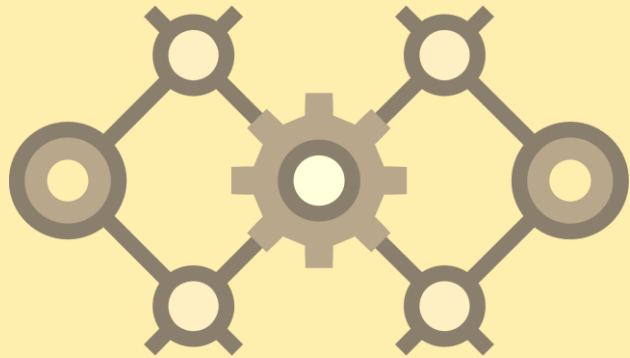




# Python을 활용한 이미지 분석

## 6. 합성곱 신경망 성능개선



## 합성곱 신경망 성능개선

**01** 합성곱 층을 통한 학습

**02** 데이터 증강

**03** 배치 정규화



1

## 합성곱 층을 통한 학습



# 1. 합성곱 층을 통한 학습



## 학습을 통한 날씨 이미지 분류

`model.fit()`을 통해 학습함

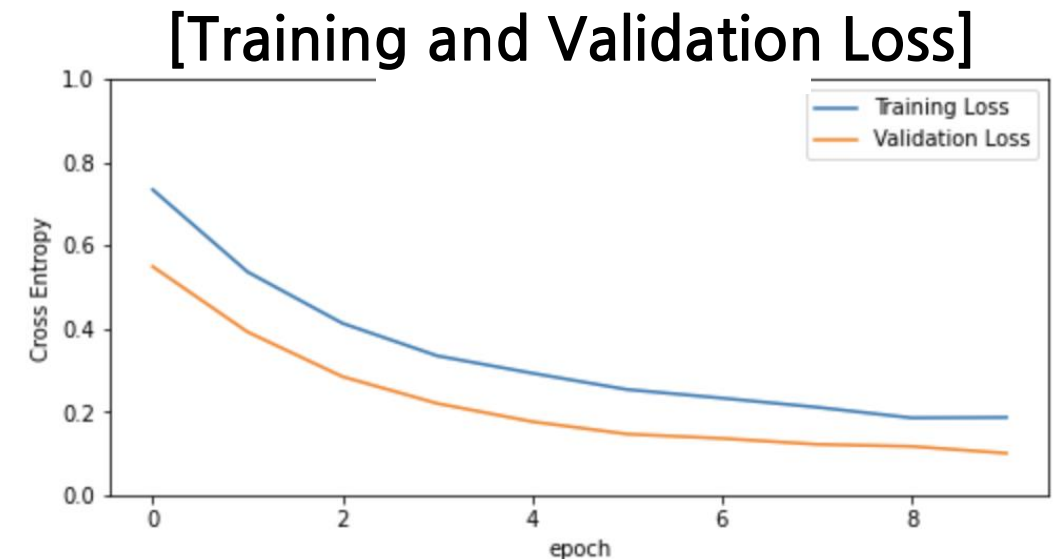
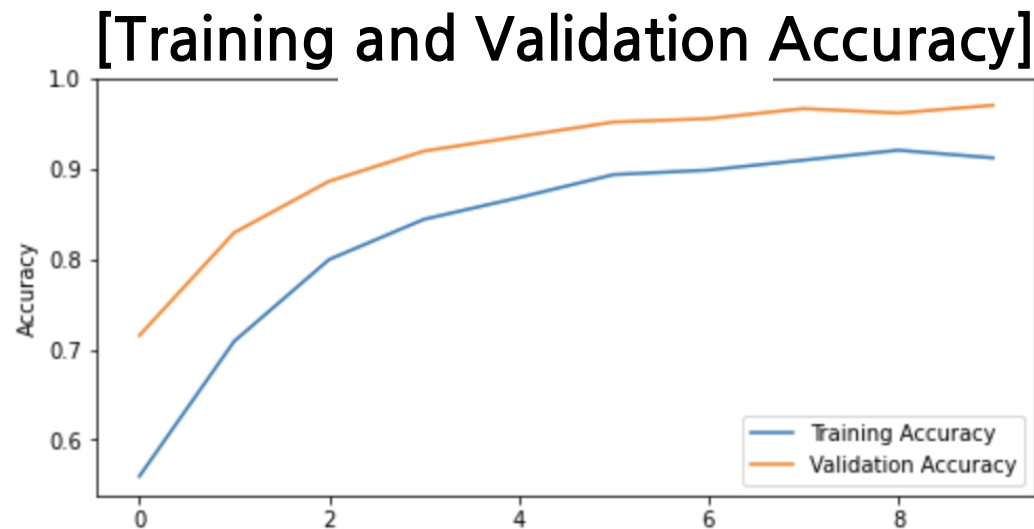
x	날씨 사진 학습 이미지
y	정답값 <b>예</b> Cloudy, Shine, Sunrise, Rainy, Foggy
batch_size	한 번에 학습할 데이터 크기
epoch	학습 횟수
verbose	로그 형식
callback	사용자 정의 함수 <b>예</b> 10번 이상 손실이 줄어들지 않으면 학습을 멈추기
validation_split	검증 데이터 비율

# 1. 합성곱 층을 통한 학습

## 학습 결과

### ■ 학습 세트와 검증 세트 비교

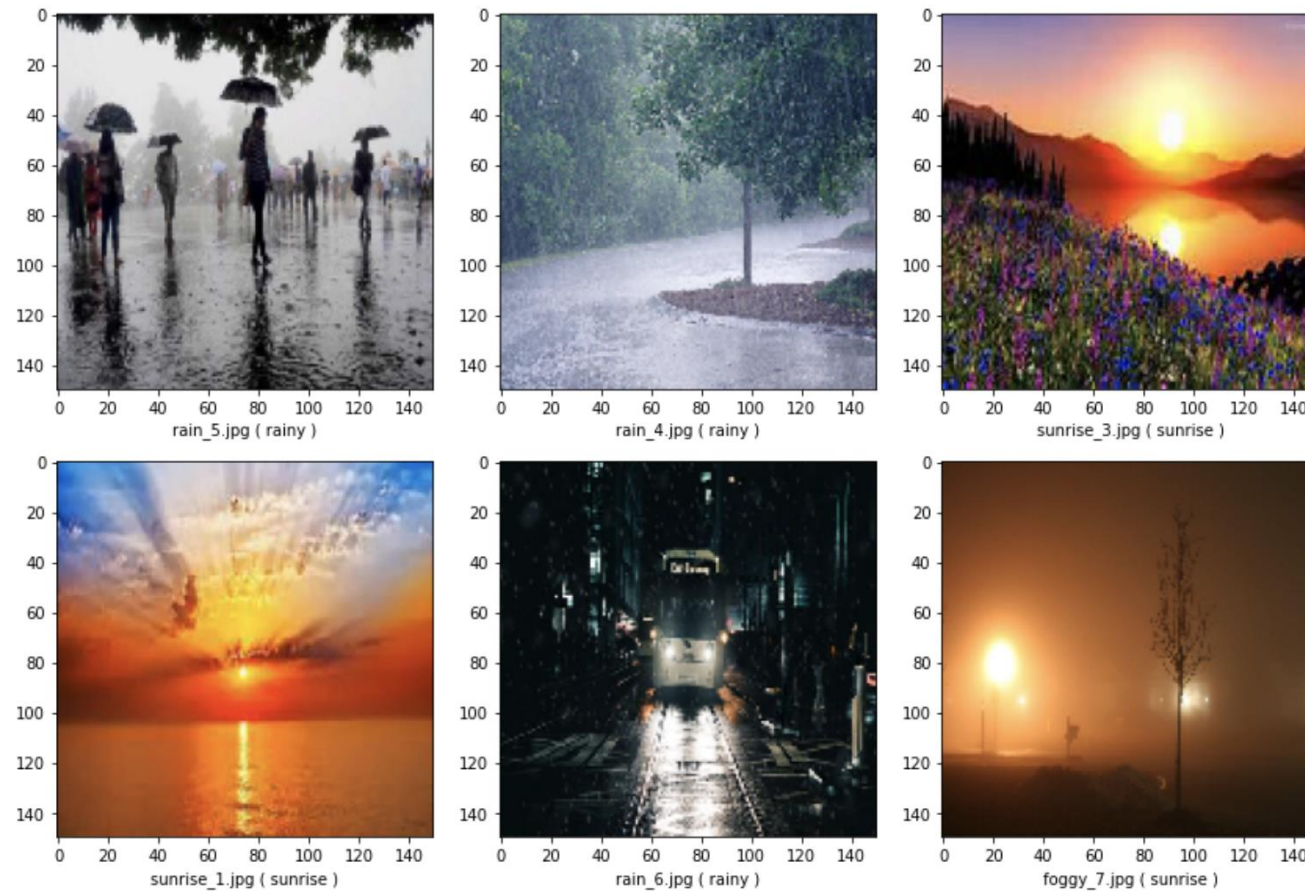
- ▶ 학습 결과의 History를 통해 Accuracy와 Loss를 비교함
  - Accuracy : 정확도로 학습, 검정 세트를 통해 정답값의 비율을 확인할 수 있음
  - Loss : 학습, 검증 세트를 통해 손실률을 확인할 수 있음
  - EarlyStopping : 학습을 하다보면 어느 시점에서 모델의 성능이 정체하는 구간이 발생, 성능 향상이 없으면 자동으로 훈련을 멈춤





`model.predict()`를 통해 예측함

- ▶ Accuracy를 통해 예측의 정확도를 측정함



# 1. 합성곱 층을 통한 학습



실습

■ 예제

문제 상황

합성곱 신경망으로 학습하기

실습 코드

```
model.fit()  
model.predict()
```

예  
시  
화  
면

```
history = model.fit(  
    X_train, y_train,  
    epochs = 10,  
    callbacks=[callbacks],  
    verbose = 2)
```

2

데이터 증강





### 데이터 증강 (Data Augmentation)

데이터 양을 늘리는 기술

- ▶ 식별하고자 하는 물체의 이미지를 변형시킬 수 있음
  - 평행 이동, 확대, 회전, 밝기 조절 등
- ▶ 이미지를 다양하게 학습할 수 있음
- ▶ 이미지 변형 시 정보가 왜곡되는 것을 유의해야 함
  - 예) 숫자 이미지 6을 회전 시 9로 오해할 수 있음
- ! **훈련 세트에만 적용**해야 함  
테스트 세트에 적용 시 현실과 다른 결과가 나올 수 있음에 유의



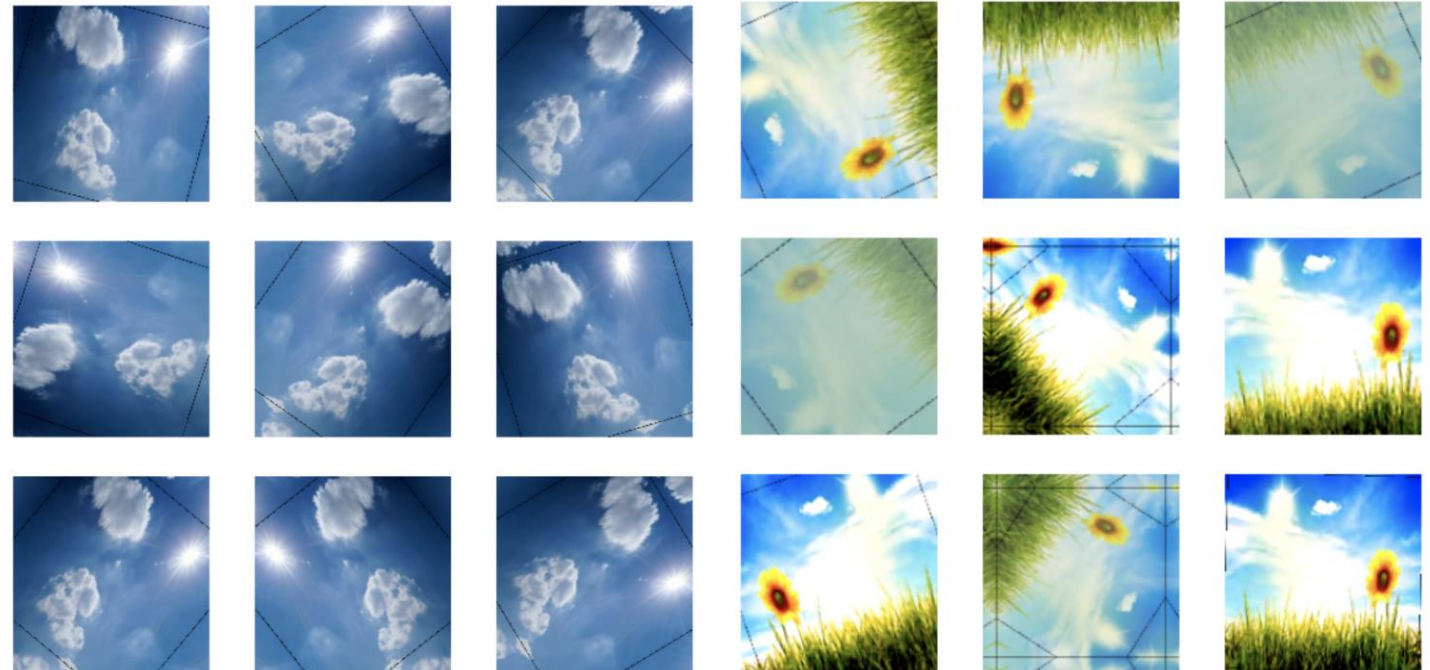
### Keras 전처리 층 사용하기

#### ■ API

```
tf.keras.layers.experimental.preprocessing
```

#### ■ 기능

- ▶ RandomContrast : 색상의 대비
- ▶ RandomCrop : 일부만 자르기
- ▶ RandomFlip : 무작위 뒤집기
- ▶ RandomRotation : 회전
- ▶ RandomZoom : 무작위, 확대/축소
- ▶ Resize : 크기 변경





### Keras 이미지 처리 도구 사용하기

#### ■ API

```
tf.keras.preprocessing.image.ImageDataGenerator
```

#### ■ 기능

- ▶ Featurewise\_center : 샘플의 특성 분포 결정
- ▶ Featurewise\_std\_normalization : 랜덤 샘플의 각 특성이 정규분포를 따르게 함
- ▶ Rotation\_range : 랜덤 샘플의 회전 범위 지정
- ▶ Width\_shift\_range : 이미지의 너비가 늘어나는 범위
- ▶ Height\_shift\_range : 이미지의 높이가 늘어나는 범위
- ▶ Shear\_range : 이미지의 기울기
- ▶ Zoom\_range : 확대/축소 범위
- ▶ Horizontal\_flip : 회전 여부





실습

예제

문제 상황

Keras의 전처리 층을 사용해 데이터 증강

실습 코드

`tf.keras.layers.experimental.preprocessing`

예  
시  
화  
면

```
IMG_SIZE = 150
```

```
data_augmentation = tf.keras.Sequential([
    tf.keras.layers.experimental.preprocessing.Resizing(IMG_SIZE, IMG_SIZE),
    tf.keras.layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical"),
    tf.keras.layers.experimental.preprocessing.RandomContrast(1),
    tf.keras.layers.experimental.preprocessing.RandomRotation(0.2),
    tf.keras.layers.experimental.preprocessing.RandomZoom(0.2),
])
```



3

## 배치 정규화

## 배치 정규화(Batch Normalization)

### ■ 특징

- 01 기울기 소실과 폭주를 해결함
- 02 값의 스케일을 변경해 입력을 정규화 함
- 03 신경망을 보다 빠르고 안정적으로 만드는 방법임
- 04 가중치 파라미터의 스케일을 규제하여 학습률을 높게 설정할 수 있기 때문에 학습 속도를 개선할 수 있음
- 05 현재 미니배치에서 평균과 표준편차를 평가하기 때문에 배치 정규화임

# 3. 배치 정규화



## 배치 정규화(Batch Normalization)

### ■ 합성곱 신경망에 배치 정규화 적용하기

- ▶ 각 층에서 활성화 함수를 통과하기 전이나 후에 모델 연산을 추가함
- ▶ 배치 정규화를 하면 훈련 세트를 표준화할 필요가 없음

```
tf.keras.Sequential([
    tf.keras.layers.Conv2D(1, (1, 1), input_shape=(None, None, 3)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Conv2D(2, 1, padding='same'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Conv2D(3, (1, 1)),
    tf.keras.layers.BatchNormalization()
])
```





실습

■ 예제

문제 상황

합성곱 신경망에 배치 정규화 층 적용

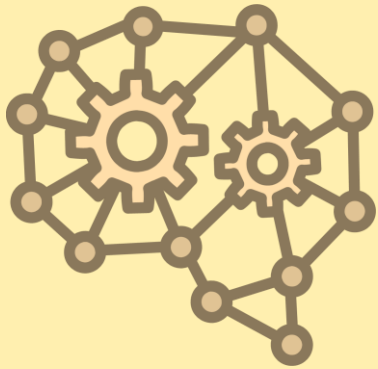
실습 코드

`tf.keras.layers.BatchNormalization()`

예  
시  
화  
면

```
tf.keras.Sequential([
    tf.keras.layers.Conv2D(1, (1, 1), input_shape=(None, None, 3)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Conv2D(2, 1, padding='same'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Conv2D(3, (1, 1)),
    tf.keras.layers.BatchNormalization()
])
```





## KEY POINT

- ▶ 합성곱 층을 통한 학습
  - `model.fit(x=학습에 사용할 이미지, y=정답값, batch_size=학습할 데이터 크기, epoch=학습 횟수)`
  - `model.predict(x=예측에 사용할 이미지)`
- ▶ 데이터 증강
  - 데이터 양을 늘리는 기술
  - 식별하고자 하는 물체의 이미지를 변형
  - 평행 이동, 확대, 회전, 밝기 조절 등
  - 이미지를 다양하게 학습이 가능
- ▶ 배치 정규화
  - 그라디언트 소실과 폭주를 해결
  - 값의 스케일을 변경해 입력을 정규화
  - 신경망을 보다 빠르고 안정적으로 만드는 방법





# Python을 활용한 이미지 분석

## 6. 합성곱 신경망 성능개선

“이번 시간을 모두 마치셨습니다.  
수고하셨습니다.”