

# Cats vs Dogs with Data Augmentation

7107029022 資管碩二 邱靖詒

## ◆ Dataset Download

Data : <https://www.kaggle.com/c/dogs-vs-cats/data>

資料夾「train/」包含 25000 張狗和貓的 jpg 檔案，每一張圖片標記如下：

cat.0.jpg  
...  
cat.124999.jpg  
dog.0.jpg  
...  
dog.124999.jpg

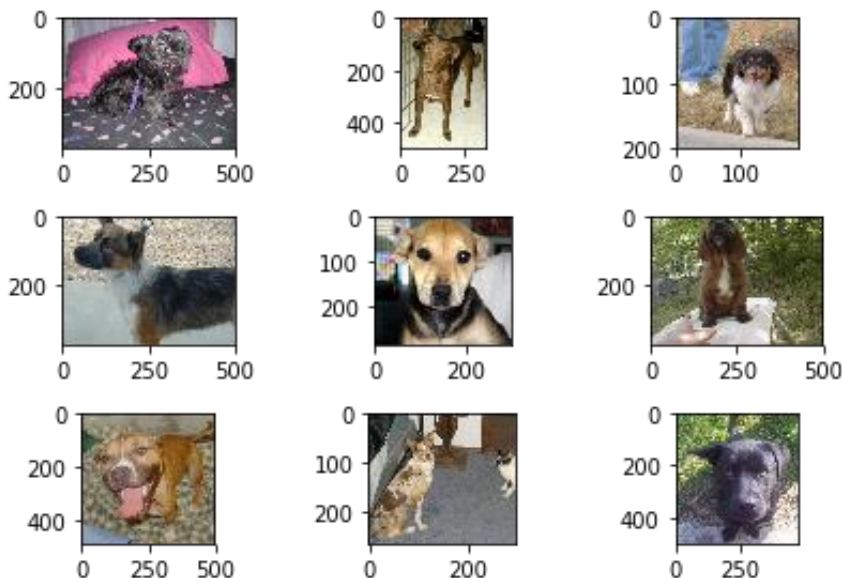
## ◆ Load Packages

```
In [16]: 1 import os, shutil
2 from matplotlib import pyplot
3 from matplotlib.image import imread
4 from os import listdir
5 from os import makedirs
6 from os import listdir
7 from shutil import copyfile
8 from random import seed
9 from random import random
10 from numpy import asarray
11 from numpy import save
12 from keras import layers
13 from keras import models
14 from keras import optimizers
15 from keras.utils import to_categorical
16 from keras.models import Sequential
17 from keras.layers import Conv2D
18 from keras.layers import MaxPooling2D
19 from keras.layers import Dense
20 from keras.layers import Flatten
21 from keras.optimizers import SGD
22 from keras.preprocessing.image import load_img
23 from keras.preprocessing.image import img_to_array
24 from keras.preprocessing import image
25 from keras.preprocessing.image import ImageDataGenerator
```

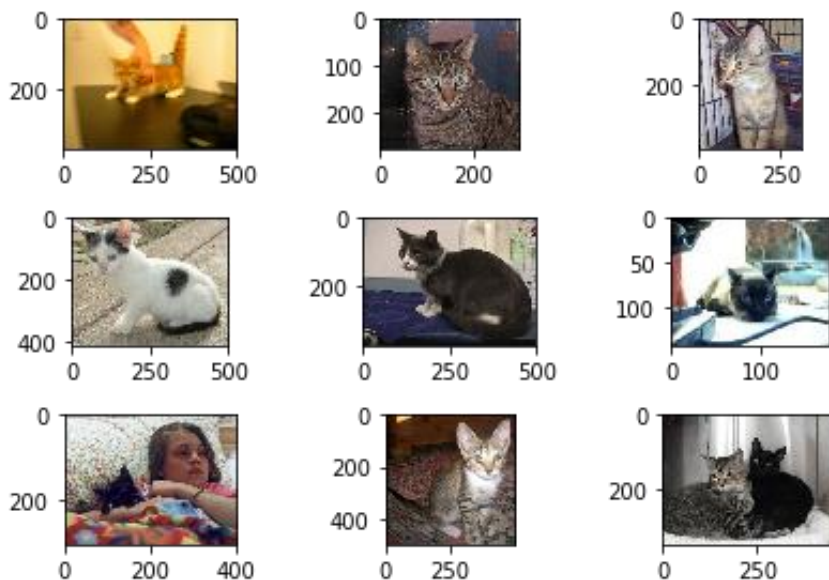
## ◆ Plot Dog and Cat Photos

分別印出貓和狗前 9 張圖片，可以看到有些照片是橫向格式，有些是縱向格式，有些是正方形。

```
In [7]: 1 folder = 'dogs-vs-cats/train/train/'
2 for i in range(9):
3     pyplot.subplot(330 + 1 + i)
4     filename = folder + 'dog.' + str(i) + '.jpg'
5     image = imread(filename)
6     pyplot.imshow(image)
7 pyplot.tight_layout()
8 pyplot.show()
```



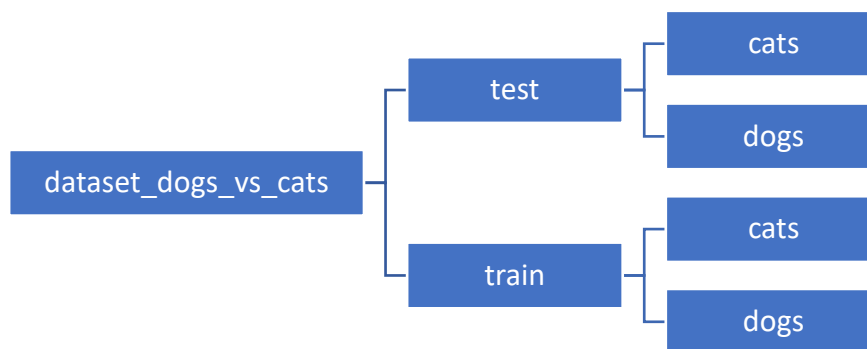
```
In [8]: 1 for i in range(9):
2         pyplot.subplot(330 + 1 + i)
3         filename = folder + 'cat.' + str(i) + '.jpg'
4         image = imread(filename)
5         pyplot.imshow(image)
6         pyplot.tight_layout()
7         pyplot.show()
```



還可以看到一張照片，幾乎看不見貓（左下角），另一隻有兩隻貓（右下角）。這說明適合該問題的任何分類器都必須具有魯棒性（Robustness），在干擾或不確定的情況下仍能保持它們的特徵行為。

#### ◆ Pre-Process Photos into Standard Directories

(i) 建立擴增資料夾



```
In [13]: 1 dataset_home = 'dataset_dogs_vs_cats/'
2         subdirs = ['train/', 'test/']
3         for subdir in subdirs:
4             labldirs = ['dogs/', 'cats/']
5             for labldir in labldirs:
6                 newdir = dataset_home + subdir + labldir
7                 mkdirs(newdir, exist_ok=True)
```

(ii) 資料夾新增數據

隨機決定將 25% 的圖片保留到測試 test 數據集中，透過固定的隨機數生成器的種子 seed(1) 來讓每次運行時都德得相同的數據分割。

```
In [15]: 1 seed(1)
2 val_ratio = 0.25
3 src_directory = 'dogs-vs-cats/train/train/'
4 for file in listdir(src_directory):
5     src = src_directory + '/' + file
6     dst_dir = 'train/'
7     if random() < val_ratio:
8         dst_dir = 'test/'
9     if file.startswith('cat'):
10        dst = dataset_home + dst_dir + 'cats/' + file
11        copyfile(src, dst)
12    elif file.startswith('dog'):
13        dst = dataset_home + dst_dir + 'dogs/' + file
14        copyfile(src, dst)
```

## ◆ Image Data Augmentation

rescale 重新縮放比例：1/255

width\_shift\_range 圖片寬度的某個比例：0.1 (資料提升時圖片水平偏移的幅度)

height\_shift\_range 圖片高度的某個比例：0.1 (資料提升時圖片垂直偏移的幅度)

horizontal\_flip 水平翻轉：True (進行隨機水平翻轉)

target\_size=(200, 200)：所有圖像大小調整成 150×150

batch\_size：64

class\_mode：因為使用二元交叉熵 binary\_crossentropy 作為損失，所以需要二元標籤

```
In [17]: 1 train_datagen = ImageDataGenerator(rescale=1.0/255.0,width_shift_range=0.1, height_shift_range=0.1, horizontal_flip=True)
2 test_datagen = ImageDataGenerator(rescale=1.0/255.0)
3 train_it = train_datagen.flow_from_directory('dataset_dogs_vs_cats/train/',class_mode='binary',
4                                             batch_size=64, target_size=(200, 200))
5 test_it = test_datagen.flow_from_directory('dataset_dogs_vs_cats/test/',class_mode='binary',
6                                           batch_size=64, target_size=(200, 200))

Found 18697 images belonging to 2 classes.
Found 6303 images belonging to 2 classes.
```

\*\*test 的資料不能進行擴增。

## ◆ Develop a CNN Model

### (i) 模型配置

激發函數：relu 搭配 sigmoid

padding：same，保留邊界處的卷積結果，通常會導致輸出 shape 與輸入 shape 相同

kernel\_initializer 權值初始化：he\_uniform，參數由[-limit, limit]的區間中均勻採樣獲得

optimizer 優化器：SGD

```
In [18]: 1 def define_model():
2     model = Sequential()
3     model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(200, 200, 3)))
4     model.add(MaxPooling2D((2, 2)))
5     model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
6     model.add(MaxPooling2D((2, 2)))
7     model.add(Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
8     model.add(MaxPooling2D((2, 2)))
9     model.add(Flatten())
10    model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
11    model.add(Dense(1, activation='sigmoid'))
12
13    opt = SGD(lr=0.001, momentum=0.9)
14    model.compile(optimizer=opt, loss='binary_crossentropy', metrics=['accuracy'])
15    return model
```

### (ii) 畫圖函式

```
In [23]: 1 def summarize_diagnostics(history):
2         pyplot.subplot(211)
3         pyplot.title('Cross Entropy Loss')
4         pyplot.plot(history.history['loss'], color='blue', label='train')
5         pyplot.plot(history.history['val_loss'], color='orange', label='test')
6
7         pyplot.subplot(212)
8         pyplot.title('Classification Accuracy')
9         pyplot.plot(history.history['acc'], color='blue', label='train')
10        pyplot.plot(history.history['val_accuracy'], color='orange', label='test')
11        pyplot.tight_layout()
12        pyplot.show()
13        pyplot.close()
```

(iii) 主程式執行

```
In [23]: 1 def summarize_diagnostics(history):
2         pyplot.subplot(211)
3         pyplot.title('Cross Entropy Loss')
4         pyplot.plot(history.history['loss'], color='blue', label='train')
5         pyplot.plot(history.history['val_loss'], color='orange', label='test')
6
7         pyplot.subplot(212)
8         pyplot.title('Classification Accuracy')
9         pyplot.plot(history.history['acc'], color='blue', label='train')
10        pyplot.plot(history.history['val_acc'], color='orange', label='test')
11        pyplot.tight_layout()
12        pyplot.show()
13        pyplot.close()
```

(iv) 結果

準確率 86.102%

```
In [22]: 1 run_test_harness()
```

Found 18697 images belonging to 2 classes.  
Found 6303 images belonging to 2 classes.  
Accuracy > 86.102

