인공지능수학 개론

(저자 김종락)

(임시적인) 목록

I. 선형대수와 인공지능
1. 파이썬 소개: 수식 중심으로
2. 선형대수 기초
2.1 실습
3. 행렬의 연산 및 역행렬
3.1 실습
4. 벡터와 공간, 행렬과 사상
4.1 실습
5. 선형변환, 고윳값, 고유벡터
5.1 실습

II. 미적분학과 인공지능
6. 미분과 적분
6.1 실습
7. 편미분과 경사 하강법
7.1 실습

III. 확률과 통계와 관련된 인공지능
8. 조건부 확률과 베이즈 정리
8.1 실습
9. 상관분석과 분산 분석
9.1 실습

IV. 머신러닝과 딥러닝와의 연계
10. 머신러닝 소개
10.1 실습
11. 딥러닝 소개
11.1 실습

Chapter 6. 미분과 적분

미분과 적분은 고등학교 때부터 배웠기 때문에 친숙한 수학으로 느껴질 수도 있다. 실제로는 미분의 개념은 중학교 때 배웠던 직선의 방정식에서 찾을 수 있다. 함수 $y=f(x)$가 구간 $[a-\delta, b+\delta]\,(\delta>0)$에서 정의되어 있을 때 두 점 $(a,f(a)),(b,f(b))$을 지나는 직선의 기울기는 $m=\frac{f(b)-f(a)}{b-a}$이 되고 이 직선의 방정식은 $y-f(a)=m(x-a)$이 된다. 만일 x가 a에 가까워 질 때 $\frac{f(x)-f(a)}{x-a}$이 어떤 값에 수렴할 때 그 값을 $f'(a)$라고 한다. 따라서 $f'(a)$는 직선 $y=f(x)$위에 있는 $(a,f(a))$에 접하는 접선의 기울기가 된다. 물리학적으로는 $t=a$라는 시간의 **순간속도**를 의미한다고 볼 수 있다. 예를들어, 한시간동안 30km를 운전하는 차의 평균속도는 30km/hr이지만, 커브길을 갈 경우 순간속도는 증가한다. 그 이유는 $t=t_1, t=t_2$의 시간차가 작을 경우 t_2-t_1이 작아지므로 그에 따른 기울기가 증가하기 때문이다. 따라서 미분은 중학생 조차에게도 가르칠 수 있는 주제라고 생각한다!

이제 적분으로 이야기를 돌려보자. 적분은 미분보다 더 어려운 것으로 이해된다. 혹은 적분을 미분의 반대개념으로 이해되기도 한다. 이는 모두 일종의 편견에 해당한다. 적분의 개념은 **면적(area)**에서 시작되었다고 볼 수 있다. 면적은 초등학교 때부터 배우는 개념이므로 적분을 설명하는 것은 그리 어렵지 않다. 예를 들어 $y=f(x)$가 구간 $[a,b]$에서 정의되고 이 구간 위에서 함수값 $f(x)$가 양수일 때, 구간 $[a,b]$와 $y=f(x)$의 그래프 사이의 면적을 **적분**이라고 한다. 예를 들어 $y=x$라고 할 때 구간 $[0,1]$와 이 직선의 그래프로 이루어진 도형은 직각 삼각형이 된다. 이 삼각형의 면적은 $\frac{1}{2}$이 된다. 한편 $y=x$의 $[0,1]$위의 적분도 $\frac{1}{2}$이 된다. $y=x$의 $[0,1]$위의 평균은 1이고 미분값도 1이 된다. 이 두 개의 값 $\frac{1}{2}$과 1을 연결하는 식이 바로 '미적분학의 기본정리'(Fundamental Theorem of Calculus)이다. 즉, $\int_0^1 x\,dx=\frac{1}{2}x^2|_0^1=1/2.$

미분과 적분은 데이터 분석에 어떻게 연결될까? 일단 미분은 비용함수(cost function)를 최소로 하는데 사용이 된다. 그때 사용되는 방법을 편미분이라고 한다. 여러 개의 변수가 있는 함수를 하나의 변수에 관하여 미분하는 것을 편미분이라고 한다. 그냥 미분하고 근본적인 차이는 없다. 비용함수를 최소로 하기 위해 변수의 값을

조정하면서 비용함수를 최소화 시키는 방법을 경사하강법이라고 한다. 한편 확률 함수를 적분하거나 강화학습에서 등장하는 보상함수를 적분할 수도 있다.

이 장에서는 여러 개념들을 소개하고 서로 어떻게 연결되는지 살펴본다.

■ 미분

● (정의) $y=f(x)$가 실수 위에서 정의되고 $x=a$일 때
$\frac{df(x)}{dx}|_{x=a}=\lim_{h\to 0}\frac{f(a+h)-f(a)}{h}$ 이 존재할 때 이를 $f'(a)$라고 한다. 일반적으로 임의의 x에 대하여

$$f'(x)=\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$$

라고 하고 이를 $f(x)$의 미분(derivative)라고 하고 **도함수**라고도 하며 y'또는 $f'(x)$로 표현한다.

● (예제)
$y=x^2$라고 하자.
(1) 이 함수의 그래프를 그리시오

(2) $f'(x)$를 정의에 의하여 구하시오.

● (예제)
$y=e^x$라고 하자.
(1) 이 함수의 그래프를 그리시오.

(2) $f'(x)$를 정의에 의하여 구하시오.

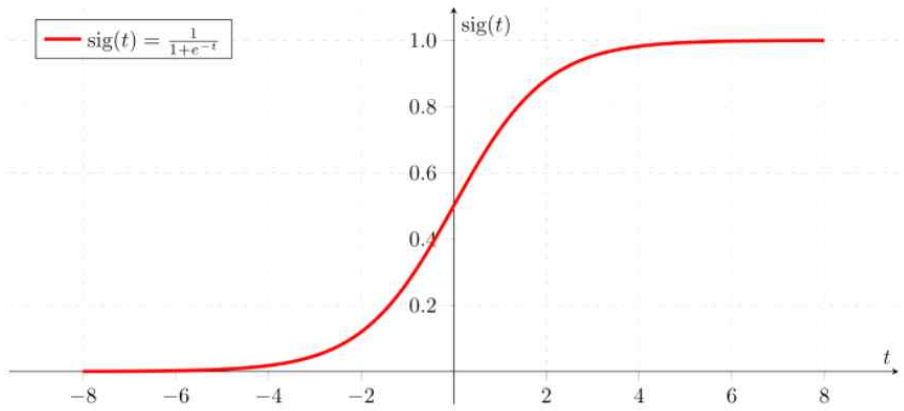
● (공식) $f(x), g(x)$가 미분가능하고 c,d를 상수라고 할 때
- $c'=0$
- $(x^n)'=nx^{n-1}\ \ (n\geq 1)$
- $(\ln x)'=x$
- $(e^x)'=e^x$
- $(cf)'=cf'$

- $(af \pm bg)' = af' \pm bg'$ (미분은 변환이다!)
- $(fg)' = f'g + fg'$ (곱셈 공식)
- $\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2}$ (나눗셈 공식)

● (예제)

$y = \sigma(x) = \frac{1}{1+e^{-x}}$라고 하자. 이것을 시그모이드sigmond 함수라고 한다. 딥러닝에서 자주 사용되는 미분 가능한 함수이다.

(1) 이 함수의 그래프는?



(2) 이 함수의 미분을 구하시오.

$$\sigma'(x) = \frac{e^{-x}}{(1+e^{-x})^2}$$

(3) $\sigma'(x) = \sigma(x)(1-\sigma(x))$를 만족함을 보이시오.

● (정리)(**미적분학의 기본정리**)

만일 함수 $f(x)$가 구간 $[a,b]$에서 연속이고 개구간 (a,b)에서 미분가능하다고 하자. f의 역도함수(anti-derivative)를 $F(x)$라고 할 때

(1) $F'(x) = f(x)$를 만족하고

(2) $\int_a^b f(x)dx = F(b) - F(a)$

가 성립한다.

(증명) $F(x) = \int_a^x f(t)dt$라고 하면

$F(b) - F(a) = \int_a^b f(t)\,dt - \int_a^a f(t)\,dt = \int_a^b f(t)dt$ 이므로 (2)가 성립한다.

또한

$$F'(x) = \lim_{h \to o} \frac{F(x+h) - F(x)}{h} = \lim_{h \to 0} \frac{\int_a^{x+h} f(t)dt - \int_a^x f(t)dt}{h}$$

$$= \lim_{h \to 0} \frac{\int_x^{x+h} f(t)dt}{h} = f(x)$$ (적분의 중간값 정리)

적분의 중간값 정리란 '$[a,b]$에서 정의된 연속함수 $f(x)$에 대하여 $\int_a^b f(x)\,dx = f(c)(b-a)$을 만족하는 $c \in (a,b)$가 존재한다.' 기하학적 의미는 $y = f(x)$의 그래프의 면적을 높이가 $f(c)$이고 밑변이 $b-a$인 정사각형의 면적과 동일하게 할 수 있는 $c \in (a,b)$가 존재한다는 것이다.

따라서 (1)이 성립한다.

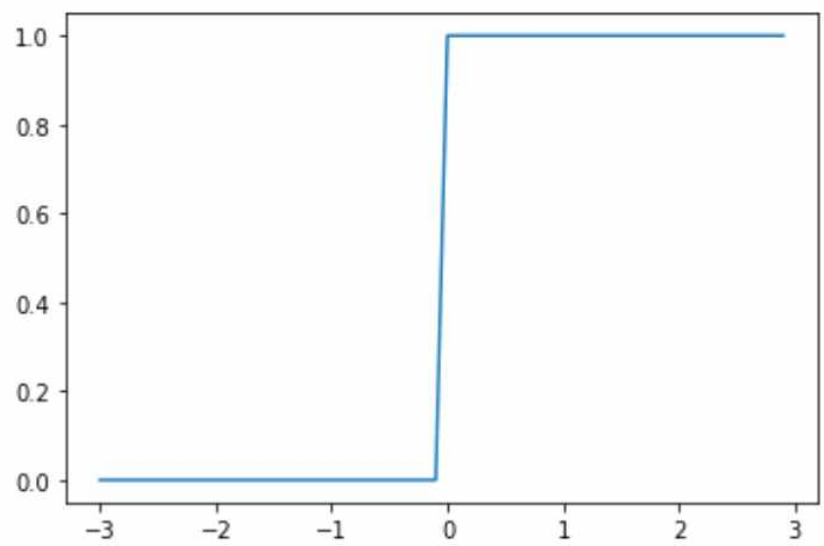
■ (파이썬 실습)

● 4개의 활성함수를 파이썬으로 표현해보자.

```
### Step function
import numpy as np
import matplotlib.pyplot as plt

def h(x):
    return np.array(x>0) # x값이 0보다 크면 True=1

x = np.arange(-3, 3, 0.1)
y = h(x)
plt.plot(x,y)
plt.show
```



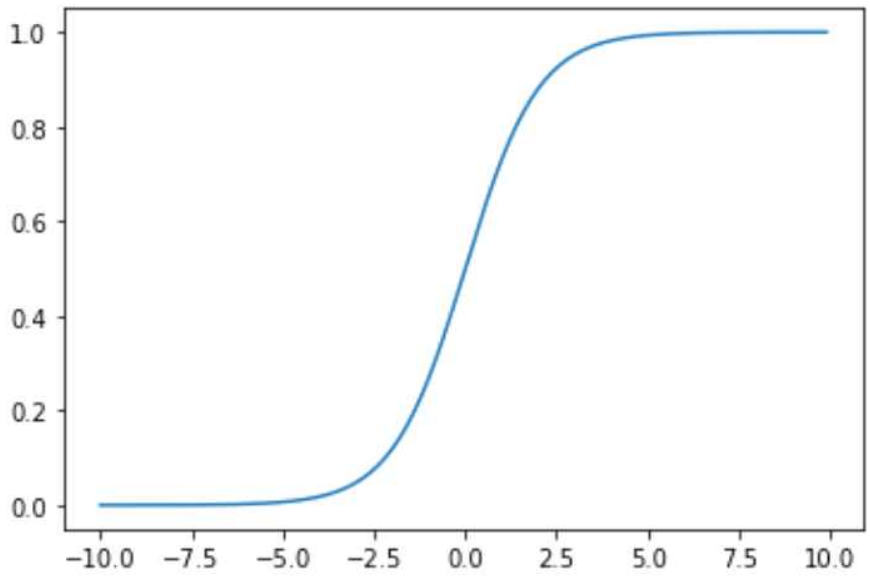
```python
#sigmoid function

import numpy as np
import matplotlib.pyplot as plt

def sigmoid(x):
    return 1/(1 + np.exp(-x))

x = np.arange(-10, 10, 0.1)
y = sigmoid(x)

plt.plot(x,y)
plt.show
```



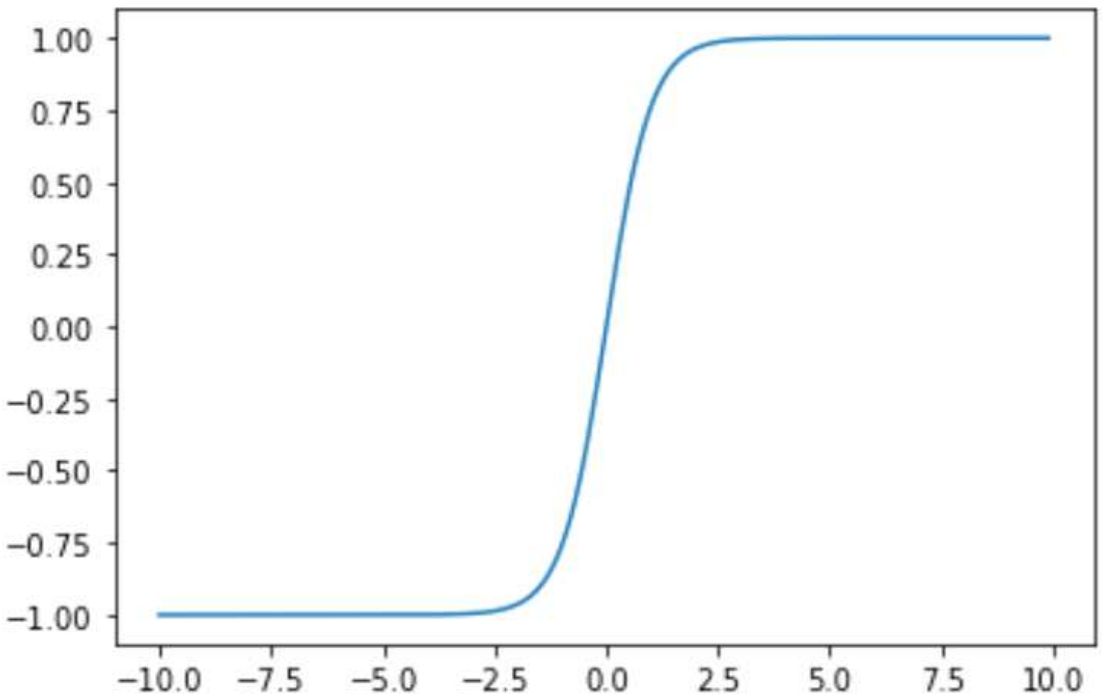
```python
### hyperbolic tangent

import numpy as np
import matplotlib.pyplot as plt

def htan(x):
    return np.tanh(x)

x = np.arange(-10, 10, 0.1)
y = htan(x)

plt.plot(x,y)
plt.show
```



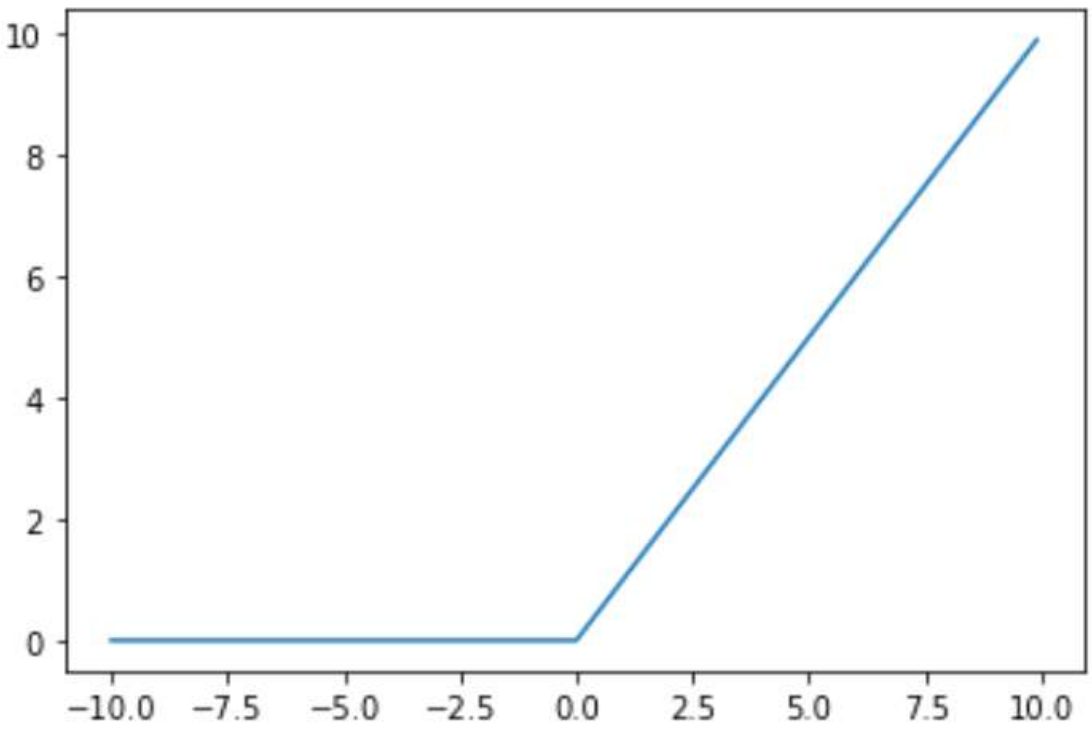
```
### ReLu


import numpy as np
import matplotlib.pyplot as plt

def relu(x):
    return np.maximum(0,x)

x = np.arange(-10, 10, 0.1)
y = relu(x)

plt.plot(x,y)
plt.show
```



```
###  함수 정하기
from scipy.misc import derivative

def f(x):
    return 3* x**2 + x +1
print(f(x))
```

```
3*x**2 + x + 1
```

```
### 도함수 구하기
def d(x):
    return derivative(f,x)
print(d(x))
```

```
-1.5*(x - 1.0)**2 + 1.5*(x + 1.0)**2 + 1.0
```

```
### 도함수 구하기(derivative)
import sympy as sp
x = sp.Symbol('x')
print(sp.diff(3* x**2 + x +1, x)) # 3x^2 +x +1
```

```
6*x + 1
```

```
derivative(f,x)
```

$$-1.5(x-1.0)^2 + 1.5(x+1.0)^2 + 1.0$$

```
### evaluate f at x=2.0
print(d(2.0))
```

```
13.0
```

● (편미분) $f(x,y,z)$를 3개의 변수가 있는 실수함수라고 할 때, f의 그래디언트 벡터는 $\nabla f = grad(f) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial x}\right)$로 정의된다. $\frac{\partial f}{\partial x}$는 x를 변수로 보고 나머지는 상수

```
### 정적분 구하기
import numpy as np
from scipy.integrate import quad

funct =lambda x: x  # 함수 y=x 정의, lambda + 변수를 사용함
solution = quad(funct, 0, 1) # 0부터 1까지 y=x 적분을 의미
print(solution)  # 0.5는 적분, 두번째 값은 오차
```

```
(0.5, 5.551115123125783e-15)
```

로 본 후 미분하여 얻어진 함수이다. $\frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$도 마찬가지이다. 이들을 편미분이라고 한다.

● (예제) $f(x,y,z)=x^2+xy+z^2$라고 할 때

$\frac{\partial f}{\partial x}=$

$\frac{\partial f}{\partial y}=$

$\frac{\partial f}{\partial z}=$

$\nabla f=$

■ (파이썬 실습)

```
### 편미분 프로그래밍

h= 1e-5

def funct3(x,y,z):
    return x**2 +x*y + z**2

def grad(f, x,y,z):
    df_dx = (f(x+h, y, z) -f(x-h,y,z))/(2*h)
    df_dy = (f(x, y+h, z) -f(x,y-h,z))/(2*h)
    df_dz = (f(x, y, z+h) -f(x,y,z-h))/(2*h)
    return [df_dx, df_dy, df_dz]

grad(funct3, 0, 1, 2)  # grad of function at point (0,1,2)
```

```
[0.9999999999843466, 0.0, 4.000000000026205]
```