동양미래대학교

인공신경망

인공지능을 위한 기초 수학

DMUAi 동양미래대학교 인공지능소프트웨어학과

김유두 교수
(kimyudoo@dongyang.ac.kr)

인공지능 이해를 위한 기초 수학

기초 수학을 왜 알아봐야 할까요?

- 기본적인 인공지능(~딥러닝)의 동작이 수학적으로 어떠한 원리로 진행되는지 확인
- 지금까지 배웠던 문제 풀이를 위한 공식, 정답 찾기 방식의 수학이 아닌, 실제 원리를 파악하여 활용 방법을 이해하자!
- 수학 원리를 통해 알아볼 내용
  - 입력 값의 패턴 분석
  - 값을 어떻게 분석 하는가?
  - 이러한 분석의 수학적인 배경와 함수

기초 수학을 왜 알아봐야 할까요?

- 아주 어렵지 않음! -> 기본적인 원리를 이해합시다.
- 원리를 이해하기 위한 수학 내용은 기초 수학 수준으로 학습해도 됨
  - 왜냐면? 복잡한 계산은 컴퓨터가 해줌!
- 가장 필요할 것 같은 내용 위주의 수학 부분만 알아봄
- 수학을 계산하는 것에 집중하는 것이 아닌, 원리를 잘 알아봅시다!

일차함수, 기울기와 y절편

함수

- 다시 한번 알아봅시다. 수학의 함수?
  - 두 집합 사이의 관계
  - x,y 변수 (보통 x는 입력, y는 출력)
  - y = f(x) 로 표시

일차함수, 기울기와 y절편

일차 함수

- 일차 함수는 y가 x에 대해 일차식으로 표현
- 기본 형식

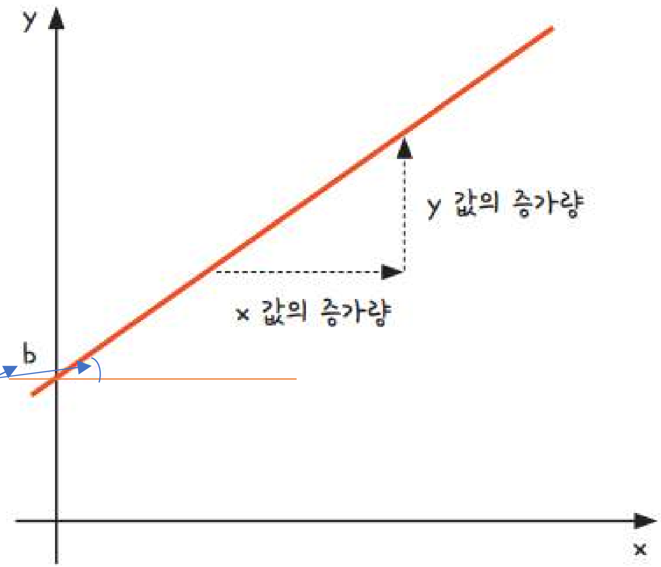
$$y = ax + b \ (a \neq 0)$$

- X가 일차인 형태, x가 일차로 남기 위해 a는 0이 아님

일차함수, 기울기와 y절편

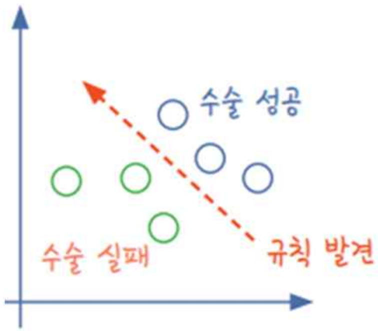
일차함수

- 식 y=ax + b 에서
  - a는 기울기
  - b는 절편
- 기울기
  - 기울어진 정도
  - x의 값 증가에 따른 y값 확인
  - 이러한 결과에 의해 기울기 a 결정
- 절편은 그래프가 축과 만나는 지점



일차함수, 기울기와 y절편

- 그럼 인공지능에서 무엇을 할 까요?
- 기존의 프로그래밍 방식
  - a와 b를 사람이 정하고 식을 넣음
- 인공지능
  - x(입력데이터)가 주어지고, y값이 있음
  - 이 때에 값을 통해 a와 b를 찾아줌

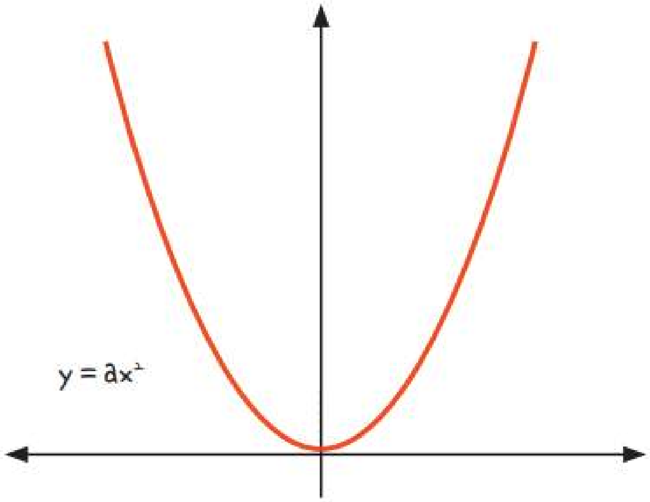


이차 함수와 최소값

이차함수

- 이차함수는?
  - y가 x에 관한 이차식으로 표현
  - 다음과 같은 식으로 표현

    $y = ax^2\ (a \neq 0)$
  - 쉽게 표현하자면?
    - 일차는 => 직선
    - 이차는 => 포물선
    - a > 0 아래로 볼록한 그래프

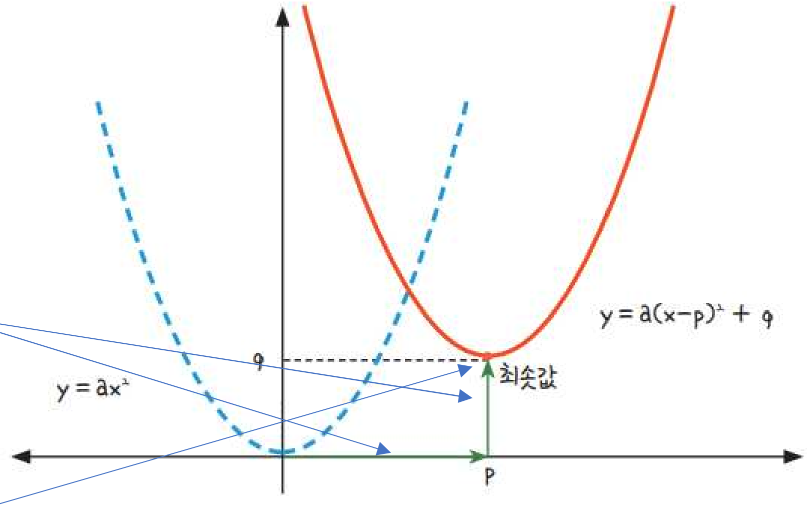


이차 함수와 최소값

이차함수와 최소값

- $y = ax^2$ 의 그래프를
  - x축 방향으로 p 만큼
  - y축 방향으로 q 만큼 평행이동

- 인공지능 에서는?
  - 포물선의 맨 아래 꼭짓점을 찾아야 함
  - 포물선의 맨 아래가 최소값
    - 이 값을 찾는 것이 인공지능이 해야 할 일



이차함수와 최소값

- 최소값은 최소 제곱법 공식으로 알아낼 수 있음
- 인공지능을 진행하는 문제에서는 최소 제곱법을 활용하기 힘듬
  - 최소 제곱법 계산을 위한 필요 조건을 알 수 없음 (데이터 중심)

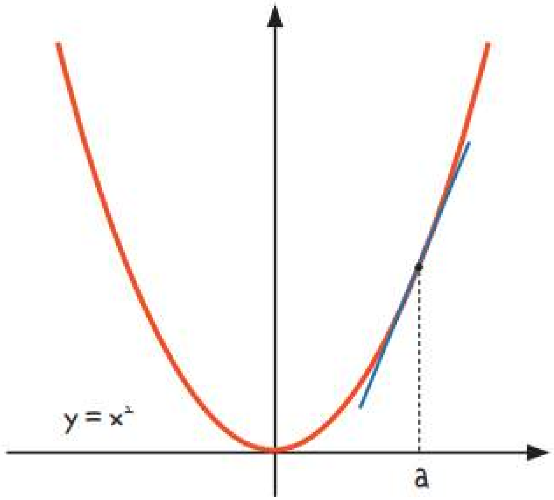
$$a = \frac{(x - x\text{ 평균})(y - y\text{ 평균})\text{의 합}}{(x - x\text{ 평균})^2\text{의 합}} \qquad \text{(식 4.1)}$$

  - 따라서, 미분과 기울기가 필요함

미분, 기울기

미분

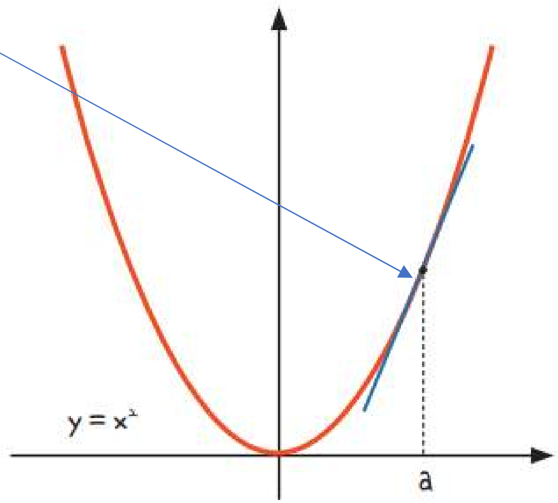
- a, b 값을 구하기 위해(이차함수) 포물선의 최소값 필요
- 최소값을 미분으로 구해야 함
- 미분과 기울기의 개념을 이해해야 함
- 그림 분석
  - $y=x^2$ 그래프
  - x축에 있는 한 점 a와 대응하는 y 값은 a^2
  - a가 오른쪽이나 왼쪽으로 조금씩 이동한다면?
    - y의 값도 조금씩 변화하지 않을까?



미분, 기울기

순간 변화율

- a가 미세하게 0에 가까울 만큼 움직임
  - y값 역시 매우 미세하게 변화
    - 너무 미세해서 실제 움직이는 것 같지 않고 방향만 아는 정도의 변화
    - 이 순간의 변화율이 순간 변화율
    - 방향성이 있으므로 직선을 그려주면 그래프와 맞닿는 접선이 그려짐
    - 이 선이 바로 기울기



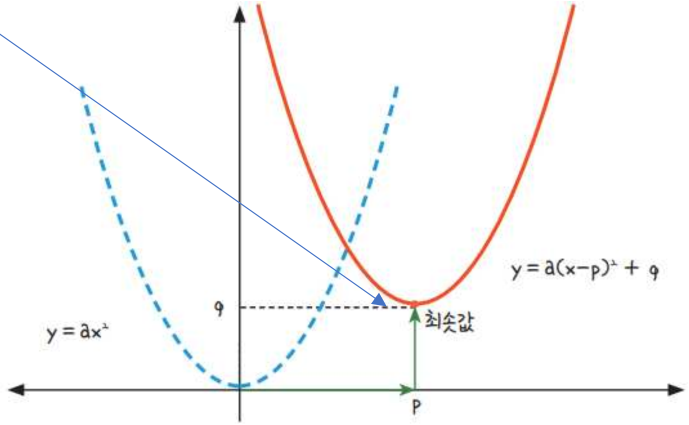
미분과 순간변화율, 기울기

- 미분을 한다는 것?
  - 순간 변화율 계산
  - 어느 순간에 변화가 일어나고 있는지 순자로 나타낸 것 => 미분 계수
    - 그래프에서 기울기를 의미함
  - 기울기가 0 일때 (x축과 평행한 직선으로 그려질 때)
    - 그래프에서 최소값의 지점이 됨

미분, 기울기

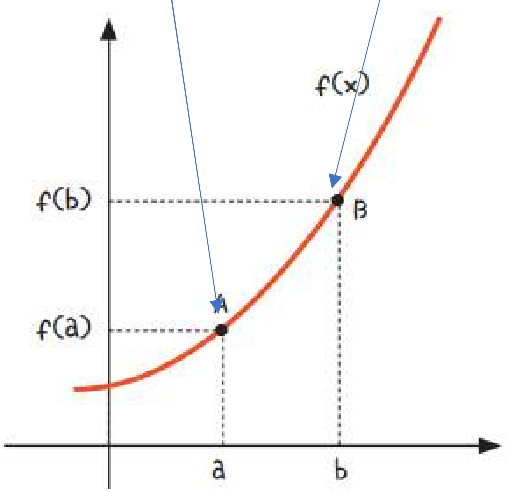
미분과 순간변화율, 기울기

- 미분을 한다는 것?
  - 순간 변화율 계산
  - 어느 순간에 변화가 일어나고 있는지 순자로 나타낸 것 => 미분 계수
    - 그래프에서 기울기를 의미함
  - 기울기가 0 일때 (x축과 평행한 직선으로 그려질 때)
    - 그래프에서 최소값의 지점이 됨



순간변화율 계산

- f(x)가 그림과 같이 주어짐
- x축 위의 두 실수 a와 b를 대입하면,
  - 두 점 A,B는 각각 A(a, f(a)), B(b, f(b))에 해당하는 곳에 표시



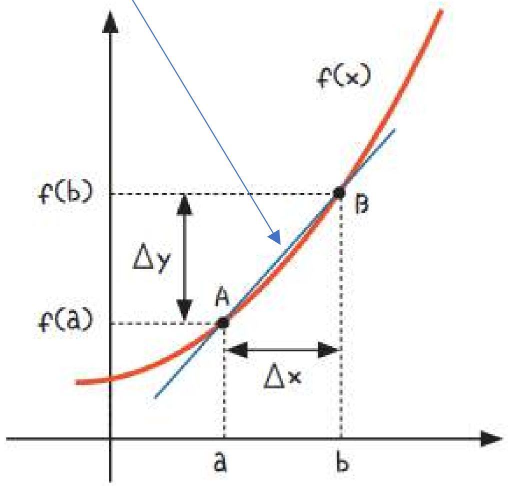
미분, 기울기

기울기 계산

- 두 점 A,B 를 이어 직선을 만들고 그 점을 지나는 직선 기울기가 그려짐
- Δ(델타)는 변화량을 나타내는 기호
- x값의 증가량 : b-a
- y값의 증가량 : f(b) – f(a)
- Δ를 통해 표현하면
  - x값의 증가량은 Δx
  - y값의 증가량은 f(a + Δy)
- 직선의 기울기 식은 다음과 같음

$$\frac{y \text{ 값의 증가량}}{x \text{ 값의 증가량}}$$

- A와 B를 지나는 직선의 기울기는?



$$\text{직선 AB의 기울기} = \frac{y \text{ 값의 증가량}}{x \text{ 값의 증가량}} = \frac{f(b)-f(a)}{b-a} = \frac{f(a+\Delta x)-f(a)}{\Delta x}$$

미분, 기울기

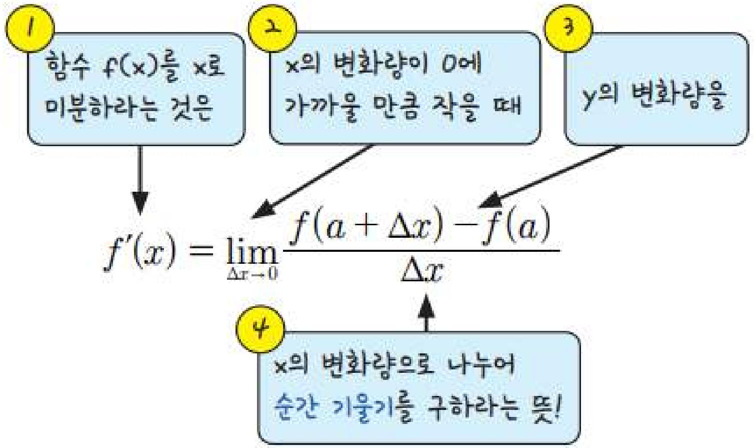
순간변화율 계산

- 이 전 페이지에서 직선 AB의 기울기를 A와 B 사이의 평균 변화율
- 미분에서 필요한건 순간 변화율
- 순간 변화율은 x의 증가량(Δx)이 0에 가까울 만큼 아주 조금 이동 했을때 순간적인 기울기
  - 따라서 극한(limit)기호를 사용해 나타냄

$$\lim_{\Delta x \to 0} \frac{f(a+\Delta x)-f(a)}{\Delta x}$$

미분 계산

- 미분의 계산 의미는 다음과 같음



미분, 기울기

미분 계산

- 미분의 기본 공식은 다음과 같음

미분의 기본 공식

1 | $f(x) = x$일 때 $f'(x) = 1$

2 | $f(x) = a$에서 a가 상수일 때 $f'(x) = 0$

3 | $f(x) = ax$에서 a가 상수일 때 $f'(x) = a$

4 | $f(x) = x^a$에서 a가 자연수일 때 $f'(x) = ax^{a-1}$

5 | $f(g(x))$에서 $f(x)$와 $g(x)$가 미분 가능할 때 $\{f(g(x))\}' = f'(g(x)) \times g'(x)$

편미분

- 미분과 비슷한 미분 하라는 의미
- 모든 변수를 미분하는 것이 아니라 원하는 한 가지 변수만 미분하고 그 외는 모두 상수로 취급하는 것 => 편미분
- $f(x) = x$와 같은 식은
  - 미분하라 = 편미분하라
  - 그 이유는 변수가 x 하나이기 때문

편미분

$f(x, y) = x^2 + yx + a$ (a는 상수)

- 이 식에서는 변수가 x, y 두개
- x에 대해서만 미분하고 싶다면, x에 관해 편미분 하라 하고 함
- 식은 아래와 같음

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

편미분

- $f(x, y) = x^2 + yx + a$를 x로 편미분 하라면?
  - 미분의 성질4에 따라 x^2 은 $2x$가 됨
  - 미분의 성질 3에 따라 yx는 y가 됨
  - 마지막 항 a는 미분의 성질 1에 따라 0 이됨
  - 정리하면

$$f(x, y) = x^2 + yx + a \text{일 때}$$

$$\frac{\partial f}{\partial x} = 2x + y$$

인공신경망

지수와 지수 함수

지수

- 지수는 다음의 형태

$$a^{\square}$$

- a를 밑, 윗부분을 지수
- a를 지수 만큼 반복해서 곱함
- 지수함수란 x가 지수 자리에 있는 경우
- 식으로 나타내면

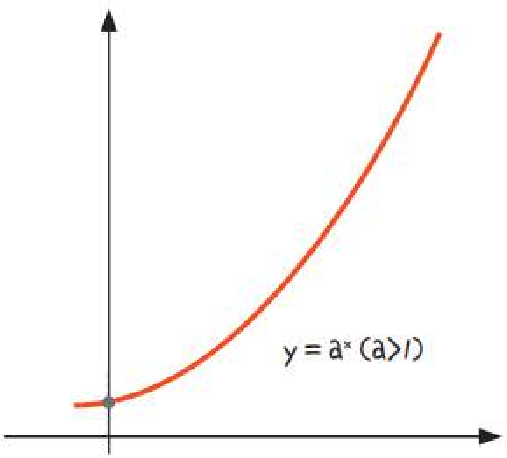
$$y = a^x \; (a \neq 1,\; a > 0)$$

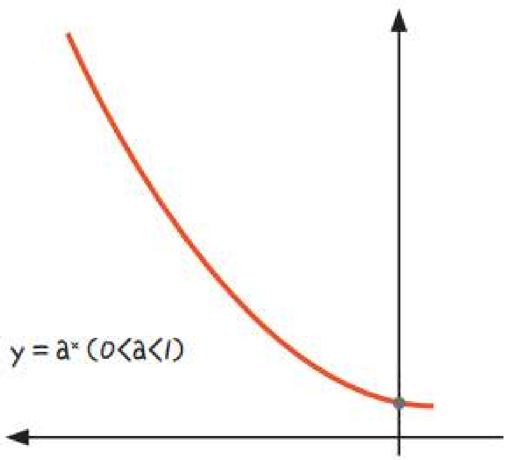
인공신경망

지수와 지수 함수

지수함수

- 밑(a)값이 중요
- 이 값이 1이면 함수가 아님
- 0보다 작으면 허수가 있어서 안됨
- 밑의 값은
  - $a > 1$ or $0 < a < 1$
- 그래프로 표현하면





시그모이드 함수

시그모이드

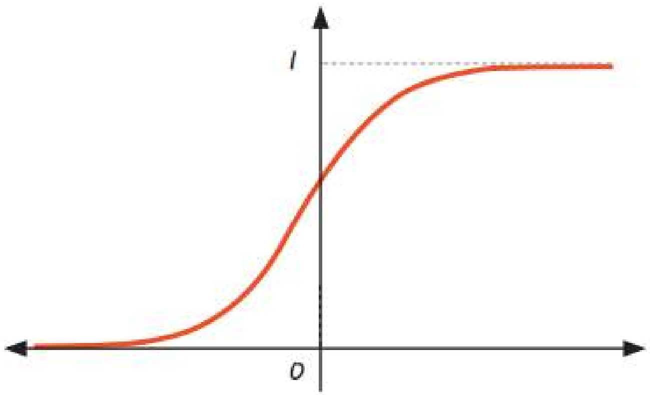
- 인공지능에서는 입력받은 신호를 얼마나 출력할지는 계산하는 과정이 반복됨
- 출력 값으로 얼마나 내보낼지를 계산 => 활성화 함수
- 활성화 함수의 대표작 => 시그모이드 함수
- 시그모이드 함수
  - 지수 함수에서 밑 값이 자연상수 e인 함수
  - 자연상수 e는 자연 로그의 밑, 오일러의 수 등으로 불림
  - 파이 π 처럼 중요하게 사용되는 무리수 그 값은 2.718281828...
- 자연 상수 e가 지수 함수에 포함되어 분모에 들어가면 시그모이드 함수가 됨

$$f(x) = \frac{1}{1+e^{-x}}$$

시그모이드 함수

시그모이드

- 그래프로 표현하면 다음과 같음



- x가 큰 값을 가지면 f(x)는 1에 가깝고
- x가 작은 값을 가지면 f(x)는 0에 가까움
- S자 형태로 그려지는 함수의 속성은 0 or 1 두 개의 값 중 하나를 고를 때 활용

로그와 로그 함수

로그

- a를 x만큼 거듭제곱한 값이 b라고 할 때, 이를 식으로 나타내면

  $a^x = b$

- a,b는 아닌데 x를 모른다면?
  - x를 구하기 위한 것이 로그
  - Logarithm 이라고 함
  - 지수에서 a와 b의 위치를 바꾸면 됨

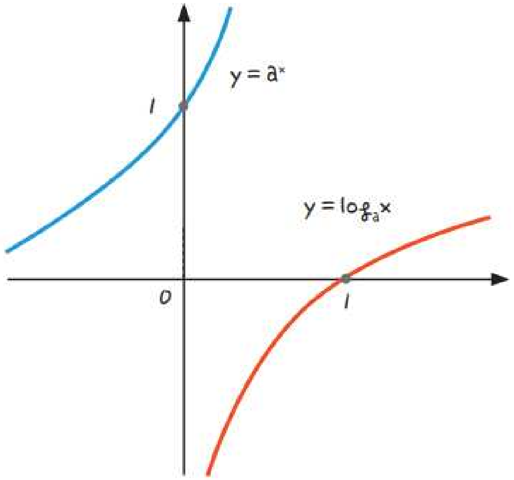
    $$a^x = b$$

    $$\log_a b = x$$

  - 로그와 지수는 역함수 관계
    - 역함수 : x와 y를 서로 바꾸어 가지는 함수

로그와 로그함수

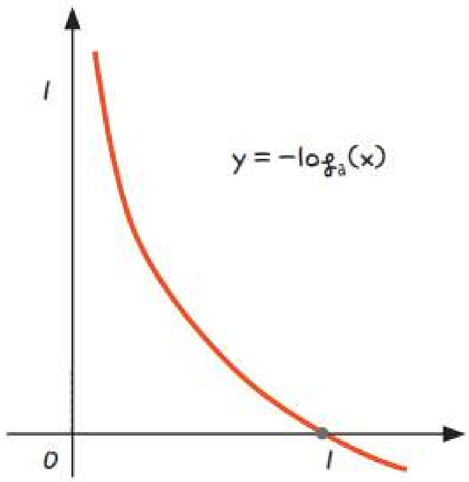
- 지수함수 $y = a^x(a \neq 1, a > 0)$ 는 로그 정의를 따라 다음과 같이 바꿈
  $x = \log_a y$
- 역함수를 만들기 위해 x와 y를 서로 바꿈
- 로그함수의 형태
  - 역함수의 그래프는 $y = x$에 대칭인 선으로 나타남
  - 그림은 지수 함수 $y = a^x$의 그래프를 $y = x$에 대칭으로 이동시킨 로그 함수 $y = \log_a x$의 그래프를 보여 줌



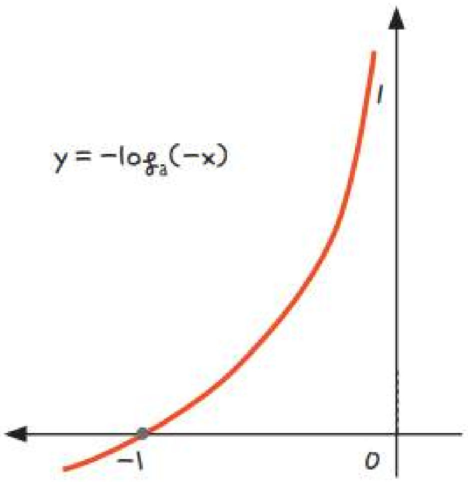
로그와 로그함수

- 로지스틱 회귀에서 x가 1에 가까워지거나 0에 가까워질수록 오차가 커지는 그래프가 필요함
  - 이를 위해 $y = \log_a x$를 x축 또는 y축으로 대칭 이동하거나 알맞게 평행 이동하면 다음과 같음

$y = -\log_a (x)$ 그래프

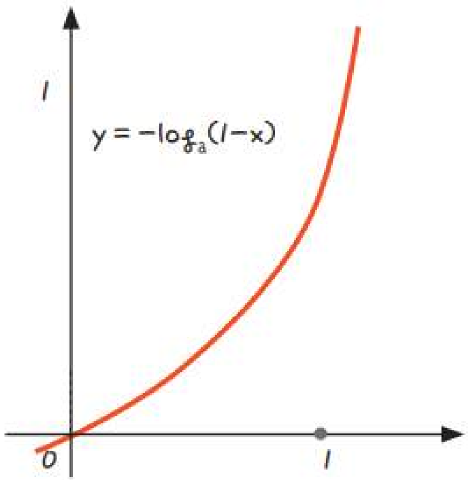


$y = -\log_a (-x)$ 그래프

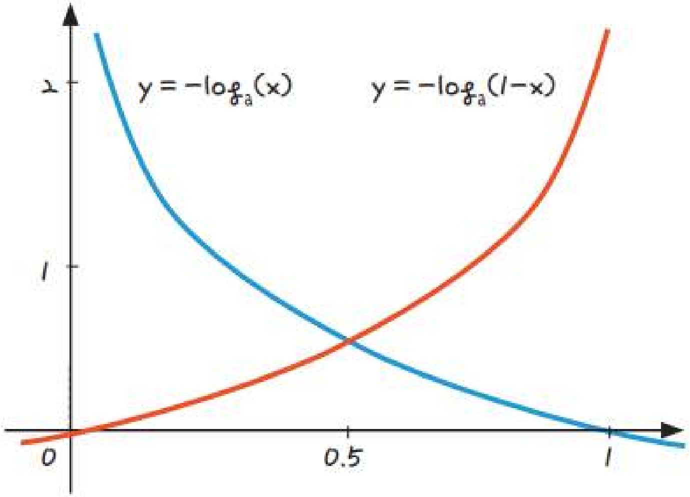


로그와 로그함수

$y = -\log_a(1-x)$ 그래프



$y = -\log_a(x)$와 $y = -\log_a(1-x)$ 그래프



시그마 기호

- 합을 표현하기 위해 만들어진 시그마 기호 $\sum$
- 1부터 n까지 F(i)에 대입해 더하라는 기호
- $\sum_{i=1}^{n} F(i)$ => $F(1)+F(2)+F(3)+\cdots+F(n)$

예측선 만들기

예측선

선형회귀

- 인공지능에서 가장 많이 쓰이는 기본이라면?
    - 선형회귀
    - 로지스틱 회귀

인공신경망

예측선

선형회귀

- 가장 좋은 예측선은
  - 선형회귀(Linear regression) 분석을 이용한 모델
  - 인공지능에서는 가장 좋은 선을 긋는 것이 중요
- 인공지능은 데이터가 들어오면 그것을 통해 규칙을 만들어야(컴퓨터가) 함
- 다양한 데이터에서 그 값에 따른 결과가 나오는 x,y 형태에서 선을 직선으로 그어보기 위해 활용하는것이 선형회귀
  - 독립적으로 변할 수 있는 값 x를 독립변수
  - 독립 변수에 따라 종속적으로 변하는 y를 종속 변수
  - 하나의 x 값만으로 y값을 설명할 수 있다면 단순 선형회귀
  - X값이 여러 개 있어야 한다면 다중 선형 회귀

가장 좋은 예측선 만들기

- 택시 이용 거리에 따른 요금만 있는 영수증의 데이터가 있음
- 이것을 활용하여 가장 좋은 예측선을 만들고 택시 요금 예측하는 인공지능 시스템을 만들어야 한다면?
- 데이터 분석

이동 거리	2km	4km	6km	8km
금액	8100원	9300원	9100원	9700원

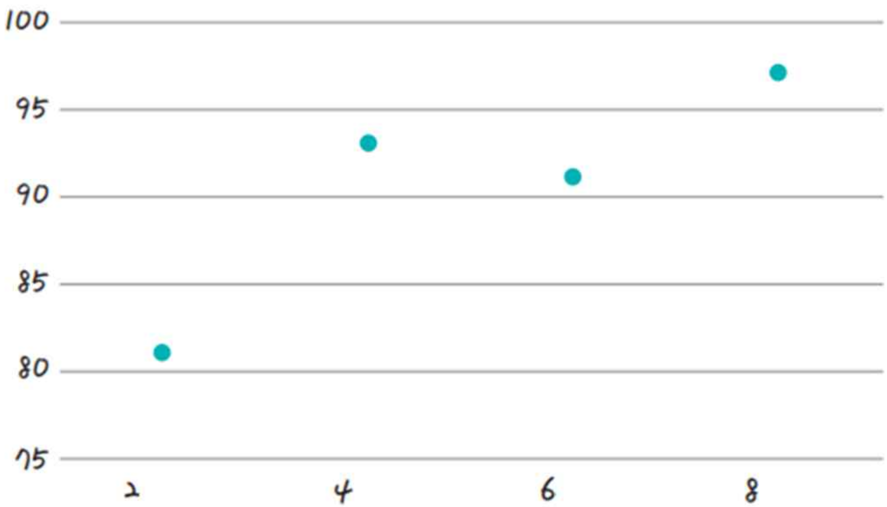
  - 이동거리를 x
  - 금액을 y
  - x와 y집합을 표현하면 (계산의 편리함을 위해 금액을 100으로 나눔)

$$X = \{2,\ 4,\ 6,\ 8\}$$
$$Y = \{81,\ 93,\ 91,\ 97\}$$

예측선

가장 좋은 예측선 만들기

- 거리와 금액을 좌표로 표현



예측선

가장 좋은 예측선 만들기

- 왼쪽이 아래로, 오른쪽이 위를 향하는 선형(선으로 표시될 수 있는 형태)로 보임
- 이 특징이 가장 잘 나올 선을 그리기
- 직선이므로 => 일차함수
- 일차함수 식?

$$y = ax + b$$

- 선을 잘 그리려면?
  - x값에 따라 y는 바뀜
  - 정확하게 계산하려면 a와 b값을 알아내야함
  - 직선의 기울기 a와 y절편 b값을 정확히 예측 해야함
  - 선형회귀는 최적의 a와 b값 찾기
- 선을 왜 잘 그려야 하나?
  - 선을 그려 놓으면 나중에 없는 데이터 부분에서도 y값을 예측
  - 예) 5km 이동 했을 때 금액은?

최소 제곱법

- 정확한 선을 긋기 위해
  - 기울기 a와 절편 b를 찾기
  - 최소제곱법 공식으로 해결
- x값(거리), y값(금액)을 이용해 기울기 a를 구하는 식 (최소제곱법 공식)

$$a = \frac{(x - x\ \text{평균})(y - y\ \text{평균})\text{의 합}}{(x - x\ \text{평균})^2\text{의 합}} \qquad (\text{식 } 4.1)$$

- 실제 데이터를 넣어보면

이동 시간(x) 평균: (2 + 4 + 6 + 8) ÷ 4 = 5
금액(y) 평균: (81+ 93 + 91 + 97) ÷ 4 = 90.5

최소 제곱법

- 식에 대입하면

$$
\begin{aligned}
a &= \frac{(2-5)(81-90.5)+(4-5)(93-90.5)+(6-5)(91-90.5)+(8-5)(97-90.5)}{(2-5)^2+(4-5)^2+(6-5)^2+(8-5)^2} \\
&= \frac{46}{20} \\
&= 2.3
\end{aligned}
$$

- 기울기는 2.3
- Y의절편 b를 구하려면

$$b = y\text{의 평균} - (x\text{의 평균} \times \text{기울기 } a) \qquad \text{(식 4.2)}$$

- 값을 모두 대입하면

$$
\begin{aligned}
b &= 90.5 - (2.3 \times 5) \\
&= 79
\end{aligned}
$$

- 절편 b는 79 따라서 직선의 방정식은

$$y = 2.3x + 79$$

최소 제곱법

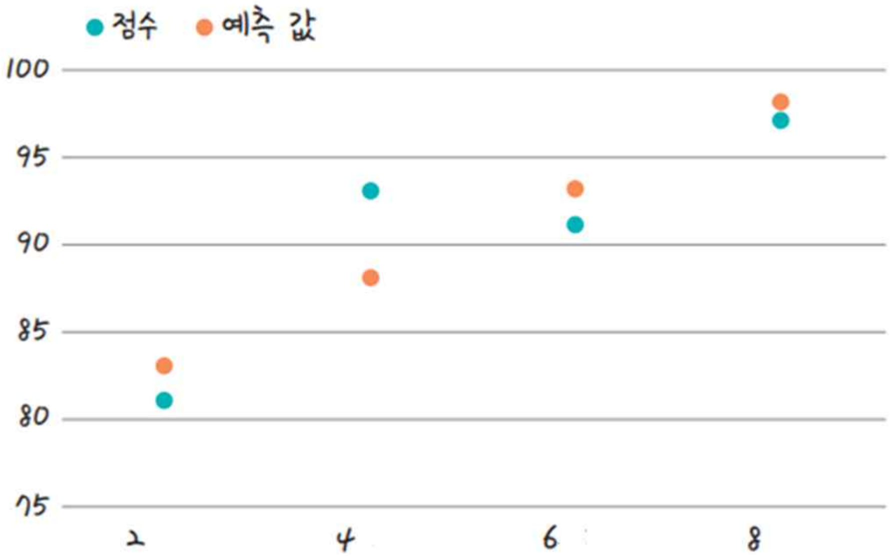
- 실제 데이터를 대입해 보면?

$$y = 2.3x + 79$$

이동 거리	2	4	6	8
금액	81	83	91	97
예측 값	83.6	88.2	92.8	97.4

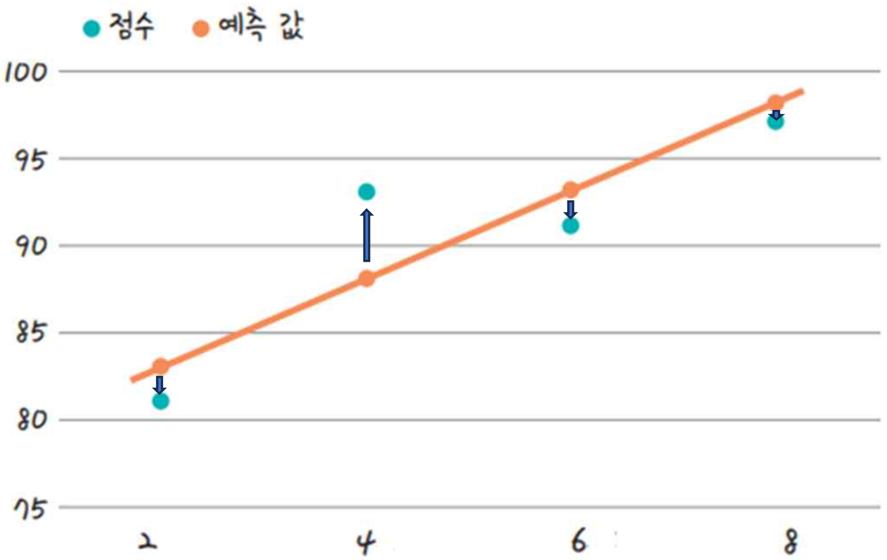
최소 제곱법

- 그림으로 찍어보면?



최소 제곱법

- 직선을 그으면?
  - 오차의 최소화



코딩으로 예측선 만들기

코딩으로 예측선 만들기

택시 이동 거리에 따른 금액 예측선 만들기

- 이동거리, 금액을 배열로 저장
- 넘파이 배열을 사용하기 위해 넘파이 import도 수행

1. 환경 준비

```
import numpy as np
```

2. 데이터 준비

```
# 이동거리와 금액 각각 x, y라는 이름의 넘파이 배열로 만듭니다.
x = np.array([2, 4, 6, 8])
y = np.array([81, 93, 91, 97])
```

코딩으로 예측선 만들기

택시 이동 거리에 따른 금액 예측선 만들기

- 기울기 a와 y절편 b 구하기 (최소 제곱근 공식)
- X의 모든 원소 평균을 구하는 넘파이 함수 mean()
- mx 변수에는 x원소들의 평균값, my에는 y원소의 평균값

```
mx = np.mean(x)
my = np.mean(y)
```

[x와 y의 평균값]

```
#x의 평균값을 구합니다.
mx = np.mean(x)

#y의 평균값을 구합니다.
my = np.mean(y)

# 출력으로 확인합니다.
print("x의 평균값:", mx)
print("y의 평균값:", my)
```

```
x의 평균값: 5.0
y의 평균값: 90.5
```

코딩으로 예측선 만들기

택시 이동 거리에 따른 금액 예측선 만들기

- 최소 제곱근 공식 중 분모 값 만들기
  - X의 각원소와 x의 평균값들의 차를 제곱

```
divisor = sum([(i - mx)**2 for i in x])
```

sum( )은 Σ에 해당하는 함수입니다.

제곱을 구하라는 의미입니다.

x의 각 원소를 한 번씩 i 자리에 대입하라는 의미입니다.

코딩으로 예측선 만들기

택시 이동 거리에 따른 금액 예측선 만들기

- 분자에 해당하는 값
  - x와 y의 편차를 곱해서 합한 값

```
def top(x, mx, y, my):
    d = 0
    for i in range(len(x)):
        d += (x[i] - mx) * (y[i] - my)
    return d
dividend = top(x, mx, y, my)
```

코딩으로 예측선 만들기

택시 이동 거리에 따른 금액 예측선 만들기

- 분모와 분자 계산 코드

[기울기 공식의 분모와 분자]

```
# 기울기 공식의 분모 부분입니다.
divisor = sum([(i - mx)**2 for i in x]

# 기울기 공식의 분자 부분입니다.
def top(x, mx, y, my):
    d = 0
    for i in range(len(x)):
        d += (x[i] - mx) * (y[i] - my)
    return d
dividend = top(x, mx, y, my)

# 출력으로 확인합니다.
print("분모:", divisor)
print("분자:", dividend)
```

```
분모: 20.0
분자: 46.0
```

택시 이동 거리에 따른 금액 예측선 만들기

- 기울기와 y절편

3. 기울기와 y 절편 구하기

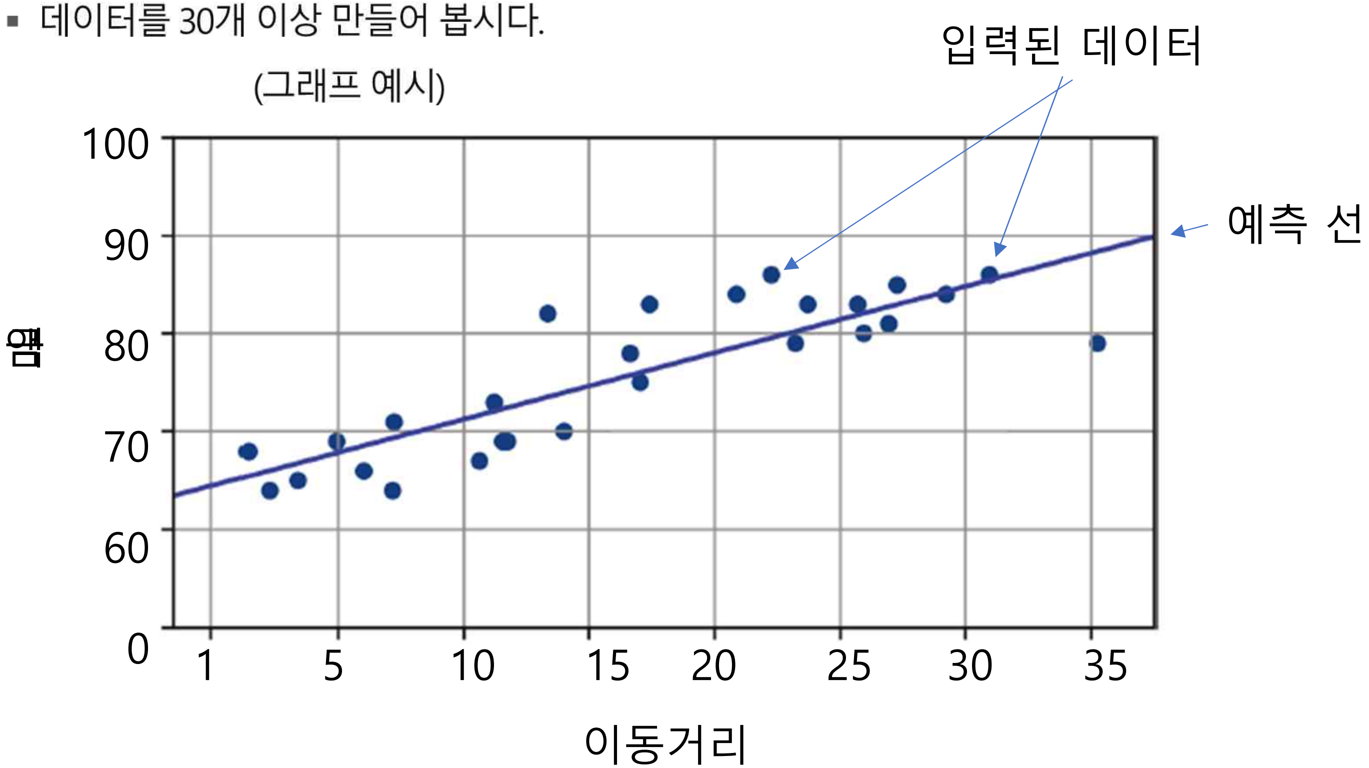
```
[ ] # 기울기 a를 구하는 공식입니다.
    a = dividend / divisor

    # y절편 b 를 구하는 공식입니다.
    b = my - (mx*a)

    # 출력으로 확인합니다.
    print("기울기 a =", a)
    print("y절편 b =", b)
```

```
기울기 a = 2.3
y절편 b = 79.0
```

택시 이동 거리에 따른 금액 예측선 만들기

- 이제 데이터를 바꿔보며 실습해 보세요.
- 그리고 해당하는 기울기와 절편 값에 따른 식을 그래프로 표현해 보세요.
  - 데이터를 30개 이상 만들어 봅시다.

(그래프 예시)



코딩으로 예측선 만들기

실습 결과 제출방법

- 첫 번째 => 데이터 표시 (표 형태)
- 두 번째 => 데이터에 대해서 계산된 식 제시
- 세 번째 => 데이터에 따른 그래프에 점 출력, 식에 따른 예측선 그림 (이전 페이지 예시와 같이)
- 위 세 가지 내용이 포함된 PPT, HWP, Docs 등 제한없이 작성하여 eclass 과제에 업로드