

2nd Week Homework**Due date : 07.24**

2주차 Study에서는 인공 신경망의 기원이 되는 퍼셉트론에 대해 먼저 배웠습니다. 퍼셉트론을 이용하여 간단한 논리 회로를 구현하였습니다. 또한, 인공 신경망의 전체적인 구조와 여러 가지 활성화 함수에 대해 알아보았고, 마지막으로 신경망에서 여러 개의 데이터가 들어왔을 때 연산 처리를 어떻게 하는 지 배웠습니다.

이제부터는 과제가 나가지만, 꼭 유의해야 할 점이 있습니다. Chapter 2부터 Chapter 6까지 전체적인 흐름을 봤을 때, Chapter 2에서 썼던 코드가 Chapter 3, 4, 5, 6에서 재활용이 됩니다. 이렇게 앞 Chapter에서 구현한 코드는 뒤 챕터에서 쓰이기 때문에 과제를 할 때마다 가이드라인에 맞게 저장을 해주셔야 합니다.

Q1. [20 points] [실습 환경 세팅]

1) 'Deep Learning'이라는 새로운 폴더를 적절한 경로에 만들어 주세요. 되도록이면, 주피터 노트북에서 바로 접근할 수 있도록 하는 것이 좋습니다.

2) 생성한 'Deep Learning' 폴더 안에 'common'이라는 이름의 폴더와 'dataset'이라는 이름의 폴더를 각각 만들어 주세요.

여기까지 했을 때, 상황을 아래와 같습니다. (`__pycache__`는 없어도 됩니다.)

Deep Learning				
이름	수정된 날짜	유형	크기	
<code>__pycache__</code>	2023-07-03 오전 1:11	파일 폴더		
<code>common</code>	2023-07-09 오후 6:48	파일 폴더		
<code>dataset</code>	2023-07-03 오전 1:41	파일 폴더		

Q2. [40 points] AND, OR, XOR, NAND 실습하기

이제부터 주피터 노트북의 ipynb 파일을 하나 만들고 실습을 시작하시면 됩니다.

1) 현재 오픈하신 ipynb 파일에 AND, NAND, OR을 구현하는 3가지 함수를 만들고, 해당 함수의 원형을 참고하여 만들어 주세요. (단, 수업시간에 배운 가중치와는 다른 가중치를 이용하세요)

Ex. AND의 경우에는 $(w_1, w_2, b) = 0.5, 0.5, -0.7$ 이었는데, 이를 제외한 다른 가중치 사용하기

`def AND(x1, x2), def NAND(x1, x2), def OR(x1, x2)`

2) 1) 단계에서 AND, NAND, OR 함수를 완성시켰다면, 아래와 같은 코드를 실행하여서 적절한 답이 나오는 지 확인하세요.

```
print('AND 게이트 결과 : ', AND(0, 0), AND(0, 1), AND(1, 0), AND(1, 1))
print('NAND 게이트 결과 : ', NAND(0, 0), NAND(0, 1), NAND(1, 0), NAND(1, 1))
print('OR 게이트 결과 : ', OR(0, 0), OR(0, 1), OR(1, 0), OR(1, 1))
```

3) XOR 게이트의 한계점을 주석으로 달아서 작성해주시고, XOR 게이트를 a에서 구현하였던 AND, NAND, OR을 적절하게 이용하여 구현하세요. 함수 원형은 아래와 같습니다.

def XOR(x1, x2)

4) XOR 게이트를 c에서와는 다르게, 아래와 같이 두 가지 선형 분류기를 이용하는 것처럼 구현하세요.

즉, AND, NAND, OR 함수에서 구현했던 것처럼 적절한 조건을 이용하여 구현하세요.

Q3. [40 points] 활성화 함수 구현하기

1) 스터디 시간에 배웠던 step_function, sigmoid, ReLU 함수를 해당 함수의 원형을 참고하여 만들어 주세요.

def step_function(x), def sigmoid(x), def relu(x)

2) 항등함수는 $f(x)=x$ 인 함수입니다. 이 함수를 아래 함수의 원형을 참고하여 만들어 주세요.

def identify_function(x)

3) 하이퍼볼릭 탄젠트 함수를 아래 함수 원형을 참고하여 만들어 주세요.

def hyperbolic_tanh(x)

4) 활성화 함수가 아래와 같을 때, 이를 구현하는 코드를 아래 함수 원형을 참고하여 만들어 주세요.

(단, Relu함수에서 처럼 np.maximum을 이용하세요)

def leaky_relu(x)

$$f(x) = \begin{cases} 0.01x & (x \leq 0) \\ x & (x > 0) \end{cases}$$

5) 마지막으로, Q1에서 만들었던 common 폴더에 'functions.py'를 만들어주세요. Ipython 파일이 아닌, py 파일이어야 합니다.)

6) 1)~4) 까지 만들었던 모든 함수를 functions.py에 copy-and-paste를 해주세요.

[결론]

Q2의 경우에는 앞으로 쓸일이 없기 때문에 그냥 임의로 ipynb 파일 하나 만들어 주시고, Q3은 앞으로 계속 쓰이기 때문에, ipynb에 하는 것도 중요하지만 common/functions.py에 무조건 있어야 합니다.

[제출 방법]

1. Q1과 관련된 사진
2. Q2와 관련한 ipynb파일 하나
3. Q3 완료 후의 functions.py 파일