

ImageDataGenerator

– flow_from_directory(..) –

박성호 (neowizard2018@gmail.com)

Image Data Augmentation (참고: <https://youtu.be/RuZ7SI-Yfec>)

- 원본 이미지에 적절한 변형을 가해서 새로운 데이터를 만들어 내는 방식으로, 다양한 데이터를 통해 딥러닝 성능향상 기대할 수 있음



원본 이미지

원본 이미지에 변형을 가해서 새롭게 생성된 이미지

- 텐서플로는 이미지 데이터 보강을 위한 ImageDataGenerator class 제공하며, ImageDataGenerator에서는 입력 파라미터를 통해서 rescale, rotation_range, width_shift_range, height_shift_range, shear_range, horizontal_flip, vertical_flip 등의 다양한 변화를 줄 수 있음 (다양한 입력 파라미터는 TensorFlow API 문서 참고)
- ImageDataGenerator를 사용할 경우 flow(), flow_from_directory() 등의 함수를 통해 이미지 데이터를 보강할 수 있음.

ImageDataGenerator flow_from_directory() 예제

[1] ImageDataGenerator 생성

```
import tensorflow as tf

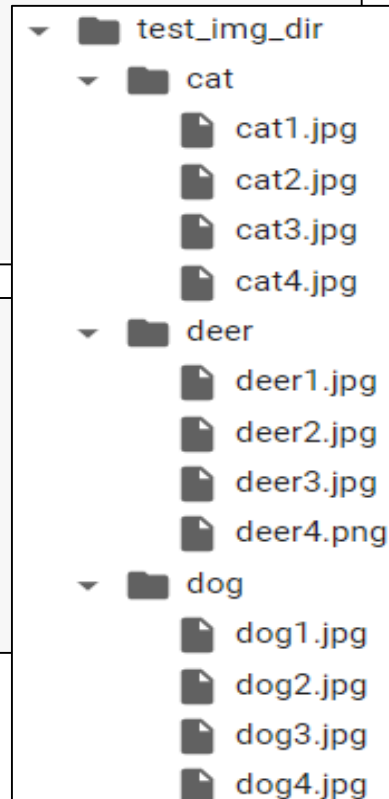
import numpy as np
import matplotlib.pyplot as plt

from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
gen = ImageDataGenerator(rescale=1./255,
                        rotation_range=10,
                        shear_range=0.2,
                        horizontal_flip=True)
```

```
import zipfile

with zipfile.ZipFile('./test_img_dir.zip') as target_file:
    target_file.extractall('test_img_dir/')
```



ImageDataGenerator flow_from_directory() 예제

[2] flow_from_directory() 적용

```
data_gen = gen.flow_from_directory(directory='./test_img_dir/',  
                                   batch_size=3,  
                                   shuffle=True,  
                                   target_size=(100, 100),  
                                   class_mode='categorical')
```

디렉토리 이름에 맞춰 자동으로 labelling

test_img_dir

- cat : cat1.jpg / cat2.jpg
- deer : deer1.jpg / deer2.jpg
- dog : dog1.jpg / dog2.jpg

```
print(data_gen.class_indices)  
print(data_gen.num_classes)  
print(data_gen.class_mode)
```

class_mode는 정답을 나타내는 방식
'binary', 'categorical', 'sparse'

```
Found 12 images belonging to 3 classes.  
{'cat': 0, 'deer': 1, 'dog': 2}  
3  
categorical
```

ImageDataGenerator flow_from_directory() 예제

[3] next() 실행 및 변형 이미지 출력

```
img, label = data_gen.next()

for i in range(len(label)):
    print('label => ', label[i])

plt.figure(figsize=(8,8))
for i in range(len(img)):

    plt.subplot(1, len(img), i+1)
    plt.xticks([])
    plt.yticks([])
    plt.title(str(label[i]))
    plt.imshow(img[i])

plt.show()
```

```
label => [0. 1. 0.]
label => [1. 0. 0.]
label => [0. 0. 1.]
```

정답을 십진수로 나타내기 위해서는 ?

[0. 1. 0.]



[1. 0. 0.]



[0. 0. 1.]

