀取的方式。

1. Iterating over a dataset

首先此部分將利用1-100的偶數作為dataset，且每一筆資料的型態為dictionary，如：{‘data’: number}。而此部分每個batch的大小為3，首先我們可以用建立一個list持續append，如下圖：



但我們希望可以將每個batch的數值都放在同個dictionary，因此我們可以遞迴每個batch，並將其中每筆資料放進list後再放入一個新的dictionary，最後再轉換成NumPy array。



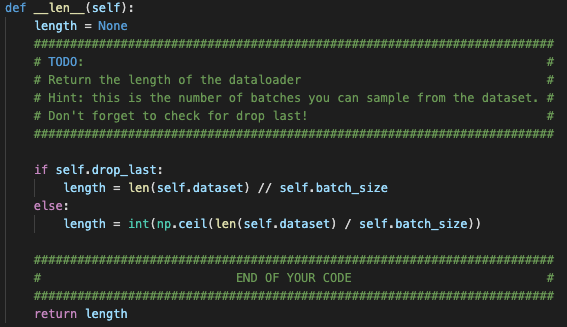
但實際上在載入dataset時，除了可以決定是否要shuffle外，通常不會一次將所有資料載入，因為這樣需要許多記憶體。因此為了節省記憶體的使用，利用generator並搭配for迴圈可以每次載入一部分的資料直到raise StopIteration並結束資料載入。

1. DataLoader Class Implementation

此部分為自己實作class DataLoader，主要為實作\_\_len\_\_()、\_\_iter\_\_()，不過和上個部分不同的地方為需要考慮每個batch是否能剛好裝到預設的batch大小。

\_\_len\_\_():

此函式決定batch的數量，因此只需將dataset的長度除以每個batch的大小，並考慮是否需要將最後未能裝滿的batch作為一個batch即可。



\_\_iter\_\_():

此部分其實就是將第一部分實作於此函式中，首先先決定資料載入順序的index，並決定是否需要shuffle載入，接下來以設定的batch大小為單位形成一個個batch，並透過generator和for迴圈將每個batch回傳。最後如果需要將未能裝滿的batch回傳，只需看最後欲裝進batch的list是否為空便可判斷。

