

vgg16

程式碼註解

Jerry Wu & NTPU Team

```
#創建一個GrphDef圖形化流程結構
graph_def = tf.GraphDef()
#將模型導出單個文件定義權重
graph_def.ParseFromString(fileContent)
```

- tf.GraphDef是用來取得圖片
- .ParseFromString用來解析讀取的內容，內容為二位元文件

```
#上傳圖片功能
```

```
cat = utils.load_image("Common-dog-behaviors-explained.jpg")
```

- `utils.load_image`用來讀取jpg檔的資料

```
with tf.Session() as sess:  
    init = tf.initialize_all_variables()  
    sess.run(init)
```

- `tf.initialize_all_variables()`:在我們使用變量時，必須先初始化變量，初始化前變量會是隨意的，這時進行操作有可能會影響到電腦內存，所以必須先將變量初始化為零

```
#使用all函式判斷image是否符合格式  
assert (0 <= img).all() and (img <= 1.0).all()
```

- assert:用來判斷是否為真，用來偵測系統是否出錯
- .all:判斷img是否有'0'或有陣列有空

```
pred = np.argsort(prob)[::-1]
```

- np.argsort:將輸入矩陣排列以小到大排列，[::-1]則是命令輸出由大而小排列

- ```
conv = tf.nn.conv2d(bottom, filt, [1, 1, 1, 1], padding='SAME')
```

- 第一個輸入是一個圖片的tensor包含圖片長,寬,維度,數量

第二個輸入是filter就是我們的卷積核，一樣是以一個tensor  
當輸入包括卷積核的(長,寬,圖像通道數,卷積核數量)

第三個輸入是strides代表卷積核移動的步伐，padding:代  
表需要在圖片最外層補零'SAME'代表需補零，輸出不影響  
圖片大小。

```
return tf.nn.max_pool(bottom, ksize=[1, 2, 2, 1],
 strides=[1, 2, 2, 1],
 padding='SAME', name=name)
```

- 和卷積核很像第一個輸入通常是接在一個feature maps後面一樣是包含長,寬,維度,數量
- 第二個參數是池化核大小[1,長,寬,1]
- 第三個參數是滑動步長
- 第四個參數一樣是在圖片外圍需不需要補零



```
with tf.variable_scope(name) as scope:
```

- 如需要重複使用參數變量

```
fc = tf.nn.bias_add(tf.matmul(x, weights), biases)
```

- `tf.nn.bias_add(value, bias)`:在多維參數加上一維偏差值

```
self.relu7 = tf.nn.dropout(self.relu7, 0.5)
```

- `tf.nn.dropout(x, keep_drop):`
- 第一個輸入是自己的輸入數據
- 第二個輸入為了防止overfitting，取一個0~1的數字，輸出的數量會是原本的keep\_drop倍，其他數值都設為零，權重質也會被keep\_drop倒數倍加在剩下的數值中，以下範例為keep\_drop設為0.5

```
Original S =
[[1. 1. 1. 1. 1. 1. 1. 1.]
 [3. 3. 3. 3. 3. 3. 3. 3.]
 [5. 5. 5. 5. 5. 5. 5. 5.]]
Dropout S =
[[2. 2. 2. 0. 0. 2. 0. 0.]
 [6. 0. 6. 0. 6. 6. 0. 0.]
 [10. 10. 10. 10. 10. 0. 0. 10.]]
```