

人工智慧與機器學習-作業一

黃彥程 109403526 資管三 A

Colab 連結:

https://colab.research.google.com/drive/1DJ_DcjvbvgACnEIY7HNtG5KKHjzeYkJa

Test Accuracy:

56.29%

過程:

檔案上傳改由本地端上傳(但有時候真的太慢會偷用原本的)

```
import random
import os

# 大家盡量先把資料保存在本地端，然後要訓練時用本地端上傳做訓練
# 以節省學術網路資源，避免 IP 被封鎖

if not os.path.isfile("./train.zip"):
    !wget -O train.zip "http://140.115.83.111/files/art/train.zip"
    !wget -O test.zip "http://140.115.83.111/files/art/test.zip"
    !unzip train.zip
    !unzip test.zip
else:
    !echo "檔案已存在"

...

from google.colab import files
if not (os.path.isdir("./train_resized/") and os.path.isdir("./test_resized/")):
    uploaded = files.upload()
    !unzip train.zip
    !unzip test.zip
else:
    !echo "檔案已存在"
...
```

將所有作家存入 `author_dictionary`，並用 `index` 變數給予代表的數字。

```
▶ # 請建立將英文映射成數字的 dict * EX: Van_Gogh --> 0
author_dict = dict()
def make_author_dict():
    index = 0

    for i in artists["name"]:
        author_dict[i] = index
        index += 1

    return author_dict

class_name = make_author_dict()

# 請建立將數字映射成英文的 dict * EX: 0 --> Van_Gogh
rev_class_name = {value: key for key, value in author_dict.items()}
```

查詢字串處理方式後發現可以使用 `rpartition` 來處理從右邊數過來的第一個分割符。

```
def get_label(pic_name):
    # 請取出 label 並轉成數字
    # EX: Claude_Monet_1.jpg -> Claude_Monet -> 1

    # 從最右邊開始找分割符並將畫家名拆出來放
    sep_pic_name = pic_name.rpartition("_")
    # e.g.: Albrecht_Du_rer_2 -> ('Albrecht_Du_rer', '_', '2')
    pic_name = sep_pic_name[0]

    # 回傳對應之label
    label = author_dict[pic_name]

    return label

def get_path(dir, pic_name):
    # 請將路徑合併
    # EX: ./train_resized/ + Claude_Monet_1.jpg => ./train_resized/Claude_Monet_1.jpg
    path = dir + pic_name

    return path

def make_paths_label(dir):
    img_list = os.listdir(dir)
    paths = []
    labels = []

    # 將preprocess完成的 path + label 用 for 迴圈放入 paths 和 labels
    for path in img_list:
        path_append = get_path(dir, path)
        label_append = get_label(path)
        paths.append(path_append)
        labels.append(label_append)

    # 將 labels 轉成 onehot
    # todo
    onehot_labels = keras.utils.to_categorical(labels, num_classes)
    return paths, onehot_labels
```

設定圖片寬高，並設定 Shuffle_buffer 為 1000

```
[11] # 決定你輸入模型的圖片長寬
      IMG_WIDTH = 256
      IMG_HEIGHT = 256
      IMG_SIZE = None
      # shuffle buffer size
      SHUFFLE_BUFFER = 1000
```

imshow 裡 img 旁要加入/255，不然會異常。

```
# 取出 Tensor 圖片來看看
plt.figure(figsize=(12, 8))
for index, (img, label) in enumerate(full_ds.take(6)):
    l = np.argmax(label.numpy())
    plt.subplot(2, 3, index + 1)
    plt.imshow(img/255)
    plt.title("Label number : {} \n Author Name : {}".format(l, rev_class_name[1]))
    plt.axis("off")
```

設定 Batch 為 32

```
# 添加 batch
# todo
BATCH_SIZE = 32

train_ds = train_ds.batch(BATCH_SIZE)
val_ds = val_ds.batch(BATCH_SIZE)
```

自訂資料增強器

```
[16] input_shape = (IMG_HEIGHT, IMG_WIDTH, 3)

# 定義資料增強器
datagen = keras.preprocessing.image.ImageDataGenerator(
    rotation_range=0.1, # 隨機旋轉角度範圍
    width_shift_range=0.1, # 隨機水平偏移範圍
    height_shift_range=0.1, # 隨機垂直偏移範圍
    shear_range=0.1, # 隨機剪切範圍
    zoom_range=0.1, # 隨機縮放範圍
    horizontal_flip=True, # 隨機水平翻轉
    vertical_flip=True,
    fill_mode='nearest' # 補齊缺失部分的方法
)
```

編輯 model

```
#自編model
model = keras.Sequential([

    keras.Input(shape=input_shape),

    keras.layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.BatchNormalization(),

    keras.layers.Conv2D(128, kernel_size=(3, 3), activation="relu"),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.BatchNormalization(),

    keras.layers.Conv2D(256, kernel_size=(3, 3), activation="relu"),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.BatchNormalization(),

    keras.layers.Conv2D(512, kernel_size=(3, 3), activation="relu"),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.BatchNormalization(),

    keras.layers.Conv2D(1024, kernel_size=(3, 3), activation="relu"),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.BatchNormalization(),

    keras.layers.GlobalAveragePooling2D(),
    keras.layers.Dropout(0.25),
    keras.layers.Dense(512, activation="relu"),
    keras.layers.Dropout(0.2),
    keras.layers.Dense(num_classes, activation="softmax"),

])
```

每次訓練紀錄:

32 64 256 1024 128	Epochs 100	Dropout 0.35 0.25	Ac 49%
128 64 256 512 256	Epochs 75	Dropout 0.35 0.25	Ac 49%
512 128 1024 64	Epochs 75	Dropout 0.35 0.25	Ac 33%
512 256 1024 512 128 256	Epochs 125	Dropout 0.35 0.25	Ac 50%
64 128 256 512 1024	Epochs 100	Dropout 0.35 0.25	Ac 50%
32 64 128 256 512 1024	Epochs 100	Dropout 0.35 0.25	Ac 27%
64 128 256 512 1024	Epochs 100	Dropout 0.25 0.2	Ac 56.29%

設定要訓練次數

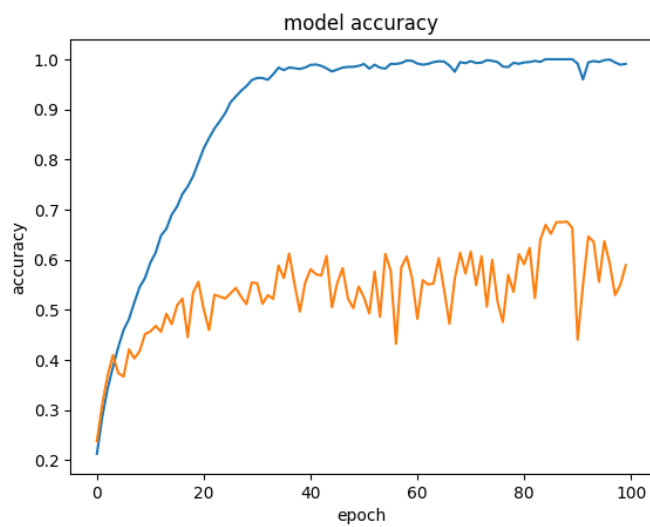
```
# todo
EPOCHS = 100

adam = tf.keras.optimizers.Adam(learning_rate=0.0002)

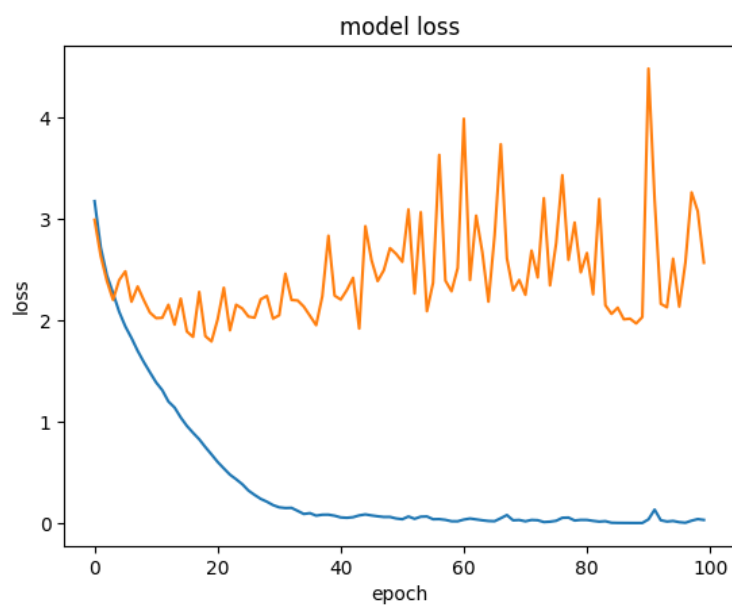
model.compile(loss="categorical_crossentropy", optimizer=adam, metrics=["accuracy"])

history = model.fit(train_ds, batch_size=BATCH_SIZE, epochs=EPOCHS, validation_data=val_ds)
```

Accuracy



Loss



Test loss & Accuracy

```
27/27 [=====] - 4s 56ms/step - loss: 2.7850 - accuracy: 0.5629  
Test loss: 2.7850468158721924  
Test accuracy: 0.5628742575645447
```

Accuracy : 56%

讀入上傳資料做預測作者

```
def predict_author(img):  
    # 寫個單圖片模型預測 function  
    # input : opencv img (height,width,3)  
    # output : 某個作家名字 E.g. Claude_Monet  
    #  
    # 參考步驟:  
    # 1. expand img dimension (height,width,3) -> (1,height,width,3)  
    # 2. 丟入模型 model.predict  
    # 3. 取出 softmax 後 (50,) 取最大值的 index 作為辨識結果  
    # 4. 將辨識結果轉為畫作家名字  
  
    author_name = ""  
    img = np.expand_dims(img, 0)  
    img_pred = model.predict(img)  
    max = tf.nn.softmax(img_pred)  
    author_name = rev_class_name.get(np.argmax(max))  
  
    return author_name
```

上傳阿伯前來比讚照片做測試

predict author : Rene_Magritte



心得:

這次是我第一次接觸機器學習的作業，從甚麼都不懂，到老師講得我好像懂，到查了一堆資料發現我還在資料前處理感到很挫折，再到真正去建立並測試 model。

我覺得雖然有挫折成分在，但是是一份有趣的作業，因為大一剛上來聽說甚麼 Python AI, C Data structure, Java class based programming，都是聽說很重要又很熱門的程式語言，但那都只是聽說，當我真的第一次使用 Python 來實作 CNN 有一種：「挖原來是這樣做的！」的感覺，每次訓練模型都很期待正確率會是多少，這種新奇感是我之前在學其他學科都沒有的，知道原來電腦是這樣學習得之後，因為我多少也對人類到底如何學習有興趣，所以看到電腦的學習方式之後也有一種很奇妙的感覺，看起來是模仿但還不完全的樣子。

讚，還蠻好玩的，雖然斷線要重跑還蠻累人的。

