

# 3차시

주피터에 코드 누적

## OX\_Classification

### MLP 분류

- OX\_데이터셋 불러오기
  - 현재경로 - /OX\_Label/O or X
  - os.path.join 활용
  - 사이즈 줄이고 넘파이 배열에 저장

```
from tensorflow.keras.preprocessing.image import load_img, img_to_array

import os
import numpy as np

path = 'OX_Label/'

resized_area = (100, 100)

x_data, y_data = [], []

for label_name in ["O", "X"]:
    label_dir = os.path.join(path, label_name)
    label = label_name

    for fname in os.listdir(label_dir):
        fpath = os.path.join(label_dir, fname)
        print(fpath)
        img = load_img(fpath, target_size = resized_area, color_mode = 'grayscale')
        arr = img_to_array(img).flatten()
        x_data.append(arr)
        y_data.append(label)

x, y = np.array(x_data), np.array(y_data)
```

- path 에 지정한 폴더에서 O X 폴더 이름을 라벨 이름으로 구분해 처리하기 위해 for문 사용
  - 각 O나 X 20개 파일 에 대해 fpath 를 os.path.join 으로 지정
  - x, y 저장
- OX\_이미지 전처리
  - load\_img 모듈을 활용해 색을 gray 로 바꾸고 300\*300 → 100\*100 으로 이미지 resize
  - img\_to\_array 모듈을 활용해 이미지를 flatten 시킨다. (모델이 MLP이므로)
- MLP로 진행시 k-fold 적용, 미적용 각각 정확도 산출

### activation function

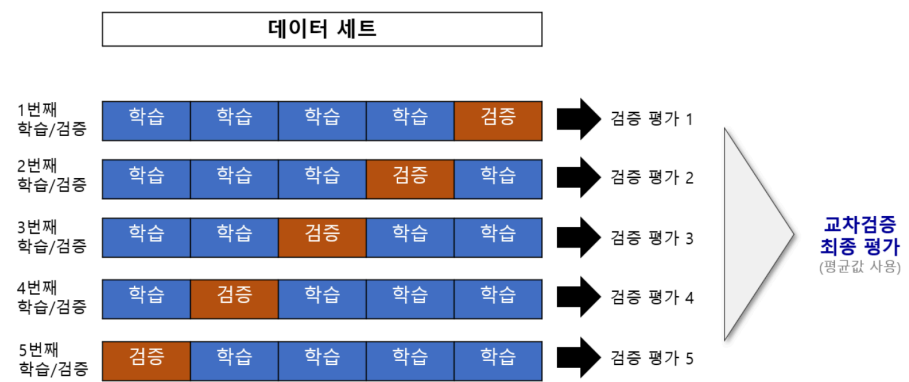
- 목적
  - 입력값을 변화시켜 출력하는 것으로(입력→은닉층→출력층 간) 사용.
  - 선형함수는 아무리 합성시켜도 선형함수이므로 다중 퍼셉트론의 의미가 사라짐
  - 주로 비선형 함수로 활성화 함수를 사용해 올바르게 분류할 수 있게 값을 변형하는 역할
- 종류, 특징
  - 입력의 총합(weighted sum)을 출력으로 넘겨줄 때 중간다리 역할
  - 이진 활성화 함수(Binary Step)
    - 계단 함수 - 0또는 1을 출력한다.( $x < 0$ 이면  $y=0$ ,  $x \geq 0$ 이면  $y=1$ )
  - 선형 활성화 함수(Linear)
    - XOR 분류를 할 수 없음
    - Hidden Layer 없이 구성된 네트워크로 구성해도 같은 기능 구현가능

- 선형 활성화 함수를 여러번 적용해도 선형이기 때문
  - 비선형 활성화 함수(Non-linear)
    - sigmoid, tanh, relu, softmax
- sigmoid
  - S자 모양의 쌍곡선함수
  - 기울기 소실 문제(Gradient Vanishing) 발생
  - 미분시 0~1사이의 값
  - (0, 1/2)에 대칭
  - 출력층에 권장
  - 분류: 이진분류, 특정 클래스에 속할 확률(미분값이 0~1이므로)
- tanh - hyperbolic tangent
  - 시그모이드 보다 부드러운 쌍곡선 형태의 tan 함수
  - 시그모이드 보다 빠른 수렴
  - 미분시 -1~1 사이의 값
  - 원점 대칭
  - 기울기 소실/폭발 문제 발생(  $\tanh$  미분 시  $\tanh^2$ 이 등장)
  - 회귀, 다중 클래스 분류(은닉층)
- relu
  - $x \geq 0$ 일때 x 출력, 이외 0 출력
  - 기울기가 양수일 때 1, 음수일 땐 0
  - 기울기 소실 문제 완화
  - 계산이 효율적
  - 음수일때 0이므로 뉴런이 죽을 수 있음(Dead Neuron)
  - $x < 0$ 일때 기울기를 약하게 감소시키는 Leaky ReLU가 있음
- softmax
  - 출력층에서 주로 사용(확률화)
  - 출력범위: 0~1
  - cross-entropy와 같이 사용시 미분 계산이 깔끔하게 떨어짐
  - 많은 데이터를 가진 클래스에 편향될 수 있음
  - 분류: 다중 클래스 분류
  - 이진분류 부적합

## 최적화 함수란?

- 목적
  - 학습 과정에서 에러를 줄이는 방법으로 이용한다.
  - 손실함수에서 최적의 가중치 값을 찾는 것이 목적이다.
  - 미분 가능한 함수의 기울기를 이용한다.
- 종류, 특징
  - GDM(Gradient Descent Method) - 기울기 경사하강법
    - 에러공간에서 기울기를 따라 내려가면서 가장 낮은(최적) 기울기를 찾아가는 방식
      - SGD
  - Adagrad(Adaptive Gradient) - 적응형 기울기
    - 학습률을 각 Parameter마다 다르게 조절 - 자주 등장 → 학습률 ↓, 적게 등장 → 학습률 ↑
    - (기울기의 미분값)<sup>2</sup>을 누적시킨다. → 학습률이 점차 0에 수렴
  - RMSprop(Root Mean Squared propagation) - 루트 평균 전파법
    - Adagrad에서 기울기 변화를 시킬 때 누적된 변화량과 최근 변화량을 비교하여 최근의 것을 더 반영하는 방식
    - (기울기의 미분값)<sup>2</sup>에 루트를 씌운 값을 가중치로 전파시킨다.
    - 학습 초기에 불안정한 가중치 변화
  - Adam(Adaptive Moment Estimation) - 모멘텀 적응 예측
    - Adagrad와 RMSprop을 같이 사용하여 각각의 단점을 보완
    - 학습 초기에 불안정한 가중치 변화 문제를 보완

k-fold



- k-fold란?
  - 가장 보편적으로 사용되는 교차검증 기법 - 사이킷런 API
  - K개로 분할된 데이터 폴드셋을 통해 K번 검증하여 평균을 내는 방법
    - Train Set을 k개의 fold로 분할한다. (주로 5, 10)
      - 폴드 하나를 검증 데이터로, 나머지를 Train Set으로 사용해 훈련/평가를 진행
      - 다음 폴드를 검증 데이터, 나머지를 Train Set으로 변경하여 k번째까지 진행한다.
- k-fold 원리
  - 같은 데이터 셋에 대해 다른 학습 데이터 / 검증 데이터를 사용해 교차로 검증하므로 검증 데이터에 편향되어 과적합하는 현상을 방지시킬 수 있다.
  - k번 진행하므로 k개의 평가값이 나오기 때문에, 전체적인 모델 성능은 평균값을 사용하여 평가한다.