

# Convolution Neural Network

# 목차

- 데이터 증강
- 전이학습
- 바운딩박스
- 이미지 전처리

# 1. 데이터 증강

## 이미지 증강 방법

1. keras 전처리 레이어 사용
2. ImageDataGenerator 사용

# 1. 데이터 증강

## 케라스 전처리

- ◆ Resize: 사이즈 조절
- ◆ Rescaling: 표준화
- ◆ RandomFlip: 가로, 세로 반전
- ◆ RandomRotation: 회전
- ◆ RandomCrop: 이미지 잘라내기

# 이미지 사이즈 조절

```
IMG_SIZE = 200
```

```
resize_and_rescale = tf.keras.Sequential([  
    layers.experimental.preprocessing.Resizing(IMG_SIZE, IMG_SIZE),  
    layers.experimental.preprocessing.Rescaling(1./255)  
])
```

```
result = resize_and_rescale(image)  
plt.imshow(result)  
plt.show()
```



# 1. 데이터 증강

## ● 케라스 ImageDataGenerator

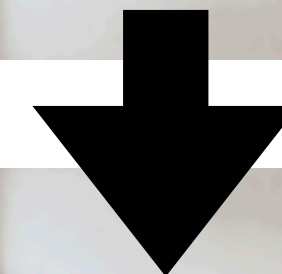
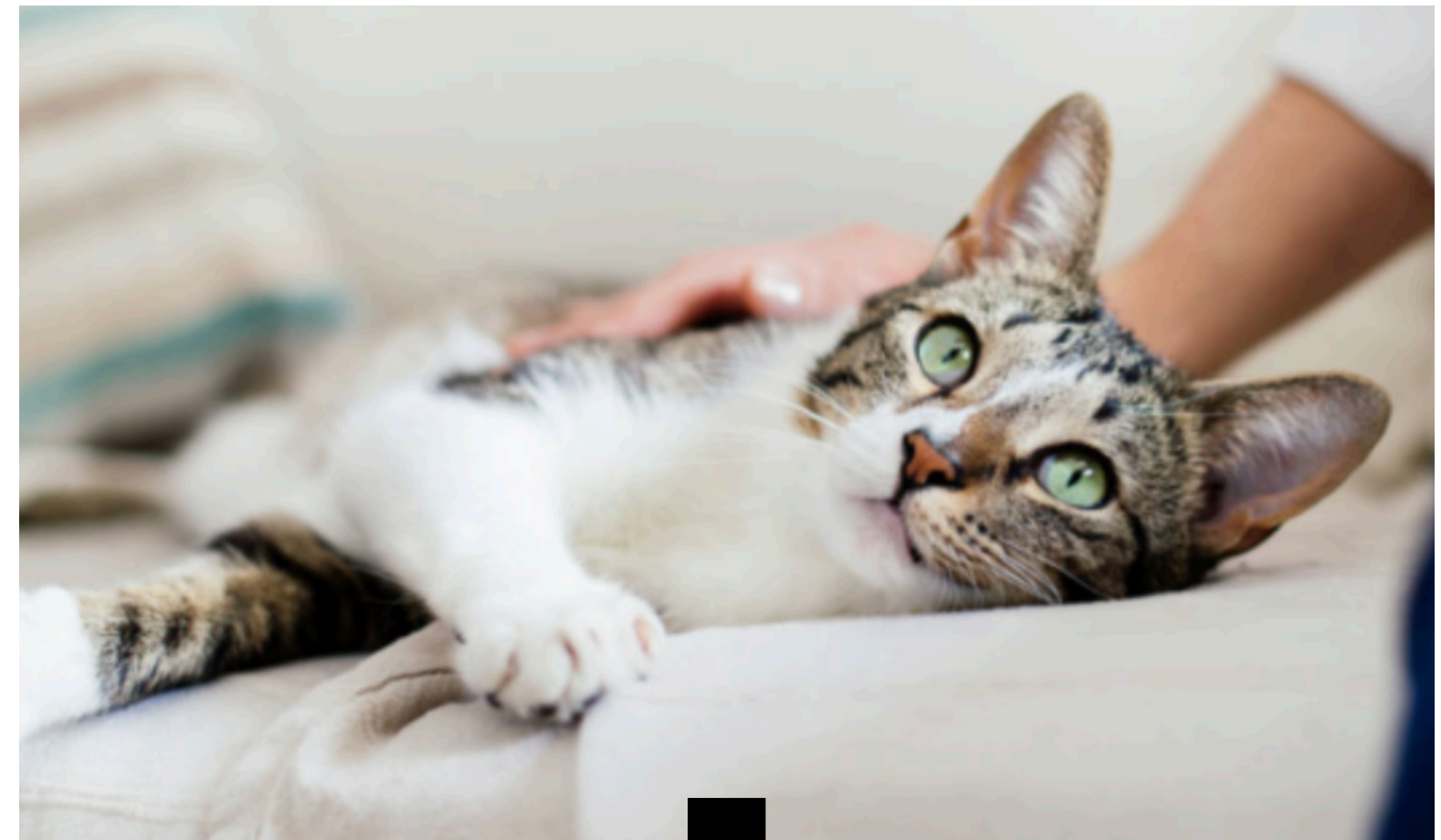
- ◆ 학습 도중 이미지에 임의 변형 및 정규화를 적용해 주고
- ◆ 변형된 이미지를 배치 단위로 불러올 수 있는 Generator을 생성.

### ◆ 공간 레벨 변형

- Flip : 상하, 좌우 반전
- Rotation : 회전
- Shift : 이동
- Zoom : 확대, 축소
- Shear : 눕히기(?)

### ◆ 픽셀 레벨 변형

- Bright : 밝기 조정
- Channel Shift : RGB 값 변경
- ZCA Whitening : Whitening 효과



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

# 1. 데이터 증강

## ● 케라스 ImageDataGenerator

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator  
data_generator = ImageDataGenerator(featurewise_center=True,  
                                     featurewise_std_normalization=True)  
data_generator = ImageDataGenerator(rescale=1/255.0)
```

## 2. 전이학습(Transfer Learning)

- 기존에 사전학습된(pre trained) 모델을 가져와, 사용하고자 하는 학습 데이터를 학습시켜 이용하는 방법
- 기존에 비슷한 도메인의 데이터를 학습한 모델이라면 현재 갖고있는 데이터가 다소 적더라도 좋은 성능을 보여 줌

## 2. 전이학습(Transfer Learning)

- 전이학습을 활용.

- 모델(ResNet, MobileNet 등)을 불러와 그대로 분류할 데이터 입력 후 분류 진행(학습 x)

- 모델을 불러온 뒤, 최상위 층(분류기)만 용도대로 재 설정하여 학습시키는 방법.  
이 때 불러온 전이학습 모델은 가중치를 동결해 학습시키지 않고,  
분류기, 또는 이후 추가한 Fully-connected layer의 가중치만 학습하여 이용.

- Fine Tuning

- 모델을 불러와 동결해 두었던 전이학습 모델의 가중치를 **(일부 또는 전부)** 학습 가능상태로 만들고 학습.  
**(일부 또는 전체)** 부분은 정해진 답이 없어 딱 잘라 말할 수 없음



## 2. 전이학습

- 전이학습 성능 개선 → fine-tuning
- 추가 성능 개선
  - 옵티마이저의 변경
  - BatchNormalization 추가
  - Dropout 추가
  - resnet50 모델까지 전체 학습 → 구조만 가져오고 전부 학습하는 것을 의미
  - Data 전처리 및 증강을 통한 학습데이터 개선
  - Fully Connected Layer 은닉층 추가 및 노드 추가
  - learning decay를 이용해 유동적인 학습 진행

데이터 준비/데이터 표시

학습 완료된 모델 준비

분류 부분의 모델 구성

모델 컴파일

콜백 생성

학습/학습 곡선

시각화/예측

# 3. 바운딩 박스(bounding box)

## ● 물체의 위치를 추정(회귀 작업)

◆ 물체 중심의 수평, 수직 좌표와 높이, 너비를 예측

◆ 모델을 크게 바꿀 필요는 없음

◆ 바운딩 박스에 널리 사용되는 지표는 IoU(Intersection over union)

- 예측한 바운딩 박스와 타겟 바운딩 박스 사이에 중첩되는 영역을 전체 영역으로 나눈 것

- tf.keras에서는 tf.keras.metrics.MeanIoU에 구현

# Recurrent Neural Network(RNN)

# 1. RNN

## ● 순환 신경망

### ▶ 입력과 출력을 시퀀스 단위로 처리

- 시퀀스란 문장 같은 단어가 나열된 것
- 이러한 시퀀스들을 처리하기 위해 고안된 모델을 시퀀스 모델
- 그중에서 RNN은 딥 러닝의 가장 기본적인 시퀀스 모델

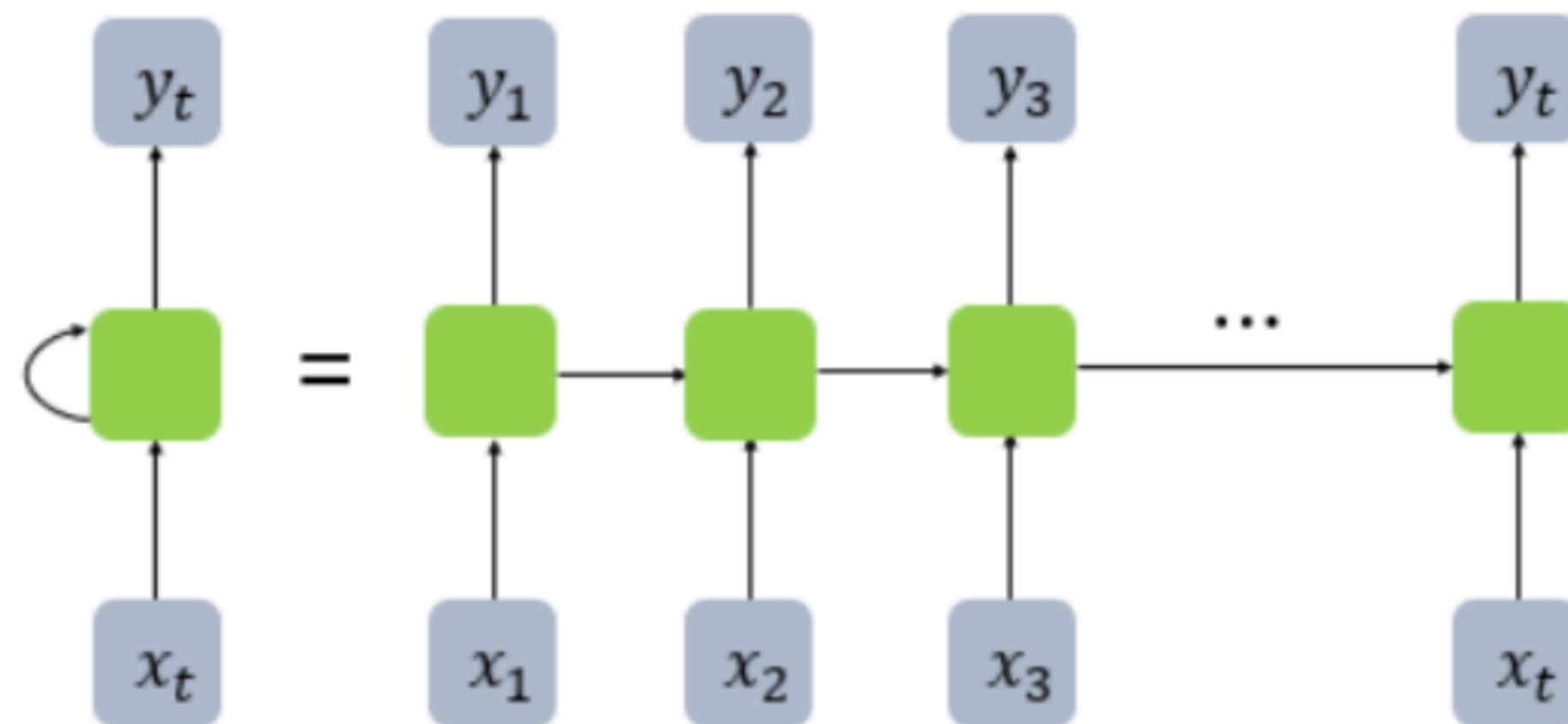
### ▶ 은닉층에서 활성화 함수를 통해 결과를 내보내는 역할을 하는 노드를 셀

- 이 셀은 이전의 값을 기억, 일종의 메모리 역할을 수행(**메모리 셀**)
- 은닉층의 메모리 셀에서 나온 값이 다음 은닉층의 메모리 셀에 입력 → 이 값을 은닉 상태

# 1. RNN

## 순환 신경망

▶ 전의 메모리 셀 출력 값과 현재의 입력 값 두 개가 입력으로 사용



▶ 입력과 출력의 길이를 다르게 설계할 수 있어 다양한 용도로 사용할 수 있는 장점

- 학습에는 back propagation의 확장인 BPTT가 사용

• 재귀적인 형태의 모델을 시간에 대해 펼쳐서 현재 시점의 에러를 최초 시점까지 전파해 학습하는 것