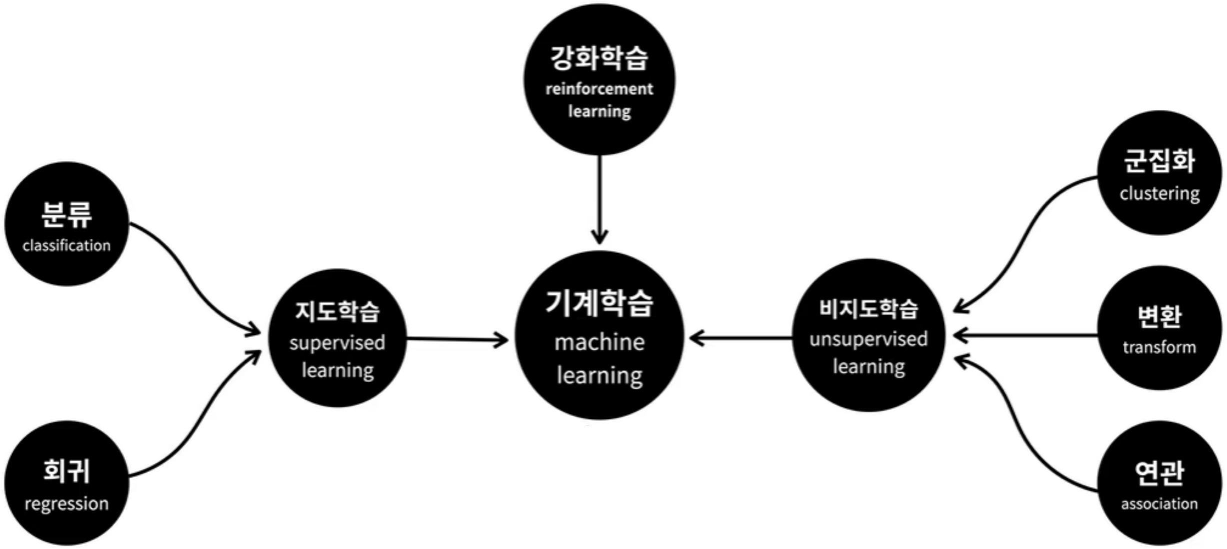
02. 신경망 기초 이론

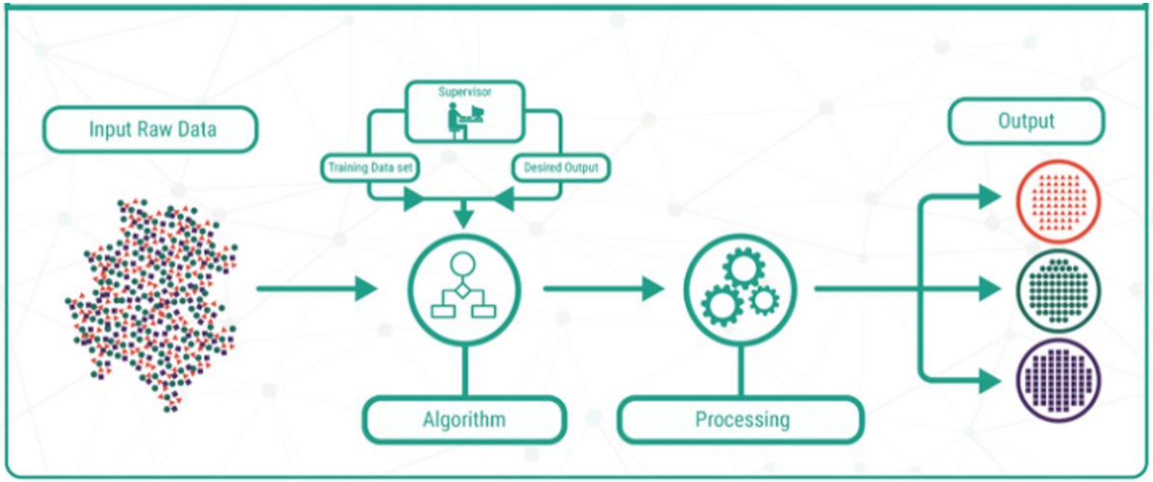
2.1 머신러닝의 정의

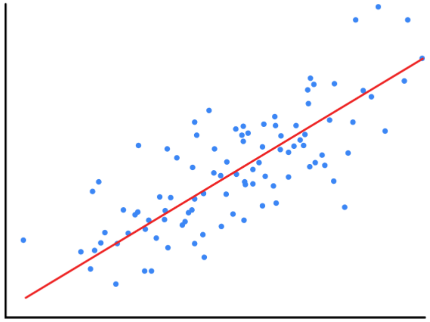
- 우리가 배우려는 기계학습은 무엇인가?
  - 경험(E)로부터 학습하여 일(T)에 대한 성능(P)을 올리는 것 (톰 미첼 교수)



2.2 지도학습

- 답이 정해진 데이터를 입력하여 정답을 찾는 규칙(함수)을 찾는 것
  - 경험(E)로부터 학습하여 일(T)에 대한 성능(P)을 올리는 것 (톰 미첼 교수)





2.2 지도학습

유전인자, 방사선 노출, 식이
X_1 X_2 X_3

음주, 흡연, 스트레스
X_4 X_5 X_6

$H(x)=w_1x_1+w_2x_2+w_3x_3+ w_4x_4+w_5x_5+w_6x_6+ b$

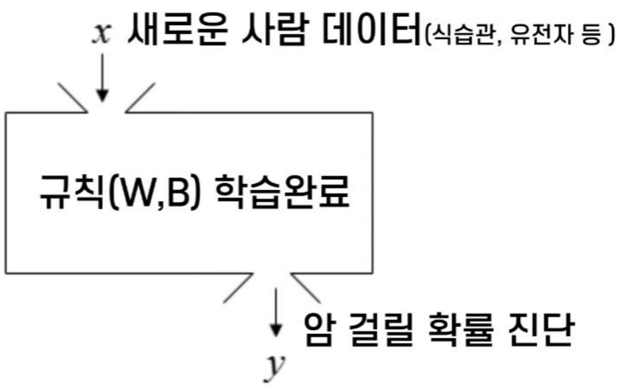
W는 가중치, X는 각 요인

$$H(x) = w_1x_1+w_2x_2+w_3x_3+...+b$$

암에 걸릴 확률 = 유전인자의 중요도(W_1)* 유전인자 수치(X_1)+
방사선노출의 중요도(W_2)*방사선 노출량(X_2)+...

(W = 각 X에 대한 중요도를 구별하기 위해 도입)

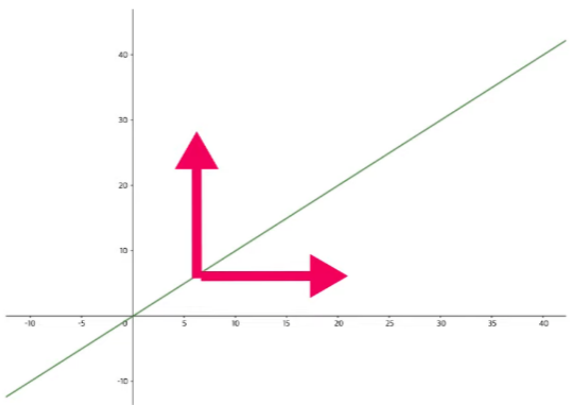
결론(암걸렸는지 유무) = 유전인자의 영향도(W_1)* 개별 유전인자(X_1)+
방사선노출의 영향도(W_2)* 개별 방사선 노출량(X_2)+
식습관 영향도(W_3)* 개별 식습관(X_3)+ ... +b

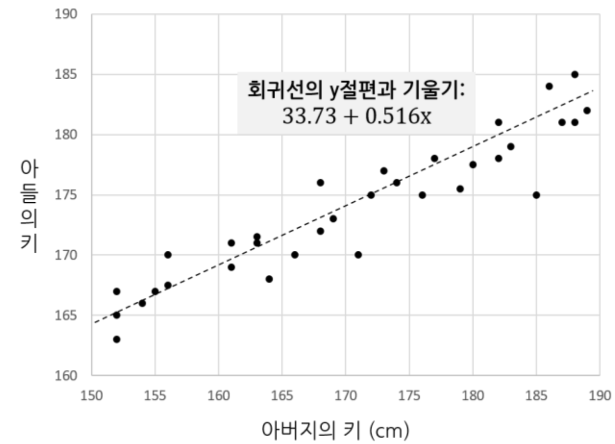


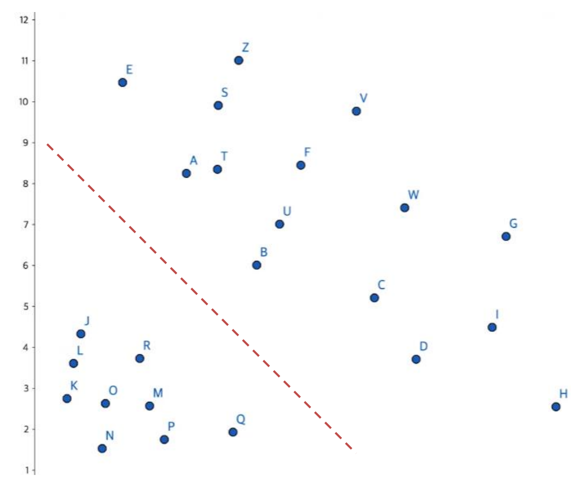
2.2 지도학습

- 선형적으로 설명이 가능한 현상
  - 선형회귀의 원리

선(Line)의 형용사형 = 선형적(Linear)



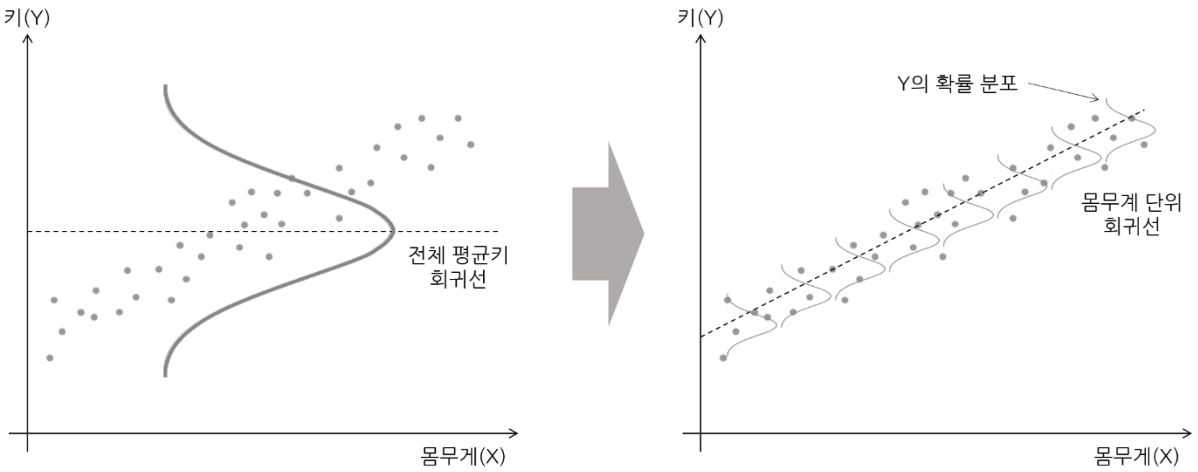


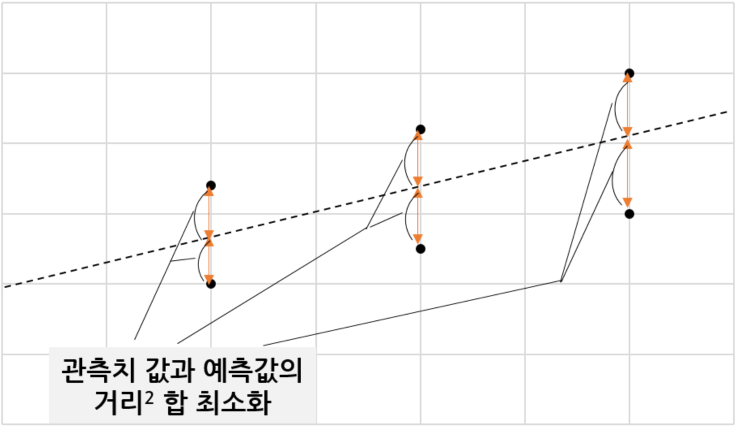


2.2 지도학습

회귀선의 기본 개념

- 예측치와 관측치들 간의 수직 거리(오차)의 제곱합을 최소로 하는 직선이 회귀선

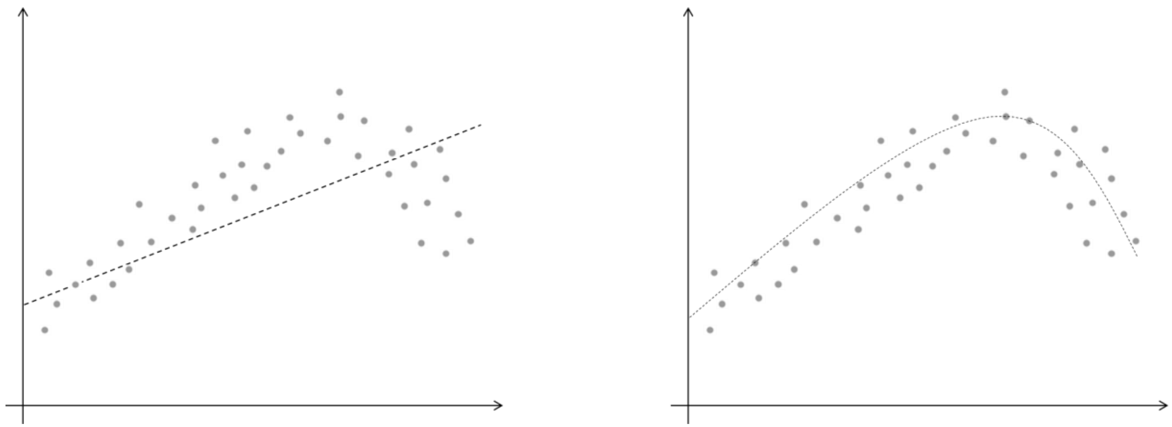




2.2 지도학습

다항회귀의 기본 개념

- 다항회귀란 독립변수와 종속변수의 관계가 비선형 관계일 때 변수에 각 특성의 제곱을 추가하여 회귀선을 비선형으로 변환하는 모델



<선형회귀선과 다항회귀선>

변수	Parameter Estimate	Standard Error	T Value	Pr > |t|	Tolerance	Variance Inflation
Intercept	24.822	0.469	52.97	<.0001	.	0
X1	0.604	0.150	4.03	0.0007	0.069	14.4405
X2	0.221	0.720	0.31	0.767	0.002	48.7001
X3	1.278	3.110	0.41	0.685	0.578	1.72923
X4	-1.394	0.236	-5.91	<.0001	0.324	3.08501
X5	-1.612	1.325	-1.22	0.224	0.211	46.8683

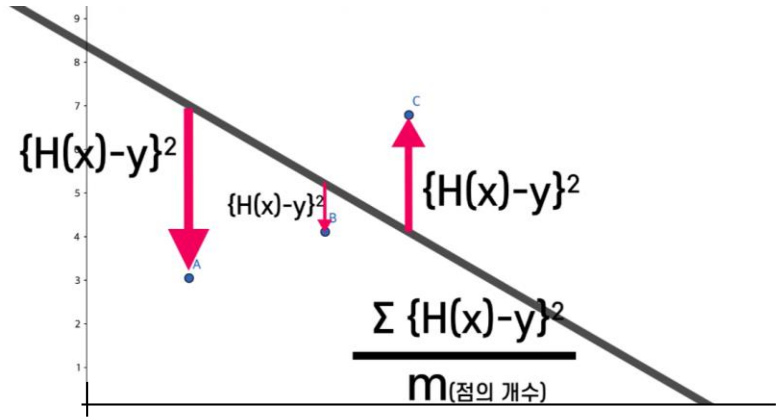
2.2 지도학습

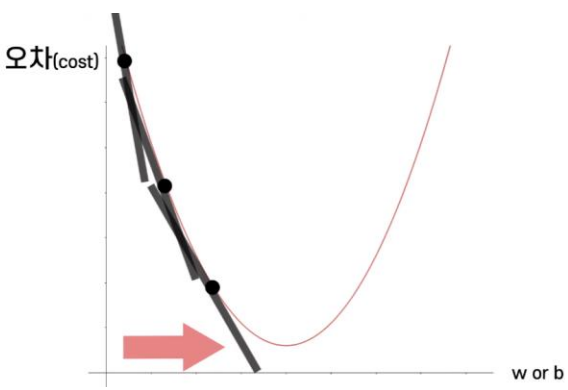
오차 함수(Cost function)의 기본 개념

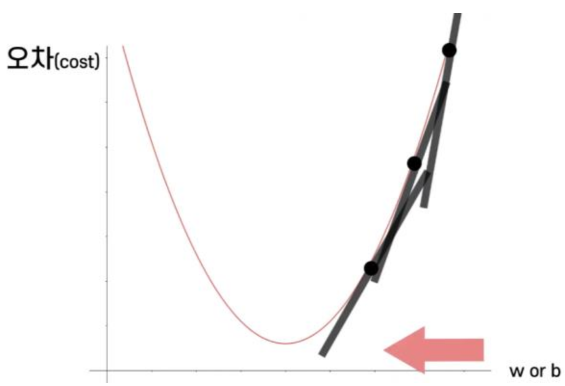
- 모델이 학습을 반복할 때마다 발생하는 오차를 표현
- 모델의 정확도를 높이기 위해 오차함수를 이용하여 오차가 0에 가까운 지점을 탐색
- 임의의 초기값(w,b)을 오차함수의 이차함수에 대입한 후 그 자리를 미분

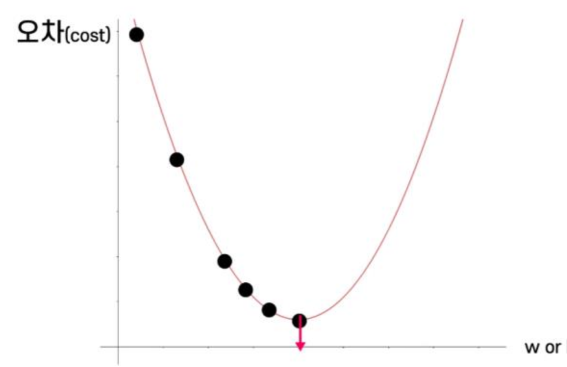
오차함수(Cost Function)

$$\frac{1}{m}\sum_{a=1}^{m}(H(x_a)-y_a)^2$$









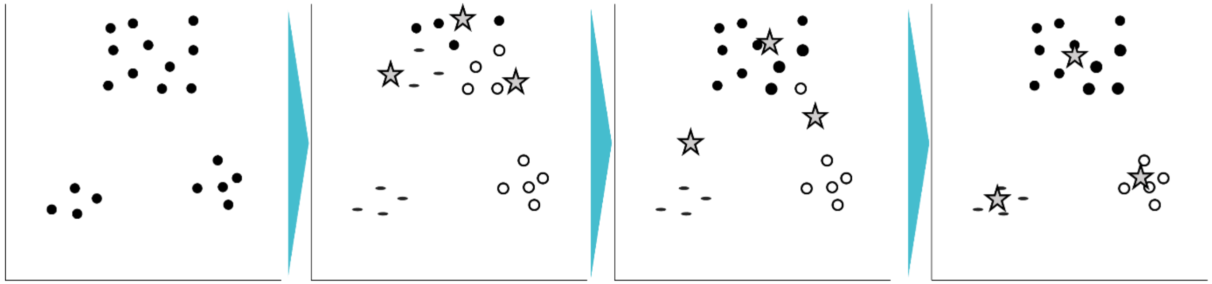
2.2 비지도학습

비지도학습의 기본 개념

- 정답(label)이 정해져 있지 않은 데이터로 학습하는 방식
- 변수 간의 패턴을 파악하거나 데이터를 군집화 하는 방법

k-means 클러스터링

- 특성이 유사한 관측치들을 묶어주는 알고리즘



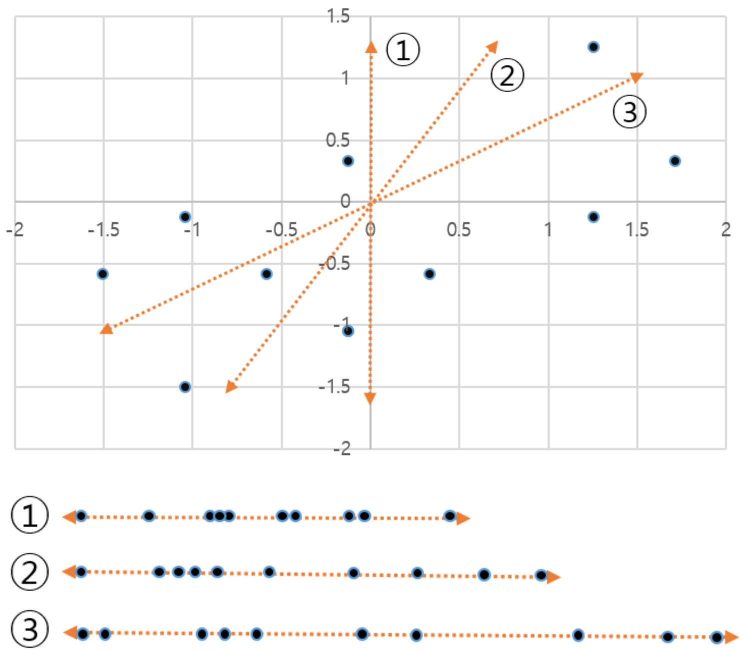
① ② ③ ④

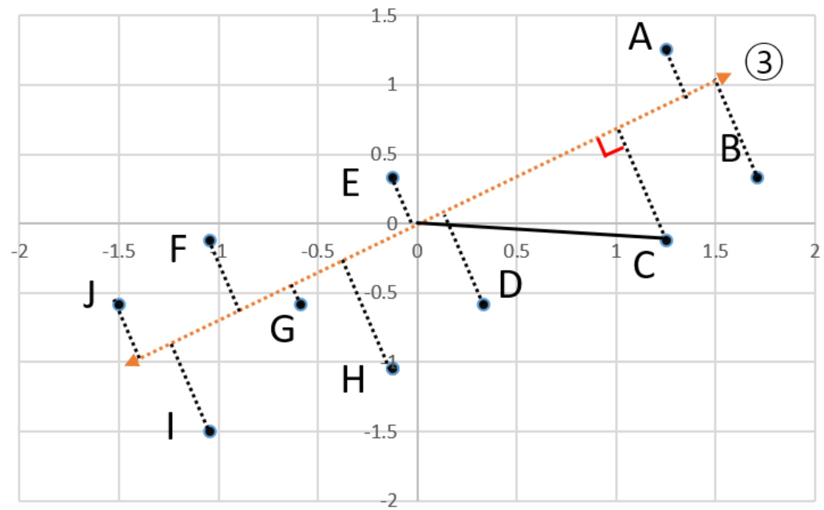
- 1단계: k 개의 중심점을 임의의 데이터 공간에 선정
- 2단계: 각 중심점과 관측치들 간의 유클리드 거리를 계산
- 3단계: 각 중심점과 거리가 가까운 관측치들을 해당 군집으로 할당
- 4단계: 할당된 군집의 관측치들과 해당 중심점과의 유클리드 거리를 계산
- 5단계: 중심점을 군집의 중앙으로 이동(군집의 관측치들 간 거리 최소 지점)
- 6단계: 중심점이 더 이상 이동하지 않을 때까지 2~5단계 반복

2.2 비지도학습

주성분 분석(Principal Component Analysis; PCA)

- 여러 개의 독립변수들을 잘 설명해줄 수 있는 주된 성분을 추출하는 기법
- 여러 개의 변수들이 소수의 특정한 소수의 변수들로 축약되도록 가공하는 것



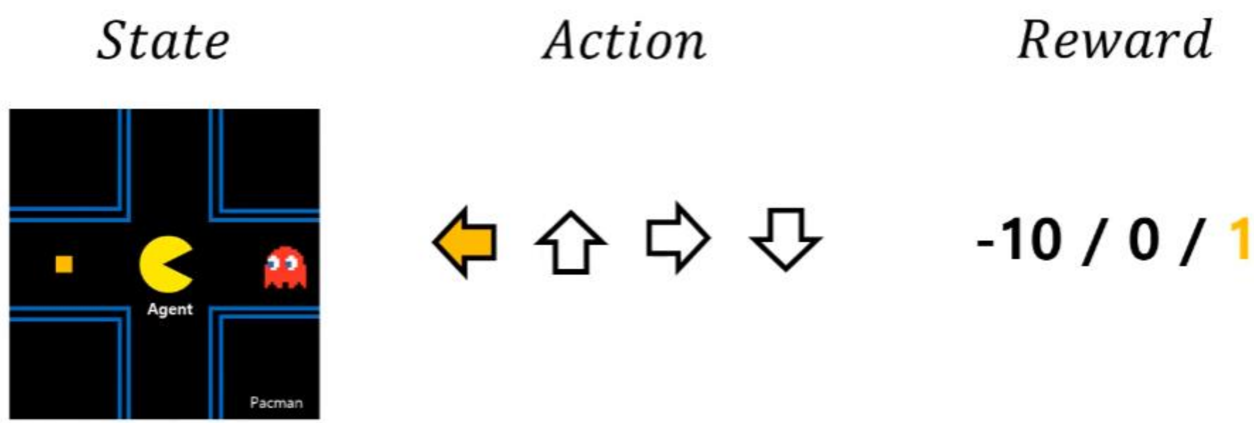


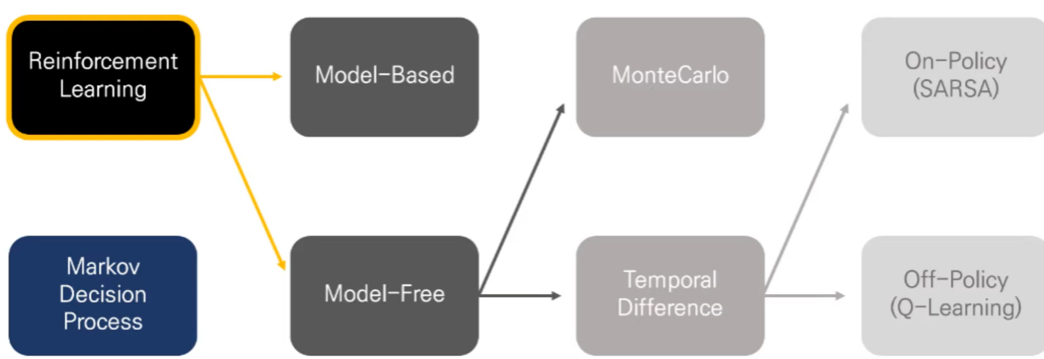
C 포인트로부터 직각으로 맞닿는 지점과 (0,0)의 거리가 최대가 되도록 하는 축을 찾는 것이다.
즉 각 포인트들이 직각으로 맞닿는 지점의 분포가 가장 넓게 퍼진 축을 구하는 것이다.

2.3 강화학습

강화학습의 기본 개념

- 강화학습은 상태(state), 행동(action), 보상(reward)으로 구성
- 순차적인 의사결정 문제에서 누적 보상을 최대화 하기 위해 시행착오를 거쳐서 상황에 따른 행동 정책을 학습

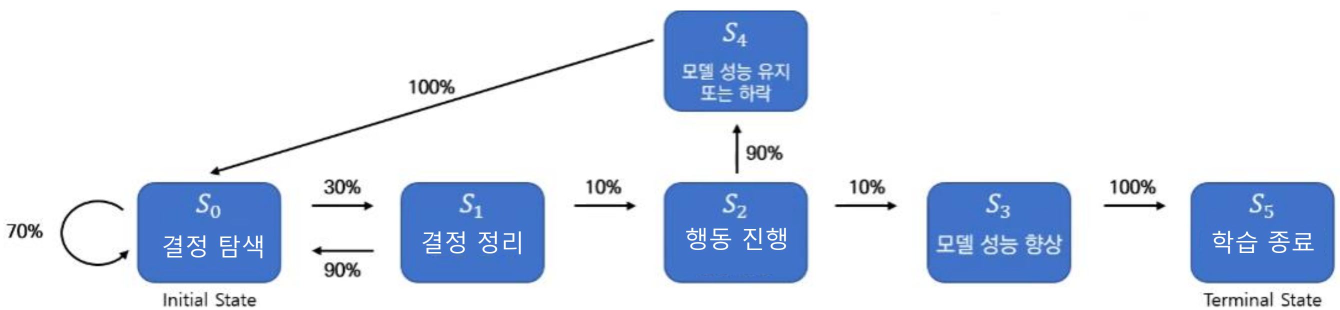




2.3 강화학습

Markov Process(MP)

- 상태와 상태전이확률로 구성되며 상태 간 순차적인 관계를 확인할 수 있음
- 상태전이확률 매트릭스
  - 상태 s에서 상태 s′로 전이할 행렬을 의미



S_{t+1}

S_t

$P_{ss'}$	S_0	S_1	S_2	S_3	S_4	S_5
S_0	0.7	0.3				
S_1	0.9		0.1			
S_2				0.1	0.9	
S_3						1.0
S_4	1.0					
S_5						

<벨만 기대방정식(Bellman expectation equation)>

$$v_\pi(s) = \sum_{a \in A} \pi(a|s) \left(r_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s') \right)$$

현재 상태에서 특정 행동을 선택할 확률

특정 다음 상태의 가치

현재 상태에서 특정 행동을 통해 얻는 보상

현재 상태에서 특정 행동을 하였을 때 특정 다음 상태로 넘어갈 확률

추천 영상 – 초거대 AI의 현재와 미래

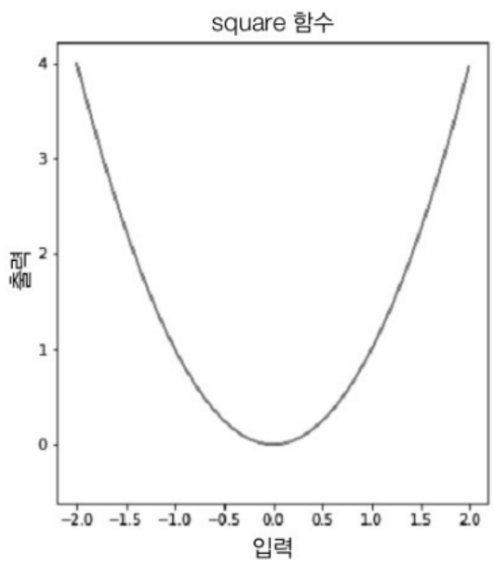
2.4 함수

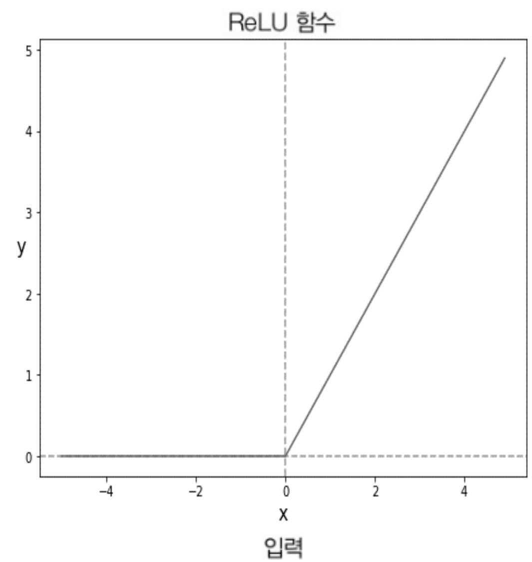
기본 함수 수식

- $f_1(x) = x^2$
- $f_2(x) = max(x,0)$

- 서로 다른 함수 f_1과 f_2 가 숫자 x 를 입력 받으며, 함숫값이 각각 x^2 과 $xmax(x,0)$

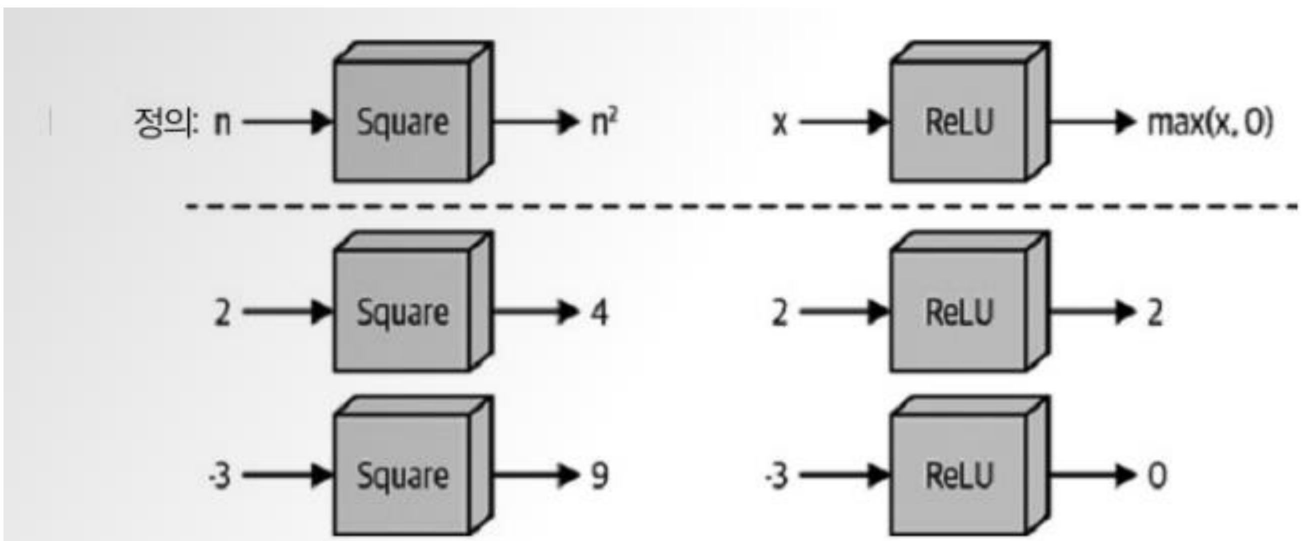
(Rectified Linear Unit, 경사함수)





2.4 함수

- 함수를 나타내는 또 다른 방법
  - 상자에 숫자로 입력값을 집어넣으면 상자 안에 정의된 규칙에 따라 계산된 출력값 산출



2.4 함수

함수를 파이썬 코드로 구현

- 넘파이의 ndarray 클래스를 사용하면 다차원 배열을 직관적이고 효율적으로 다룰 수 있음

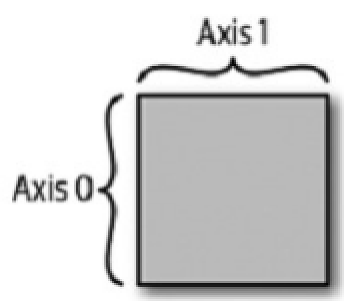
```
print("파이썬 리스트를 사용한 연산:")
a = [1,2,3]
b = [4,5,6]
print("a+b:", a + b)
try:
    print(a * b)
except TypeError:
    print("a*b 파이썬 리스트에 a*b와 같은 연산을 할 수 없음")
print()
print("넘파이 배열을 사용한 연산:")
a = np.array([1,2,3])
b = np.array([4,5,6])
print("a+b:", a + b)
print("a*b:", a * b)
```

```
파이썬 리스트를 사용한 연산:
a+b: [1, 2, 3, 4, 5, 6]
a*b 파이썬 리스트에 a*b와 같은 연산을 할 수 없음

넘파이 배열을 사용한 연산:
a+b: [5 7 9]
a*b: [ 4 10 18]
```

2.4 함수

함수를 파이썬 코드로 구현

- 넘파이의 ndarray 클래스는 n차원 배열을 다루는 데 필요한 다양한 기능 제공
- 여러 개의 축을 가지며 축에는 각각 0부터 시작하는 인덱스가 매겨짐
- 2차원 ndarray 클래스를 예로 들면,
  열방향 축이 axis = 0
  행방향 축이 axis = 1
-



<axis = 0이 열방향 축이고 axis = 1이 행방향 축인 2차원 넘파이 배열>

2.4 함수

함수를 파이썬 코드로 구현

- Axis 0 방향으로(2차원 배열의 열방향) 합을 구하는 방법으로 '배열을 요약'하면, 원래 배열보다 한 차원 낮은 배열이 반환

```
a = np.array([[1, 2],
              [3, 4]])
print('a:')
print(a)
print('a.sum(axis=0):', a.sum(axis=0))
print('a.sum(axis=1):', a.sum(axis=1))
```

```
a:
[[1 2]
 [3 4]]
a.sum(axis=0): [4 6]
a.sum(axis=1): [3 7]
```

- 배열에 마지막 축의 방향으로 다른 1차원 배열을 합하는 연산도 가능

```
a = np.array([[1,2,3],
              [4,5,6]])
print(a)

b = np.array([10,20,30])
print("a + b:\n", a + b)
```

```
a + b:
 [[11 22 33]
 [14 25 36]]
```

2.5 도함수

도함수란?

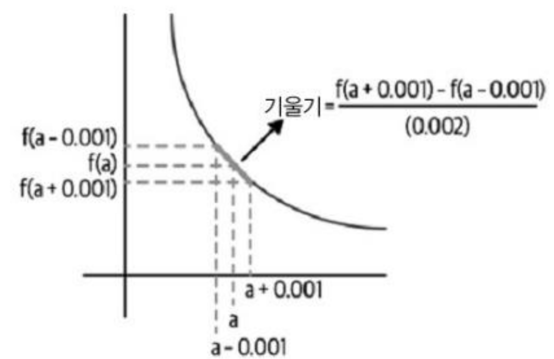
- 특정한 x값이 아닌 임의의 x에 관한 순간변화율을 나타낸(일반화한) 것
- 함수 y=f(x)를 x에 관해서 미분해 얻어지는 함수 y=f'(x)를 뜻함
- 일반적으로 f(x)의 미분계수, 순간변화율 이라고도 함

- 함수의 입력값에 대한 함숫값 f의 변화율을 정확히 계산하기 위해 극한 사용

$$\frac{df}{du}(a) = \lim_{\Delta \to 0} \frac{f(a+\Delta) - f(a-\Delta)}{2 \times \Delta}$$

- 극한값은 △에 매우 작은 값, 이를테면 0.001을 대입하는 방법으로 다음과 같이 표현

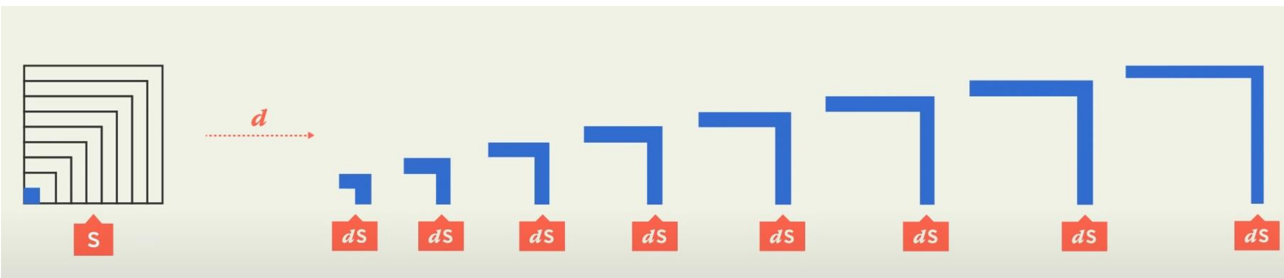
$$\frac{df}{du}(a) = \frac{f(a+0.001) - f(a-0.001)}{0.002}$$

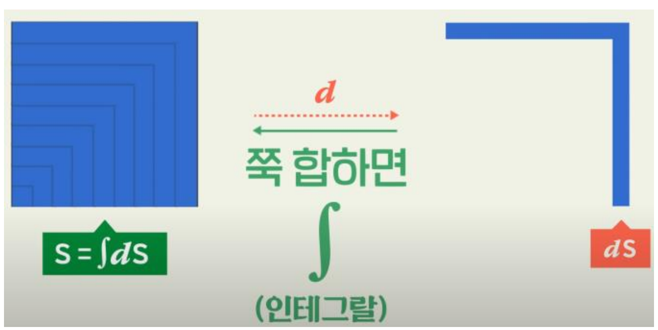


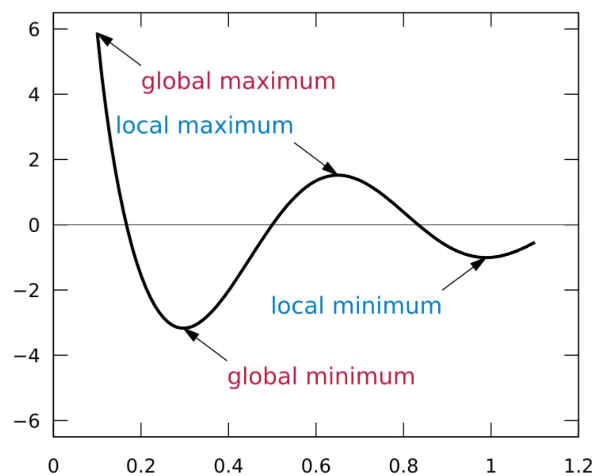
2.5 도함수

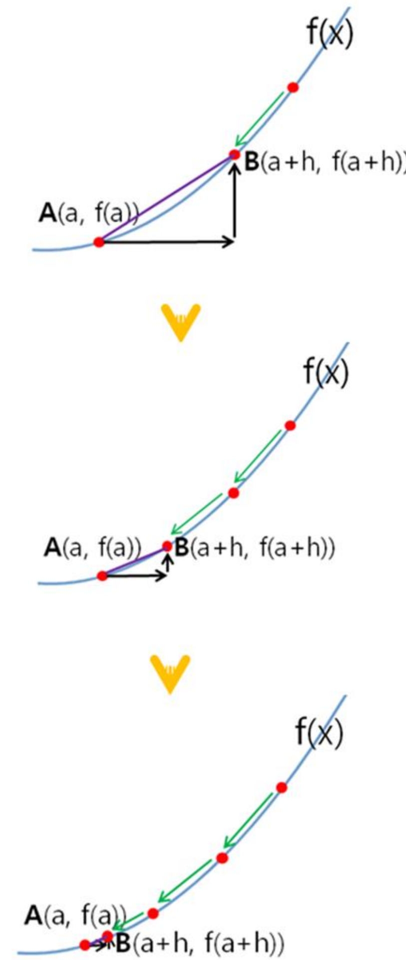
미분 먼저 되짚기

- 순간의 변화를 예측하는 수학적 도구
- 한 점에서의 기울기를 의미 (곡선 그래프의 기울기 측정)
- 미분 함수를 통해 해당 AI 모델의 성능을 평가
- AI 모델이 어느 방향으로 학습해야 성능이 좋아지는지 계산





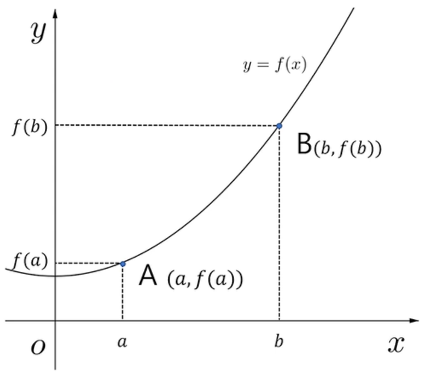




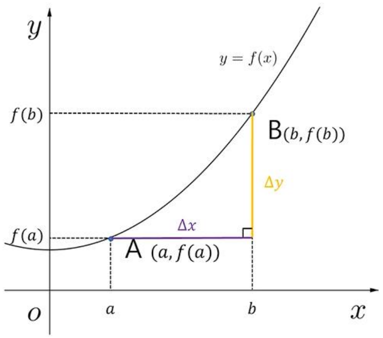
2.5 도함수

미분 먼저 되짚기

- **평균** 변화율
  - 두 점 사이의 기울기

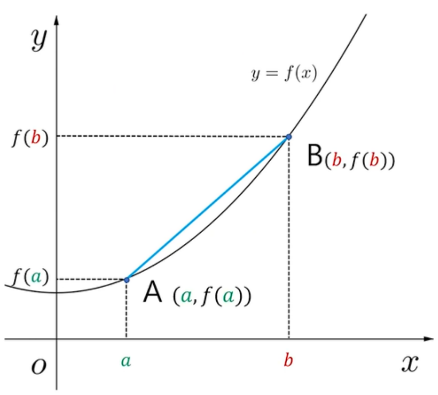


$$\frac{\Delta y}{\Delta x} = \frac{y\text{ 증분}}{x\text{ 증분}}$$



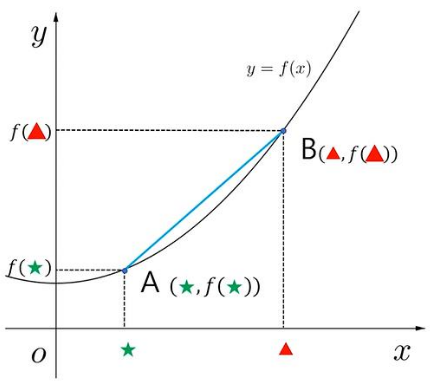
$$\frac{\Delta y}{\Delta x} = \frac{y\text{ 증분}}{x\text{ 증분}}$$

$$= \frac{f(b) - f(a)}{b - a}$$



$= \overline{AB}$ 의 기울기

$$= \frac{f(b) - f(a)}{b - a}$$



$= \overline{AB}$ 의 기울기

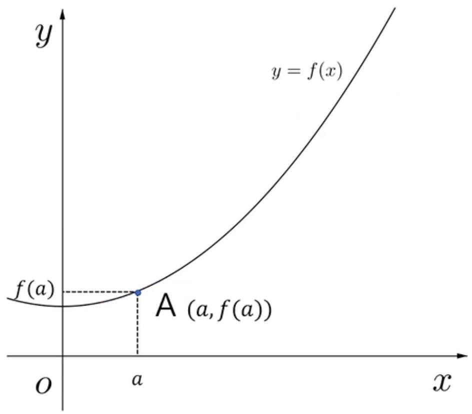
$$= \frac{f(▲) - f(★)}{▲ - ★}$$

모양이 같아야 함

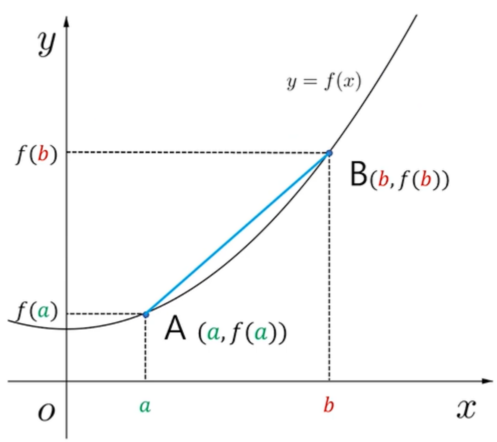
2.5 도함수

■ 미분 먼저 되짚기

- **순간** 변화율
  - 한 점의 기울기



$\mathrm{A}(a, f(a))$ 에서의 순간변화율



$\mathrm{A}(a, f(a))$ 에서의 순간변화율

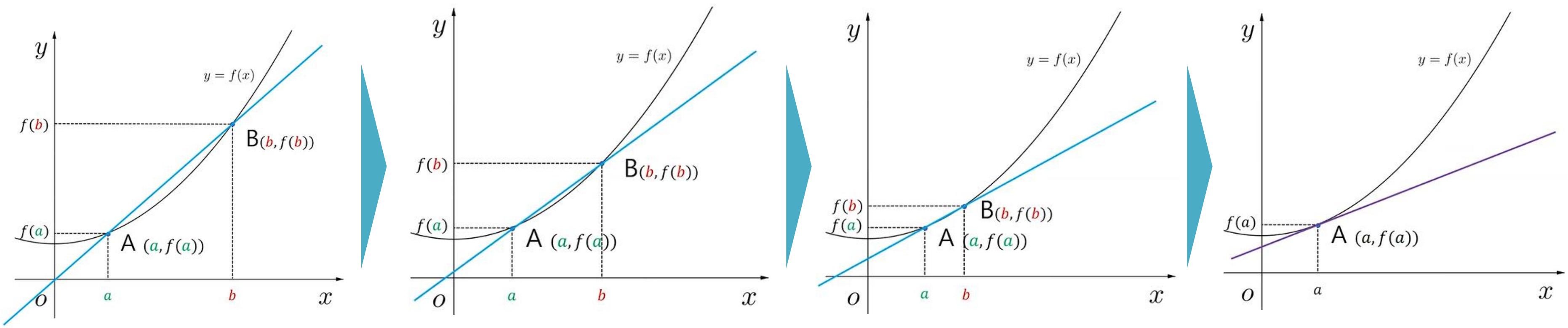
$\overline{AB}$ 의 평균변화율

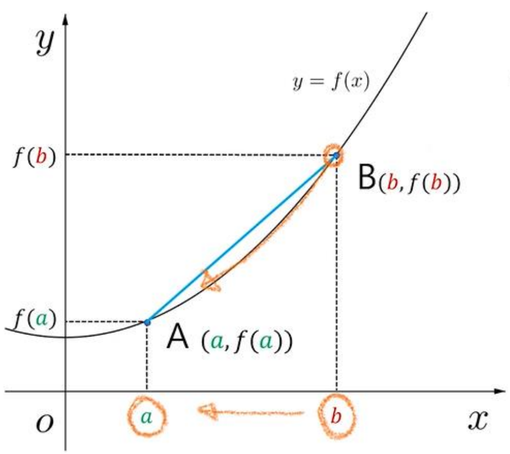
$$= \frac{f(b) - f(a)}{b - a}$$

2.5 도함수

■ 미분 먼저 되짚기

- **순간** 변화율
  - 한 점의 기울기





$A(a, f(a))$ 에서의 순간변화율

$= \lim_{b \to a} \overline{AB}$ 의 평균변화율

$$= \lim_{b \to a} \frac{f(b) - f(a)}{b - a}$$

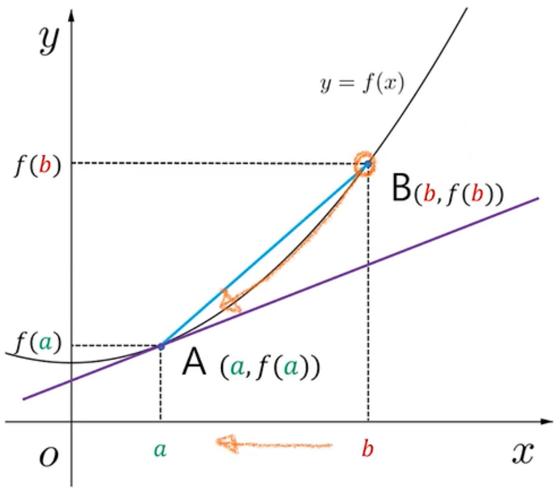
동점

정점

2.5 도함수

미분 먼저 되짚기

- 미분 계수



$$f'(a) = \lim_{b \to a} \frac{f(b) - f(a)}{b - a}$$

$$= \lim_{x \to a} \frac{f(x) - f(a)}{x - a}$$

$$= \lim_{a+h \to a} \frac{f(a+h) - f(a)}{a+h-a}$$

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

2.5 도함수

미분 먼저 되짚기

- 미분 계수 연습문제

다항함수 $f(x)$에 대하여 $\lim_{h \to 0} \frac{f(4+h)-f(4)}{3h} = 7$일 때, $f'(4)$의 값을 구하시오.

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

$$f'(4) = \lim_{4+h \to 4} \frac{f(4+h)-f(4)}{4+h-4}$$

$$= \lim_{h \to 0} \frac{f(4+h)-f(4)}{h}$$

$$\lim_{h \to 0} \frac{f(4+h)-f(4)}{3h} = \frac{f'(4)}{3} = 7$$

$$f'(4) = 21$$

2.5 도함수

- 미분한다 = 도함수를 구한다

$$f(x) \xrightarrow{\text{미분}} f'(x)$$

도함수

$f(x) = x^2 + 3x$ 일 때,

$$f'(1) = \lim_{x \to 1} \frac{f(x) - f(1)}{x - 1} = \lim_{x \to 1} \frac{x^2 + 3x - 4}{x - 1}$$

$$= \lim_{x \to 1} \frac{(x-1)(x+4)}{x-1} = 5$$

$$f'(2) = \lim_{x \to 2} \frac{(x-2)(x+5)}{x-2} = 7$$

$$f'(3) = \lim_{x \to 3} \frac{(x-3)(x+6)}{x-3} = 9$$

$$\vdots$$

$f'(x)$
도함수

2.5 도함수

■ 도함수의 정의

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

■ 도함수 예시 $f(x) = x^2$

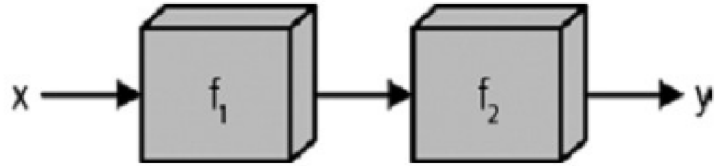
$$\begin{aligned} f'(x) &= \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \\ &= \lim_{h \to 0} \frac{(x+h)^2 - x^2}{h} \\ &= \lim_{h \to 0} \frac{x^2 + 2hx + h^2 - x^2}{h} \\ &= \lim_{h \to 0} (2x + h) \\ &= 2x \end{aligned}$$

$\Rightarrow$ x^2**의 도함수**

2.6 합성함수

합성함수란?

- 함수는 다른 함수를 **안는** 방식으로 또 다른 함수를 **합성**할 수 있음
- 입력값이 첫 번째 함수에서 한 번 변환된 다음, 그 출력이 다시 두 번째 함수에 입력되고 다시 변환된 후 그 값이 출력됨



합성함수의 수식

- 연산 순서는 안쪽 함수 다음 바깥쪽 함수

$$f_2(f_1(x)) = y$$

2.6 합성함수

합성함수코드 구현

- 입력된 데이터가 두 함수의 합성함수로 처리되도록 정의

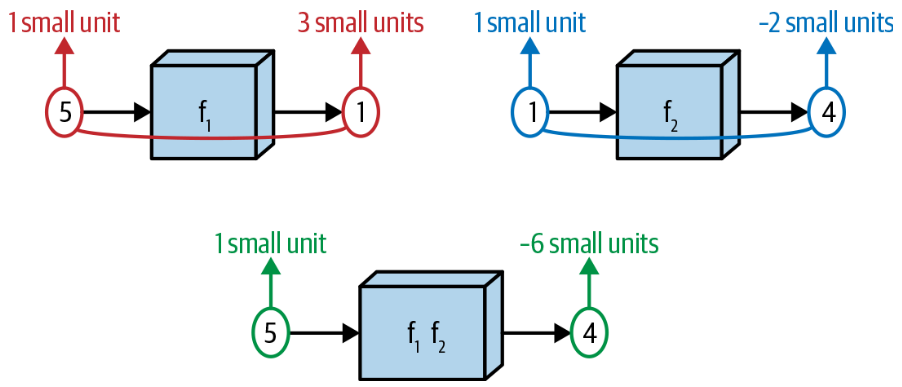
```
from typing import List

# ndarray를 인자로 받고 ndarray를 반환하는 함수
Array_Function = Callable[[ndarray], ndarray]

# Chain은 함수의 리스트다.
Chain = List[Array_Function]
```

```
def chain_length_2(chain: Chain,
                   x: ndarray) -> ndarray:
    '''
    두 함수를 연쇄(chain)적으로 평가
    '''
    assert len(chain) == 2, \
    "인자 chain의 길이는 2여야 함"

    f1 = chain[0]
    f2 = chain[1]

    return f2(f1(x))
```

2.7 연쇄법칙

연쇄법칙이란?

- 연쇄법칙(chain rule)을 이용해 합성함수의 도함수를 계산
- 딥러닝 모델은 수학적으로 보면 합성함수이며, 모델을 학습하려면 합성함수의 도함수가 반드시 필요



2.7 연쇄법칙

연쇄법칙 그래프

- 함숫값이 증가할 때는 도함숫값이 양수
- 함숫값이 변화하지 않을 때는 도함숫값이 0
- 함숫값이 감소할 때는 도함숫값이 음수
- 합성함수를 구성하는 각각의 함수가 미분 가능하다는 조건을 만족하면 수식, 코드로 합성 함수의 도함수 계산 가능