

Python AI 프로젝트 - 얼굴 감정 인식 모델

작성자: 권수연

1. 데이터 전처리

필요하다고 생각되는 전처리 과정 순서대로 나열하면 아래와 같음

- 1) Dataset의 이미지파일 불러오기(구글드라이브)
- 2) 48*48 size, grayscale로 바꾸고 np.reshape() 통해 array로 변경

감정마다 반복 진행하고 list로 label과 함께 저장

(kaggle에 Data정보가 부족해 알 수 없지만 현재로서는 모든 이미지가 이미 48*48인 것으로 보임)
- 3) label은 one hot encoding 진행

1-1. import

```
import pandas as pd
import tensorflow as tf
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt
import cv2
import xml.etree.ElementTree as et
```

1-2. image file 가져오기 (구글드라이브 이용)

```
list_all = [['image', 'emotion']]

try:
    for i in range(0,300):

        image_angry = cv2.imread(path_angry + str(i) + ".jpg")

        try:
            image = np.mean(image_angry, axis=2)
            image = image.astype(np.float32) / 255
            image = np.reshape(image, (1, 48, 48))

            list_all.append([image, 'angry'])
        except:
            pass

except FileNotFoundError:
    pass

len(list_all)
```

- * 한계 : 구글 드라이브의 내 드라이브 폴더에서만 파일 불러오기가 가능한 것으로 보임
 - * 이미지를 for문으로 반복해서 가져오고 array로 바꾼 다음 리스트에 추가함 (try except로 예외처리)
- 0.jpg 부터 300.jpg 사이에 실제 존재하는 이미지 파일은 38개 -> list에 성공적으로 저장됨

1-3. RGB to Gray & np.reshape() (2가지 방법 모두 가능)

```
# 흑백으로 바꾸고 255로 나누는 방법 1
from skimage import color
from skimage import io

imgGray = color.rgb2gray(image_angry)

imgGray.shape

imgGray = np.reshape(imgGray, (1, 48, 48))

imgGray

array([[[[0.28235294, 0.30588235, 0.31764706, ..., 0.20392157,
          0.16862745, 0.15686275],
         [0.25490196, 0.24313725, 0.23137255, ..., 0.21568627,
          0.20392157, 0.17647059],
         [0.18431373, 0.16470588, 0.23529412, ..., 0.18431373,
          0.21176471, 0.19215686],
         ...,
         [0.35686275, 0.25490196, 0.18039216, ..., 0.29019608,
          0.21176471, 0.16862745],
         [0.29803922, 0.31372549, 0.29803922, ..., 0.42352941,
          0.27058824, 0.18039216],
         [0.30196078, 0.29019608, 0.31764706, ..., 0.41176471,
          0.42352941, 0.32156863]]]])
```

```
# 흑백으로 바꾸고 255로 나누는 방법 2(수업시간)
try:
    image = np.mean(image_angry, axis=2)
except:
    pass

image = image.astype(np.float32) / 255
image = np.reshape(image, (1, 48, 48))
image

array([[[[0.28235295, 0.30588236, 0.31764707, ..., 0.20392157,
          0.16862746, 0.15686275],
         [0.25490198, 0.24313726, 0.23137255, ..., 0.21568628,
          0.20392157, 0.1764706 ],
         [0.18431373, 0.16470589, 0.23529412, ..., 0.18431373,
          0.21176471, 0.19215687],
         ...,
         [0.35686275, 0.25490198, 0.18039216, ..., 0.2901961 ,
          0.21176471, 0.16862746],
         [0.29803923, 0.3137255 , 0.29803923, ..., 0.42352942,
          0.27058825, 0.18039216],
         [0.3019608 , 0.2901961 , 0.31764707, ..., 0.4117647 ,
          0.42352942, 0.32156864]]], dtype=float32)
```

1-4. Label One Hot Encoding

angry, happy, fear...로 저장했던 라벨을 모두 0-6으로 one-hot encoding 진행

아래와 같은 코드로 간단히 가능

```
# one-hot encoding
train_labels = keras.utils.to_categorical(train_labels, 7)
test_labels = keras.utils.to_categorical(test_labels, 7)
```

2. 얼굴 감정 인식 모델

- CNN등 이미지&영상 분석에 사용되는 인공신경망을 사용하여 모델 쌓기 - compile - 훈련
(이미지 처리 쪽으로 강의 더 하신다고 하니, 모델은 미리 만들고 수업 내용 기반으로 계속 보강하는 방향으로?)
- 오버 피팅 방지를 위한 코드 필수
- 성능 개선: Dropout 등

3. 마스크 쓴 감정인식 모델로 발전

이미지 파일을 보면, 대부분 아래 절반을 잘라냈을 때 눈과 코 일부만 남게 됨

-> 데이터 전처리를 이미지 아래 절반만 잘라내는 방식으로 다시 진행하고 동일한 모델로(입력 레이어 등 일부만 수정) 훈련

4. 참고하면 좋을 자료들

<https://www.youtube.com/c/%ED%87%B4%EA%B7%BC%ED%9B%84%EB%94%B4%EC%A7%93>

<https://blog.naver.com/kiakass>