

# Chapter07\_2.

## Cat vs Dog Classifier model

작성자 : 김진성

# 목차

1. Cats vs Dogs classifier
2. ImageDataGenerator
3. Dropout 적용
4. Image 증식

# 1. Cats vs Dogs classifier

- ✓ 케라스 창시자에게 배우는 딥러닝(<https://github.com/gilbutITbook/006975>) 참고
- ✓ Cats vs Dogs image classifier
- ✓ ImageDataGenerator : 이미지 학습을 위한 데이터 셋 생성
- ✓ Image 증식 : image 회전 각도, 수평/수직 이동, 확대 및 뒤집기 등의 범위를 설정하여 image의 수 random 확장(과적합 해결 기법)

## 2. ImageDataGenerator

### ○ 특정 폴더의 이미지 자료를 대상으로 데이터셋 생성 기능

```
# image 디렉터리 위치 지정
base_dir = "../cats_and_dogs"
train_dir = os.path.join(base_dir, 'train_dir')
validation_dir = os.path.join(base_dir, 'validation_dir')

from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_data = ImageDataGenerator(rescale=1./255) # 이미지 정규화

# image 전처리
train_generator = train_data.flow_from_directory(
    train_dir, # image 저장 위치
    target_size=(150,150), # image resize
    batch_size=35, # batch size
    class_mode='binary') # binary label
```

## 2. ImageDataGenerator

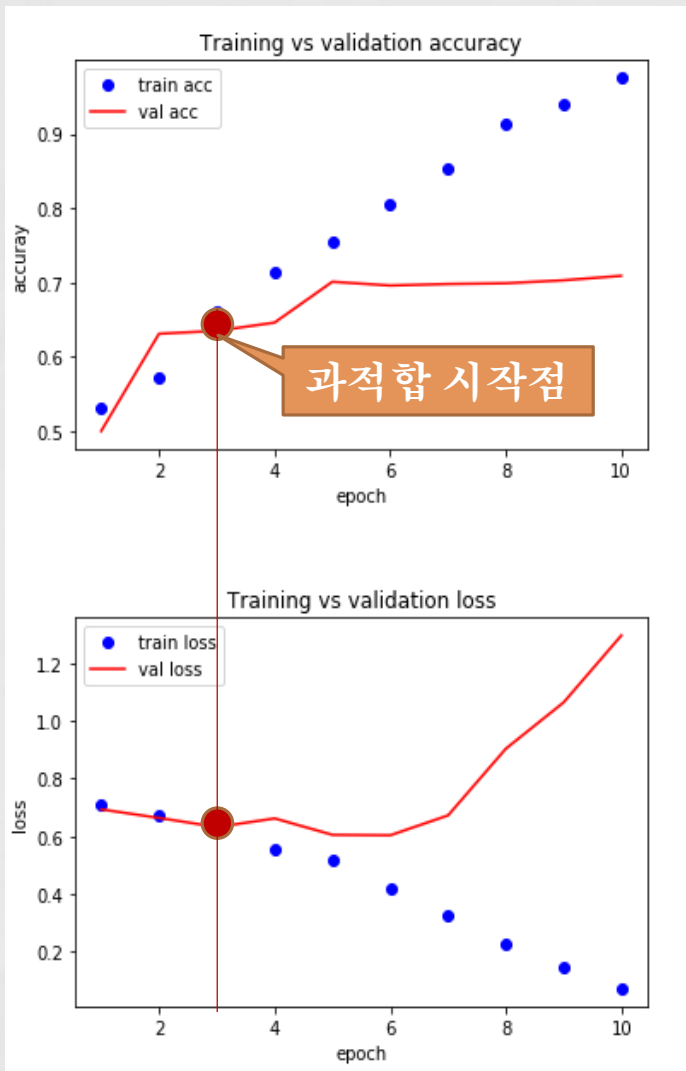
특정 폴더의 이미지 자료를 대상으로 데이터셋 생성 기능

- 주요 기능

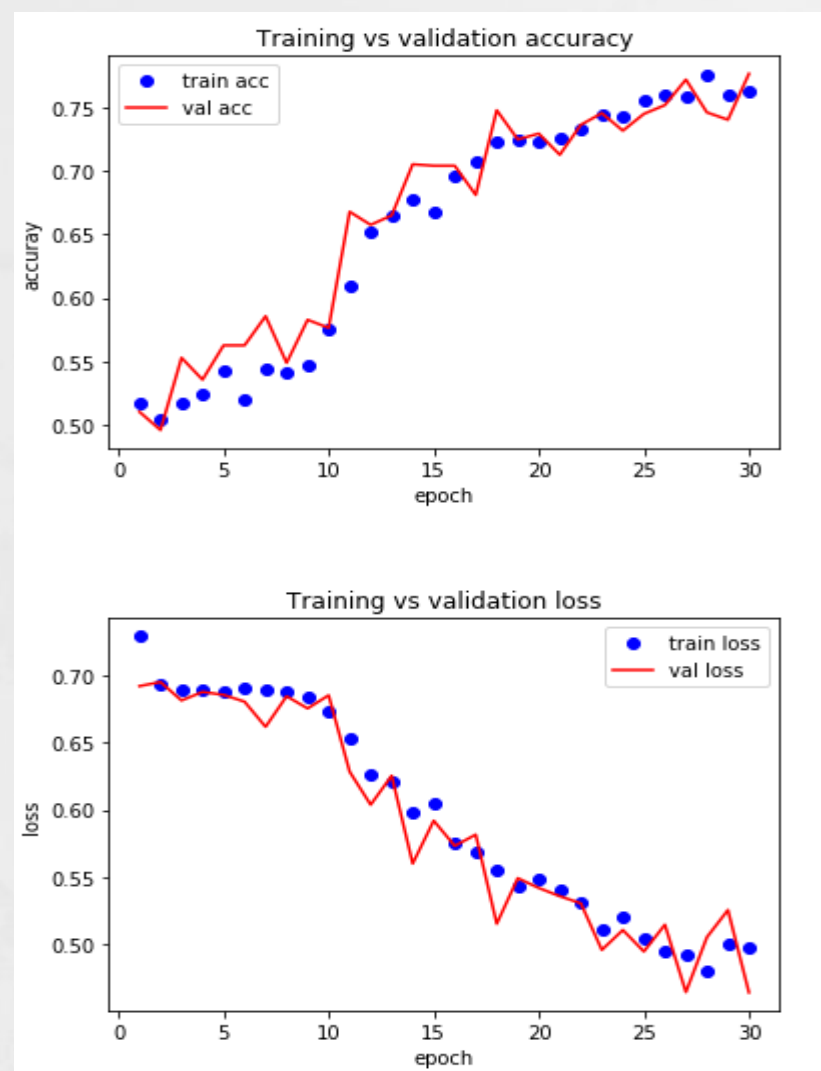
- 특정 폴더의 이미지를 학습 데이터셋으로 생성하는 기능
- 픽셀 정보 정규화 기능 : 0 ~ 1
- 이미지 크기 조절 기능 : 세로, 가로 픽셀 resize
- batch size 지정 기능 : 1회 모델에 공급할 이미지 크기
- 이미지 class 지정 기능 : 분류 이미지 범주

### 3. 과적합 해결 방안

과적합(overfitting) 발생



과적합 해결: Dropout & image 증식



## ● Dropout Tensorflow code

```
# 1. CNN Model layer
model = Sequential()

# Convolution layer1 : kernel[3,3,3,32]
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=input_shape))
model.add(MaxPool2D(pool_size=(2,2)))

# Convolution layer2 : kernel[3,3,32,64]
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))

# Convolution layer3 : kernel[5,5,64,128]
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))

# Flatten layer :4d -> 2d
model.add(Flatten())
# 드롭아웃 - 과적합 해결 - fully connected 층 이전에 배치
model.add(Dropout(0.5)) # 과적합 해결

# Affine layer(Fully connected layer1) : [n, 256]
model.add(Dense(256, activation = 'relu'))

# Output layer(Fully connected layer2) : [256, 1]
model.add(Dense(1, activation = 'sigmoid'))
```

## 4. Image 증식

- 훈련셋 이미지 크기를 임의로 늘리기 위해서 image 회전 각도, 수평/수직 이동, 확대 및 뒤집기 등의 범위를 설정하여 image의 수를 random 하게 늘려서 과적합(overfitting) 해결 기법

```
train_data = ImageDataGenerator(  
    rescale=1./255,  
    rotation_range = 40, # image 회전 각도 범위(+, - 범위)  
    width_shift_range = 0.2, # image 수평 이동 범위  
    height_shift_range = 0.2, # image 수직 이동 범위  
    shear_range = 0.2, # image 전단 각도 범위  
    zoom_range=0.2, # image 확대 범위  
    horizontal_flip=True,) # image 수평 뒤집기 범위
```