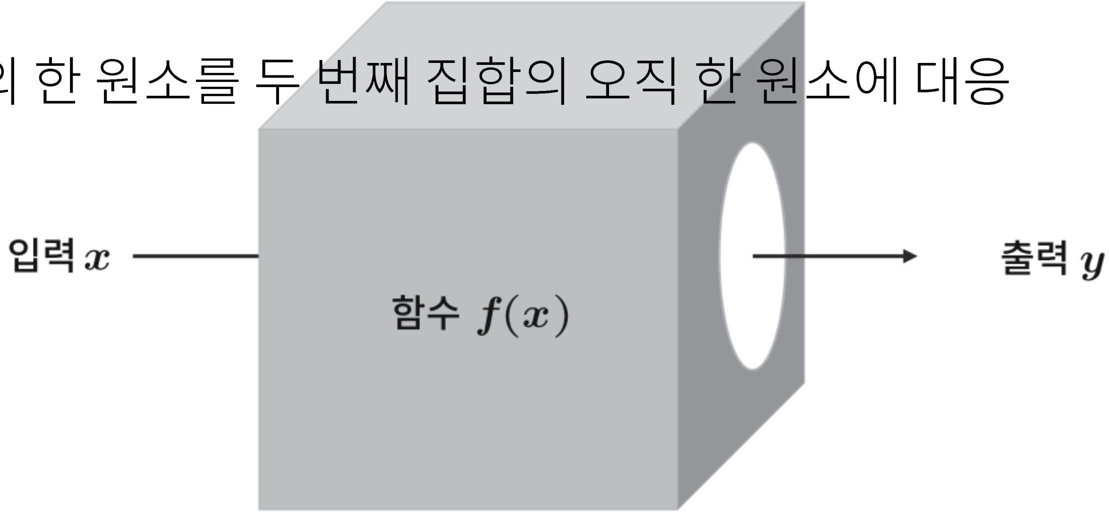
인공지능 이산수학

함수

조준우
metamath@gmail.com

함수란?

- 입력되는 변수와 출력되는 변수가 있을 때
- 입력되는 변수가 어떤 형태로든 연산이 되어 출력변수로 나온다.
- 함수 : 첫 번째 집합의 임의의 한 원소를 두 번째 집합의 오직 한 원소에 대응시키는 대응 관계



함수의 표기

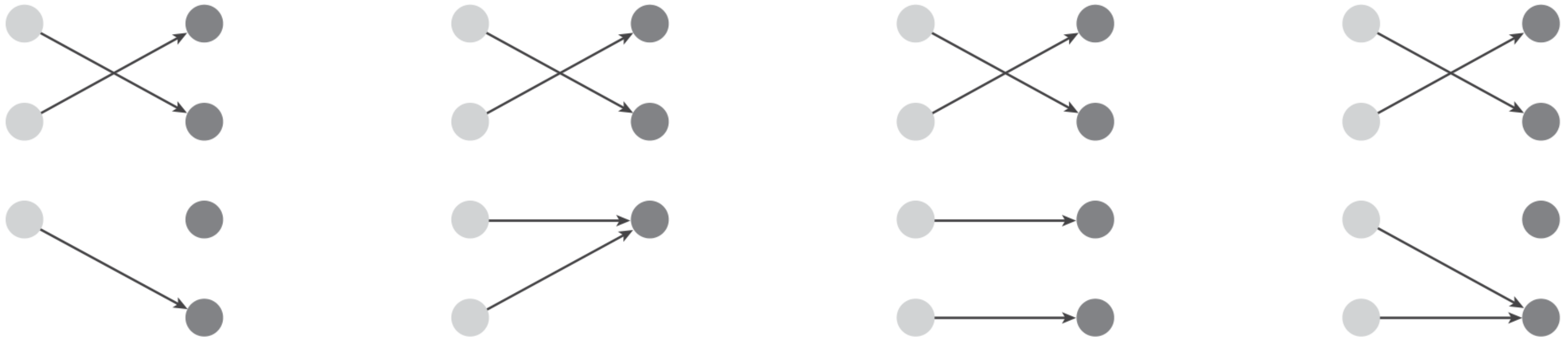
- 표기
  - $f: X \to Y$, X : 정의역domain, Y : 공역codomain
  - X: 독립변수, Y: 종속변수

- 다른 표현
  - $f(X) = Y$

- 표기 예제
  - 실수 집합 : $\mathbb{R}$로 쓸 때 $f : \mathbb{R} \to \mathbb{R}$는 실수를 실수로 대응시키는 함수, 예 : $f : x \to x^2$
  - $f : \mathbb{R}^2 \to \mathbb{R}, f(x_1, x_2) = x_1 + x_2$

함수 대응관계의 특징

- 성질 : 사랑의 작대기



일대일 함수one-to-one (단사함수 Injection)

전사 함수onto (Surjection)

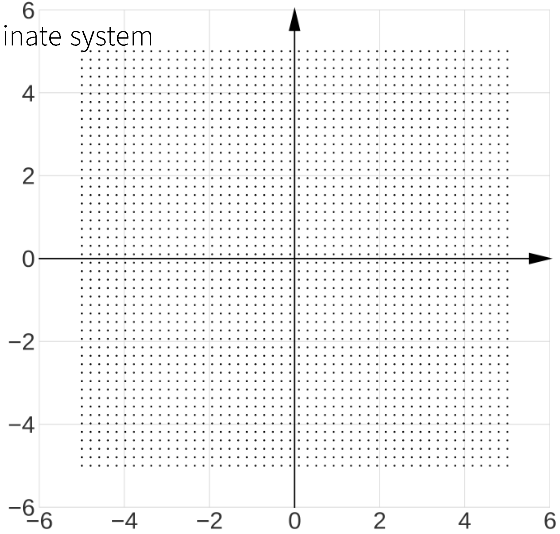
일대일 대응 함수 one-to-one correspondence (전단사함수 Bijection)

함수 (Function)

함수의 표현: 좌표계

- 좌표계$_{\text{coordinate system}}$
  - 공간에 존재하는 대상을 고유한 숫자로 나타내기 위해 사용하는 시스템

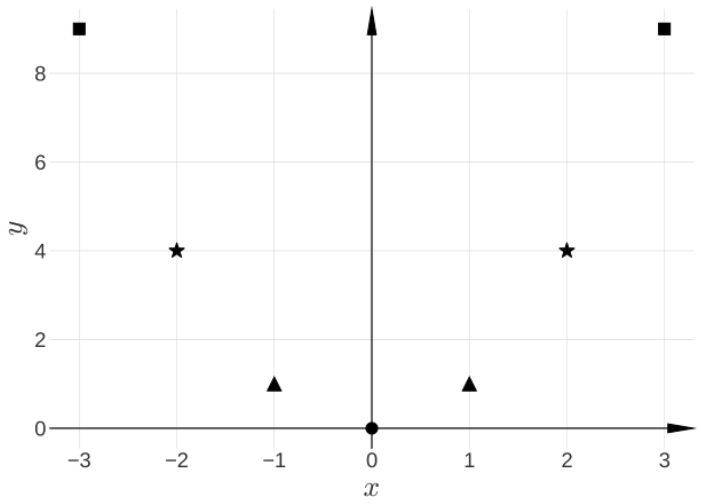
- 직교 좌표계$_{\text{rectangular coordinate system}}$
  - 데카르트 좌표계$_{\text{cartesian coordinate system}}$

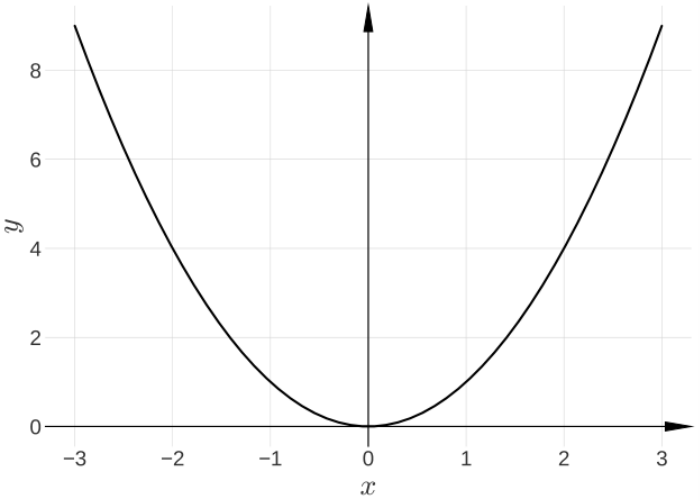


데카르트RENÉ DESCARTES(1596~1650)

함수의 표현: 그래프

- 입력과 출력의 관계를 만족시키는 점을 찍자
- 예: $y = x^2$

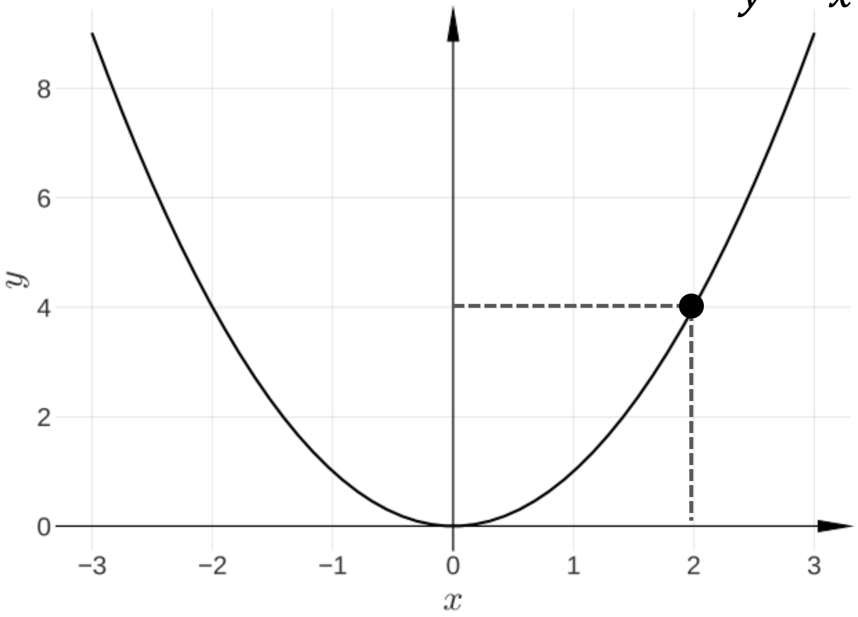




함수의 그래프와 대응관계

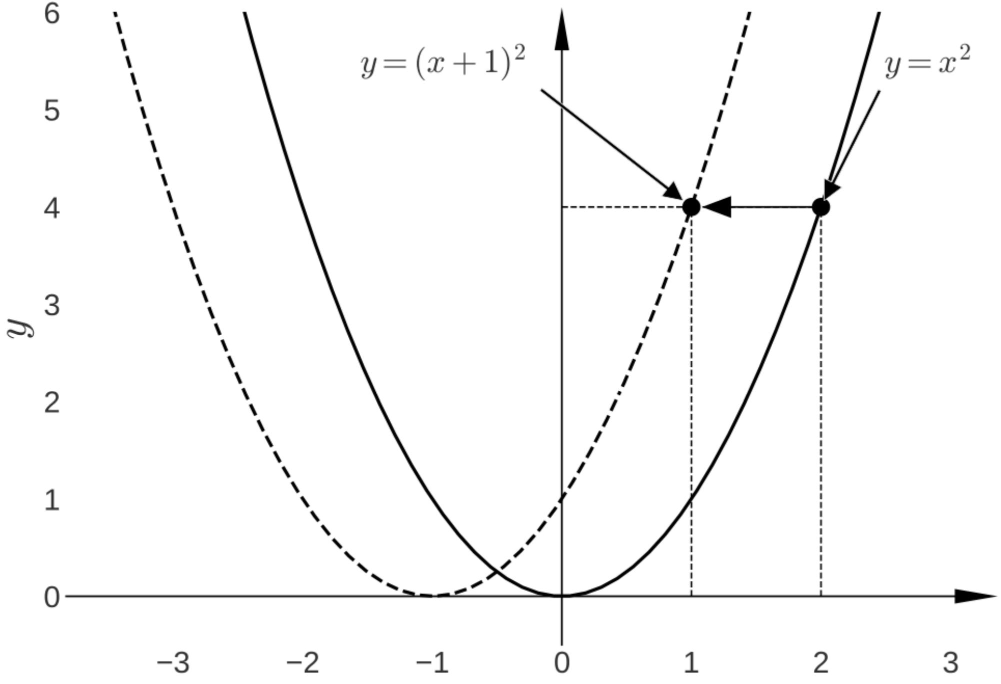
- 대응관계가 바뀌면 점들이 이동한다!

$y = x^2 \longrightarrow y = (x+1)^2$



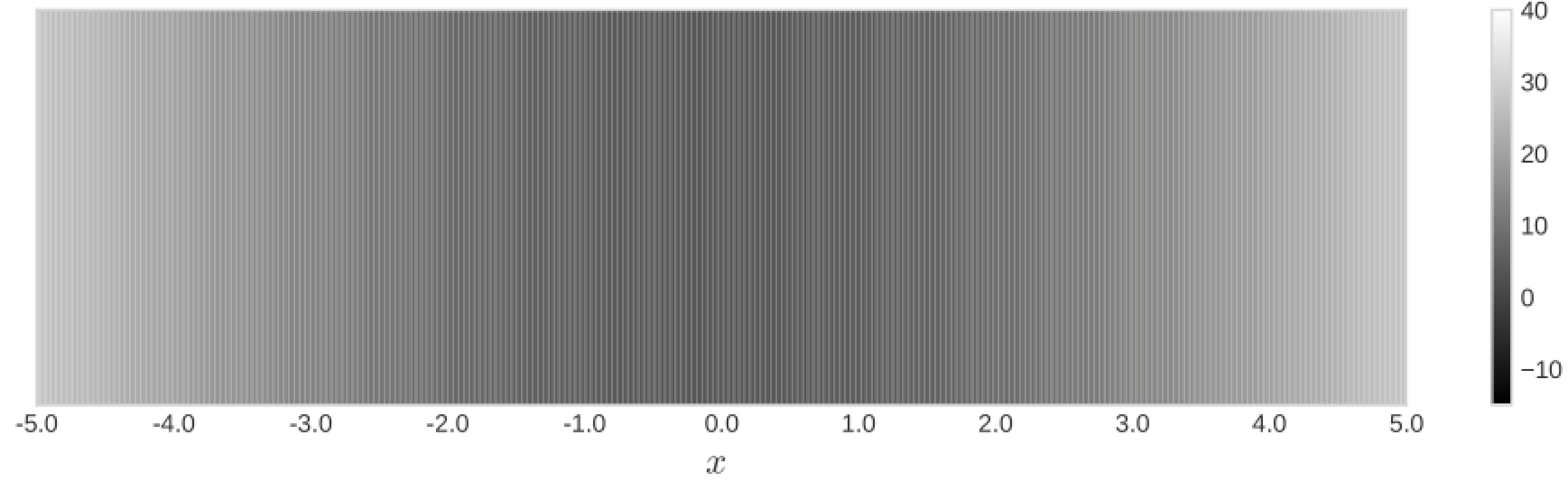
함수의 그래프와 대응관계

- 대응관계가 바뀌면 점들이 이동한다!



2차 곡선은 포물선?

- 꼭 포물선이어야 할까?
- 데카르트의 유산을 따르지 않는다면?
- 밝기로 나타낸 $y = x^2$



다항함수

- 단항식 : 2, x, x^2, $x/3$등 상수 또는 변수의 항 하나로 이루어진 식

- 다항식 : 단항식 여러 개의 더하기로 연결된 식
  - $3x^2 + 2$
  - 다항식의 차수는 가장 높은 차수

변수(variable) 차수(degree)

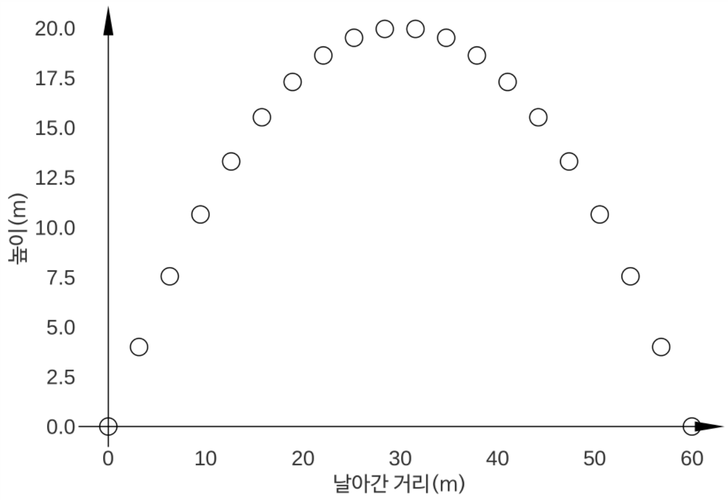
$$3x^2 + 2$$

계수(coefficient) 상수(costant)

- 다항함수 : 다항식으로 구성된 함수
  - $f(x) = x^2 + 2x + 3$

다항함수의 활용예

- 포탄이 날아가는 궤적을 아래처럼 사진 찍었다면…
- 이 궤적을 식으로 가지고 있을 수 있을까?



$$y = x(x-60)$$ (0,0), (60,0)

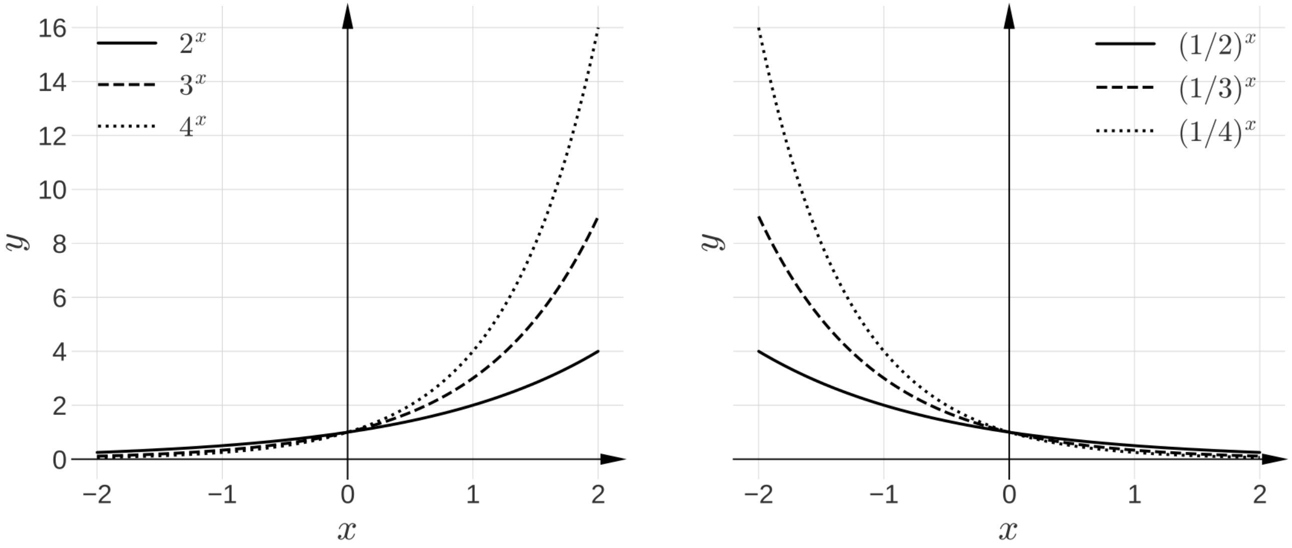
$$y = ax(x-60)$$ (30,20)

$$y = -\frac{1}{45}x(x-60)$$

- 이렇게 현상을 임의의 도구(함수)로 표현하는 수학 과정: 모델링

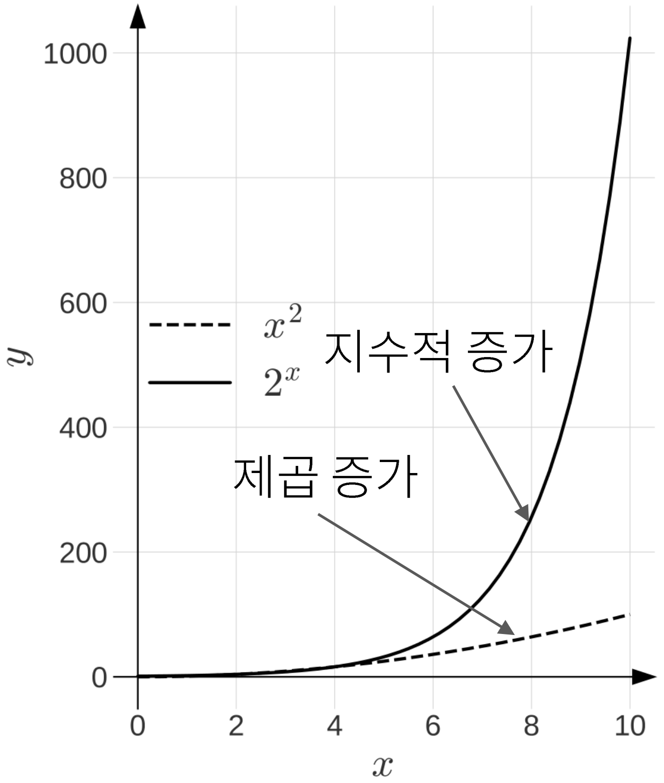
지수함수

- 정의 : $y = a^x, a > 0, a \neq 1$
- 그래프
  - $a > 1$: 양의 방향으로 갈 수록 증가
  - $0 < a < 1$: 양의 방향으로 갈 수록 감소



지수함수는 얼마나 빨리 증가하나?

- $y = 2^x$와 $y = x^2$ 비교
- *엄청 빨리 증가한다.*
- 알바 에피소드
  - A회사: 한 달간 일당 100만원!
  - B회사: 첫날 1원, 둘째 날 2원, 셋째 날 4원, 넷째 날 8원…
  - 어느 회사에서 알바를 해야? 🤔

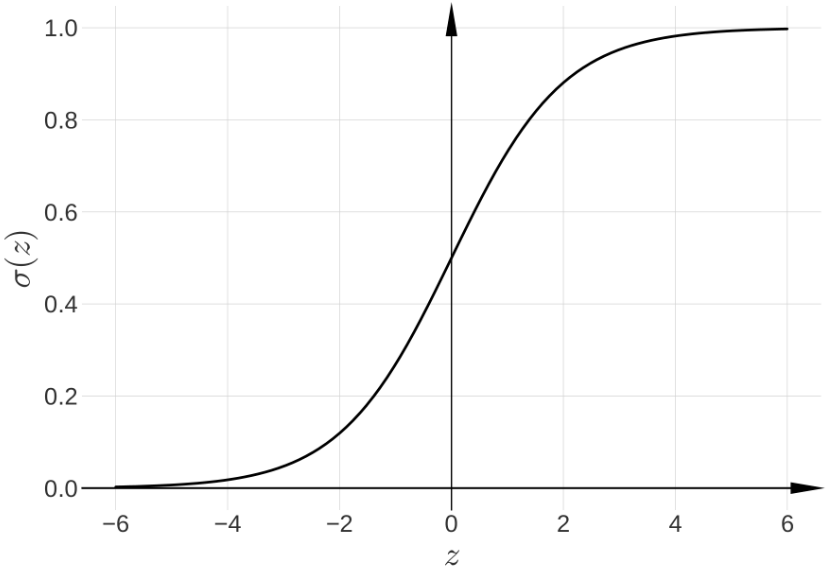


function.ipynb

Logistic Sigmoid Function

- 신경망에서 뉴런의 활성을 결정하는 활성함수로 사용

$$\sigma(z) = \frac{1}{1+e^{-z}}$$



- 모든 실수 z를 0~1사이로 찌그러트림
- 출력을 확률로 해석

로그log

- 로그는 숫자

$$2^{\square} = 16$$

A: 여기 뭐 들어가야 되나?
B: 4!

- 여기는?

$$2^{\square} = 17$$

B: 4보다 조금 큰? 5는 아니고..

- 2를 17로 만들기 위해 2의 어깨 위에 거듭 제곱 되어야 하는 숫자

$$\log_2 17 \qquad 2^{\log_2 17} = 17$$

로그log

- a를 x로 만들기 위해 a의 어깨 위에 거듭 제곱 되어야 하는 숫자

$$\log_a x = c$$

어떤 숫자를 올려야 하나?

$$a^c = x$$

이 숫자를

이 숫자로 만들고 싶은데

로그 성질: 1

- 정의를 잘 이용하면 다음을 보일 수 있다.

- $\log_a 1 = 0, \log_a a = 1, a^{\log_a 1} = 1, a^{\log_a a} = a$
- $\log_a MN = \log_a M + \log_a N$
  - 유도 $\log_a MN$은 a를 MN으로 만들기 위해 어깨 위에 올라가는 수

$$a^{\log_a MN} = MN$$

$$a^{\log_a M} = M \qquad a^{\log_a N} = N$$

$$a^{\log_a M} \times a^{\log_a N} = MN$$

$$a^{\log_a M + \log_a N} = MN$$

√NOTE

$$\log_a M^k = k \log_a M$$

로그 성질: 2

- $\log_a \frac{M}{N} = \log_a M - \log_a N$
  - 유도 로그 정의대로 써주면

$$a^{\log_a \frac{M}{N}} = \frac{M}{N}$$

$$a^{\log_a M} = M \qquad a^{\log_a N} = N$$

$$\frac{a^{\log_a M}}{a^{\log_a N}} = \frac{M}{N}$$

$$a^{\log_a M - \log_a N} = \frac{M}{N}$$

로그 성질: 3

- $\log_a b = \dfrac{\log_c b}{\log_c a}$
  - 유도 로그 정의대로 써주면

$$a^{\log_a b} = b \qquad \log_c\left(a^{\log_a b}\right) = \log_c b$$

$$\log_a b \log_c a = \log_c b \qquad \because \log_a M^k = k\log_a M$$

$$\log_a b = \frac{\log_c b}{\log_c a} \qquad \because \log_c a \neq 0$$

역함수Inverse function

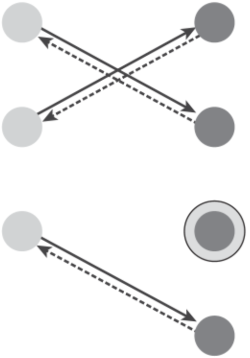
- 정의 : $f: X \rightarrow Y$일 때 $g: Y \rightarrow X$인 함수가 있어서 $f(x) = y$일 때 $g(y) = x$를 만족하는 함수

- 표기법 : $f^{-1}: Y \rightarrow X, x = f^{-1}(y)$

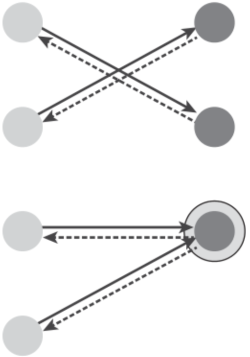
- 함수와 그의 역함수의 그래프는 $y = x$에 대칭

역함수Inverse func.

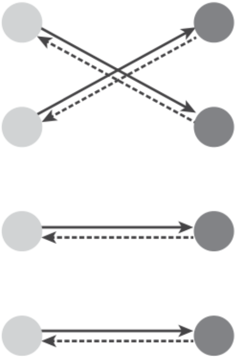
- 존재 : 일대일 대응에서만 존재



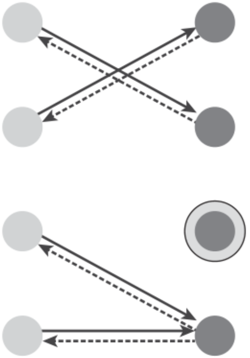
일대일 함수one-to-one
(단사함수 Injection)



전사 함수onto
(Surjection)



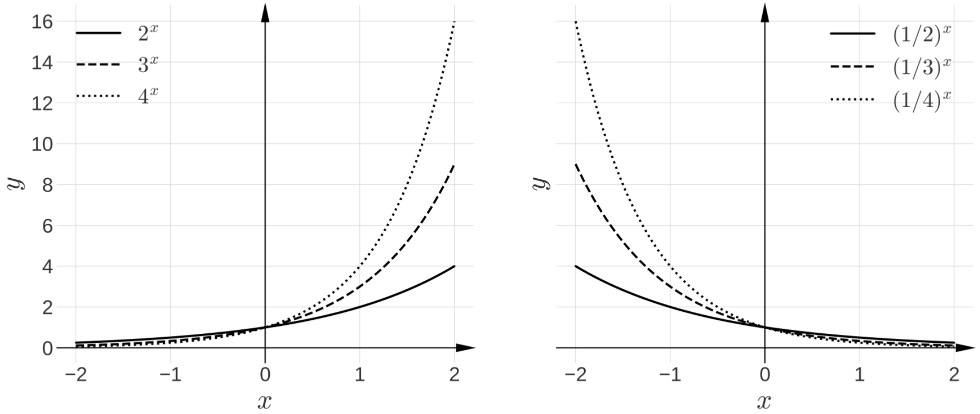
일대일 대응 함수
one-to-one correspondence
(전단사함수 Bijection)



함수
(Function)

로그함수

- 지수함수는 일대일 대응함수



입력

$$a^y = x$$

출력

- 지수함수의 역함수로써 로그함수

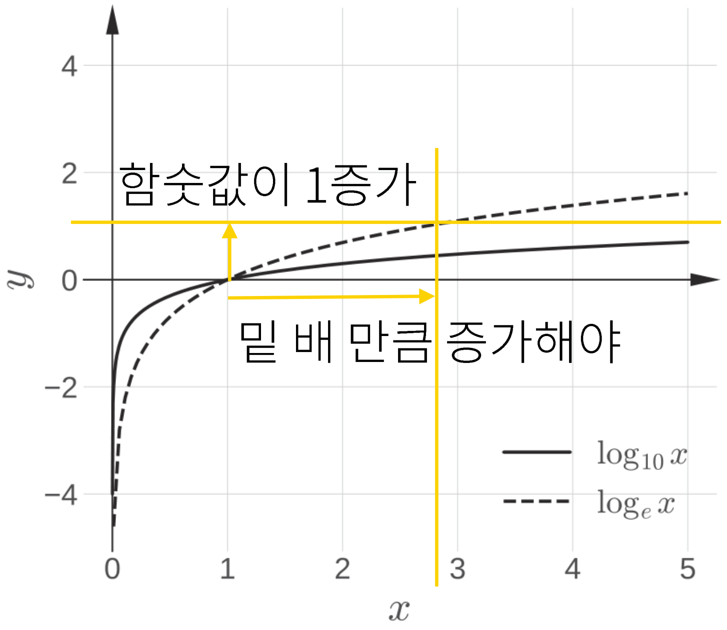
$$y = \log_a x \qquad a > 0,\ a \neq 1$$

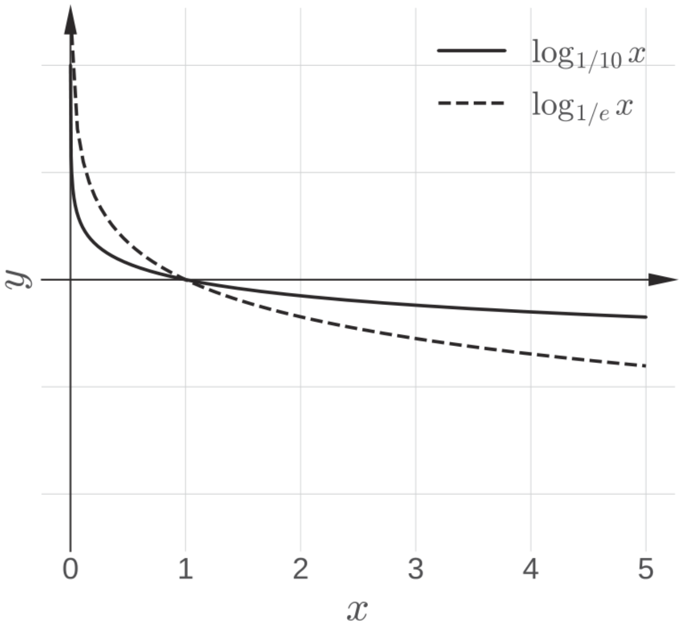
출력 입력

$$y = \log_a x$$

: 로그함수의 그래프

- 지수함수와 다르게 천천히 변하는 특성





: 로그의 활용: MSLE

- MSE: Mean Squared Error

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

- MSLE: Mean Squared Logarithmic Error

$$MSLE = \frac{1}{N}\sum_{i=1}^{N}(\log(\hat{y}_i + 1) - \log(y_i + 1))^2$$

```
gt_1 = 10
pred_1 = 15

gt_2 = 1000
pred_2 = 1005
```

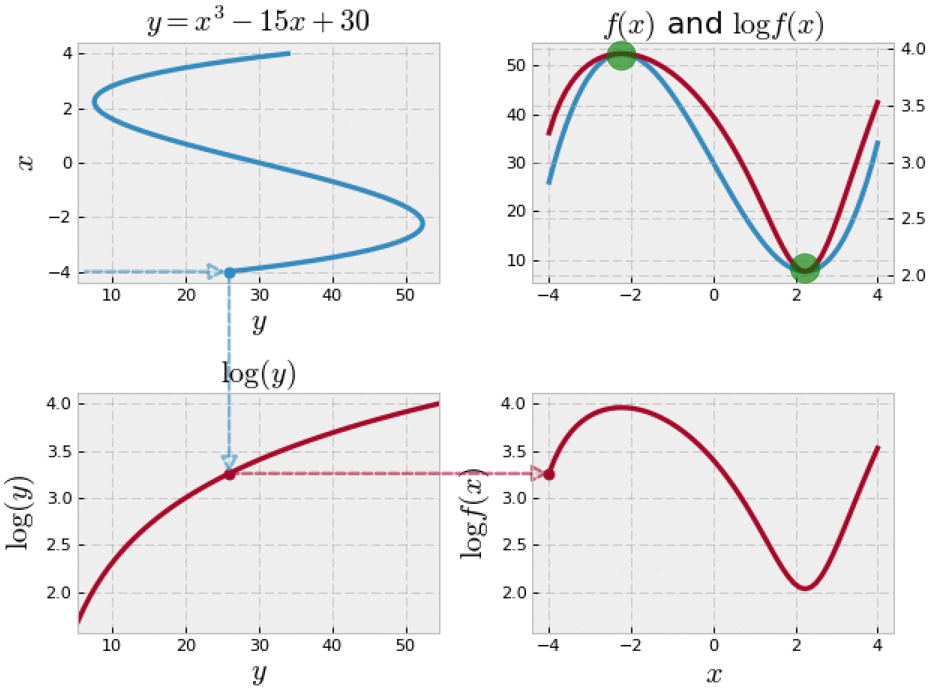
두 예측 모두 똑같이 5 틀린 것?

```
def mse(pred, gt):
    return (pred-gt)**2

def msle(pred, gt):
    return (np.log(pred+1) - np.log(gt+1))**2

print(f"mse 1:{mse(pred_1, gt_1):.6f}")
print(f"mse 2:{mse(pred_2, gt_2):.6f}")

print(f"msle 1:{msle(pred_1, gt_1):.6f}")
print(f"msle 2:{msle(pred_2, gt_2):.6f}")
```

CO function.ipynb

👀: 로그의 중요한 성질

- 볼록한 지점(극점)의 위치를 변화시키지 않는다.



스칼라와 벡터

- 스칼라Scalar :
  - Scale/skeil/ : ~을 자로 재다., 얼마만큼 자(저울)로 재다.
  - 양만으로 표시할 수 있는 물리량
  - 수직선상의 점 하나로 정의
  - 온도, 속력, 부피, 넓이
  - 숫자 하나, 1cm, 10°C, $20m^2$

- 벡터Vector
  - Vehere : 라틴어로 “운반하다”, 어떤 물체를 얼마만큼 어디로 운반하다.
  - 양과 방향으로 표현할 수 있는 물리량
  - 공간의 두 점 또는 원점과 한 점으로 정의
  - 속도, 힘
  - 숫자 여러 개, $\mathrm{v} = (1,2,3)^{\mathrm{T}}$

벡터 표기

- 볼드 소문자

x 스칼라 $\mathbf{x}$ 벡터

- 벡터는 특별한 언급이 없다면 기본으로 **열벡터**

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\mathbf{x} = (x_1, x_2)^{\mathrm{T}}$$

열벡터$_{\text{column vector}}$

$$\mathbf{x} = (x_1, x_2)$$

행벡터$_{\text{row vector}}$

함수의 종류

- 일반적인 함수의 종류
  - 다항함수, 분수함수, 지수함수, 로그함수, 삼각함수, ……
  - $f(x) = 3x + 2$, $f(x) = \frac{3x^2+2}{x+1}$, $f(x) = 2^x$, $f(x) = \log 3x$

- 입력과 출력의 형태에 따른 분류

일변수-스칼라함수
univariable scalar func.

다변수-스칼라함수
multivariable scalar func.

일변수-벡터함수
univariable vector func.

다변수-벡터함수
multivariable vector func.

일변수-스칼라함수

- $y = f(x), f: \mathbb{R} \rightarrow \mathbb{R}$

- 고교에서 많이 봤던 익히 알고 있는 함수
  - 다항함수, 분수함수, 지수함수, 로그함수, 삼각함수

- 입력: 스칼라, 출력 : 스칼라
  - 즉, 입력이 숫자 하나가 들어가고 출력으로 숫자 하나가 나온다.

- $f(x) = x^2, f(x) = 2^x, f(x) = \log 3x$

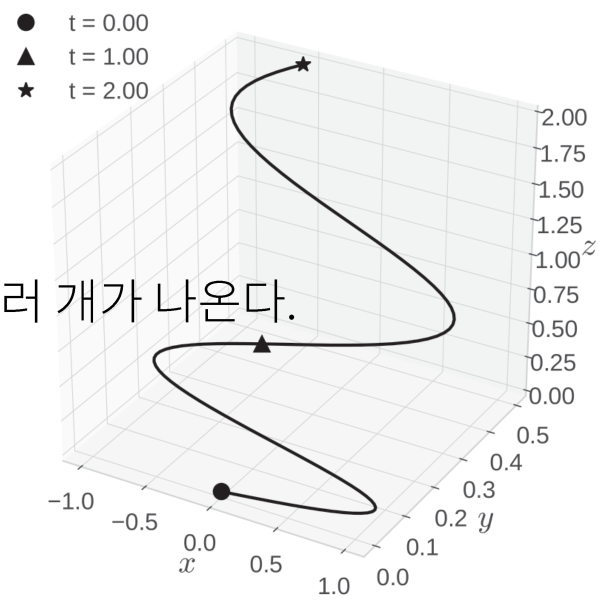
일변수-벡터함수

- $\mathrm{r}(t) = \left(f_1(t), f_2(t), \dots, f_n(t)\right)^{\mathrm{T}}, f: \mathbb{R} \to \mathbb{R}^n$

- 평면 또는 공간에 존재하는 곡선
  - 시간에 따른 물체의 이동 경로

- 입력: 스칼라, 출력 : 벡터
  - 즉, 입력이 숫자 하나가 들어가고 출력으로 숫자 여러 개가 나온다.

- $f(x(t), y(t), z(t)) = \left(\sin(6t), \frac{1}{4}t, \frac{t^2}{2}\right)^{\mathrm{T}}$



다변수-스칼라함수

- $y = f(\mathbf{x}), f: \mathbb{R}^n \rightarrow \mathbb{R}$

- 스칼라 장을 정의
  - 공간에서의 온도장, 대기의 기압장

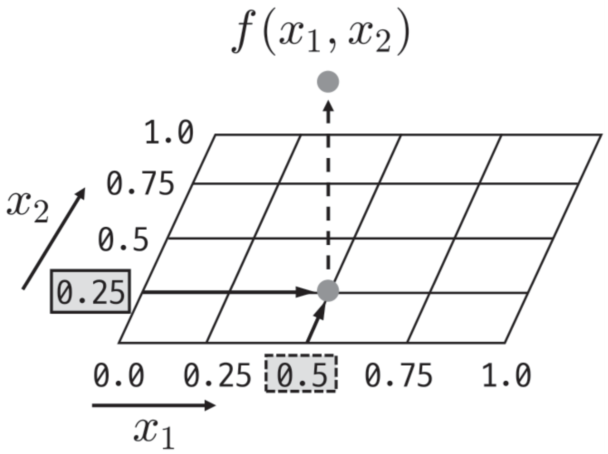
- 입력: 벡터, 출력 : 스칼라
  - 즉, 입력이 숫자 여러 개가 들어가고 출력으로 숫자 하나가 나온다.

- $f(x, y) = x + y, f: \mathbb{R}^2 \rightarrow \mathbb{R}$

다변수 스칼라함수의 그래프

- 2변수 실함수 그래프 그리기. $z = f(x_1, x_2), f: \mathbb{R}^2 \to \mathbb{R}$
- numpy.meshgrid을 이용해서 xy평면에 적절히 그리드를 생성

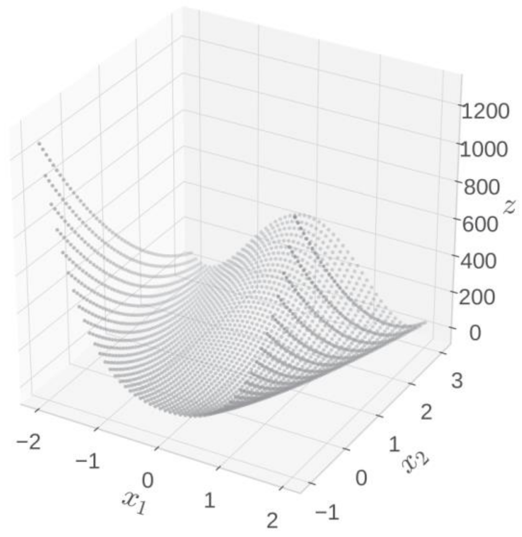
```
x1 = np.linspace(0, 1, 5)
X2 = np.linspace(0, 1, 5)
X1, X2 = np.meshgrid(x1, x2)

print(X1)
[[ 0.    0.25  0.5   0.75  1.  ]
 [ 0.    0.25  0.5   0.75  1.  ]
 [ 0.    0.25  0.5   0.75  1.  ]
 [ 0.    0.25  0.5   0.75  1.  ]
 [ 0.    0.25  0.5   0.75  1.  ]]

print(X2)
[[ 0.    0.    0.    0.    0.  ]
 [ 0.25  0.25  0.25  0.25  0.25]
 [ 0.5   0.5   0.5   0.5   0.5 ]
 [ 0.75  0.75  0.75  0.75  0.75]
 [ 1.    1.    1.    1.    1.  ]]
```



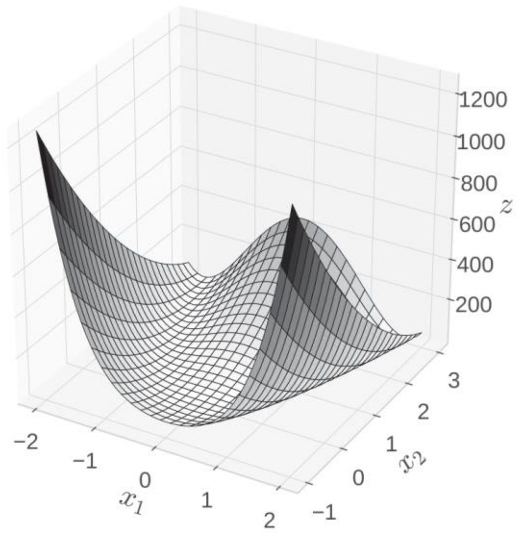
다변수 스칼라함수의 그래프

```
x1 = np.linspace(-2, 2, 50)
x2 = np.linspace(-1, 3, 50)
X1, X2 = np.meshgrid(x1, x2)
Z = 50*(X2 - X1**2)**2 + (2-X1)**2
```

function.ipynb



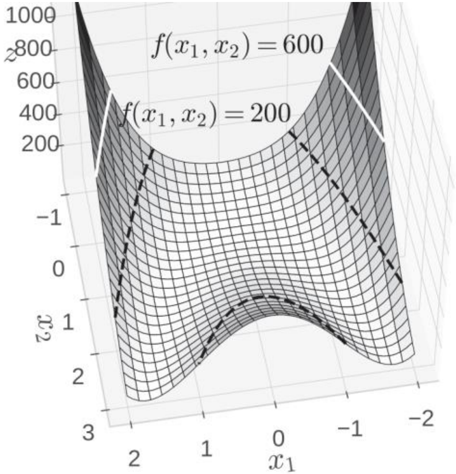
ax.scatter3D(X1, X2, Z, marker='.’)



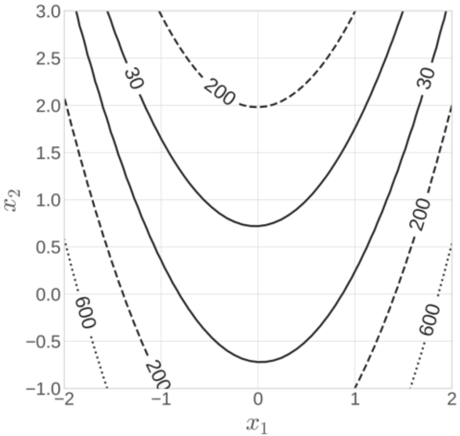
ax.plot_surface(X1, X2, Z)

한쪽 변이 숫자로 고정?

- 음함수$_{\text{implicit function}}$ 형태 $f(x, y) = c$
- 이변수 스칼라함수 $f(x_1, x_2) = c$, 등고선 그래프



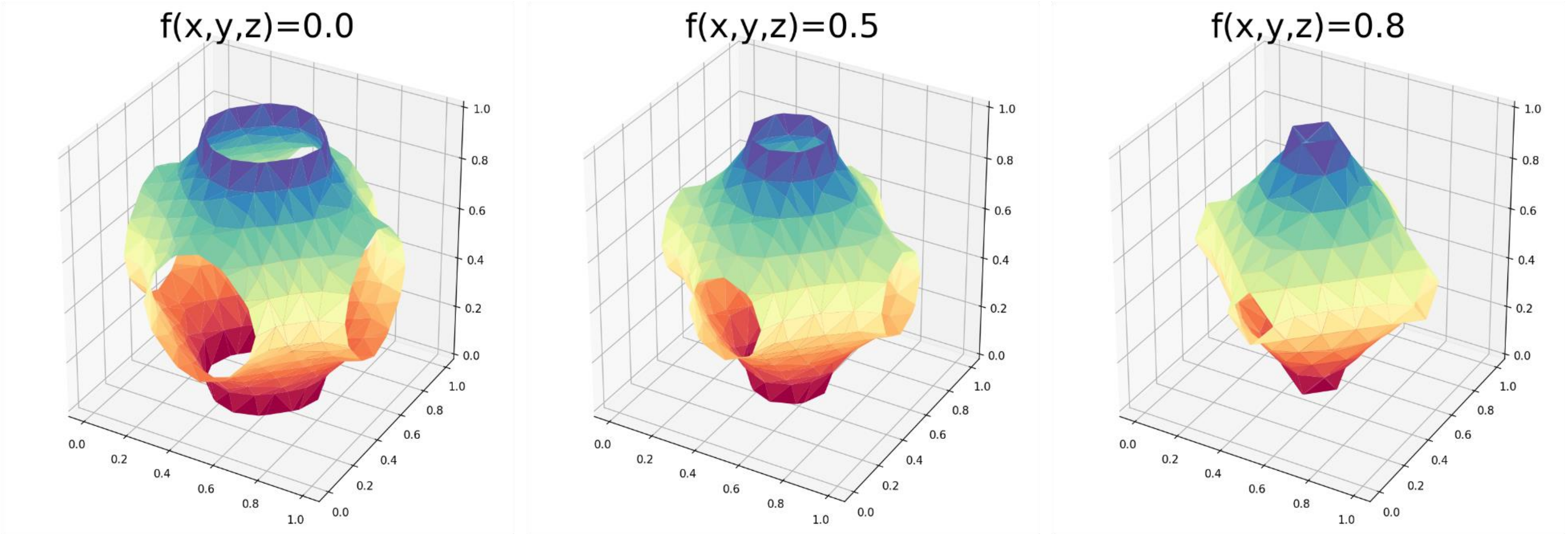
특정 c로 생성되는
등고선을 평면에 모으면



- 삼변수 실함수의 경우 $w = f(x, y, z)$ 형태는 그래프로 그릴 수 없음
  - $f(x, y, z) = c$ 는 3차원 공간에서 곡면

변수 3개?

- $f(x, y, z)$: 3차원 좌표축에 그림 그리기 불가능
- $f(x, y, z) = c$: 함숫값이 동일한 곡면, iso-surface



👀: 평면과 직선의 방정식

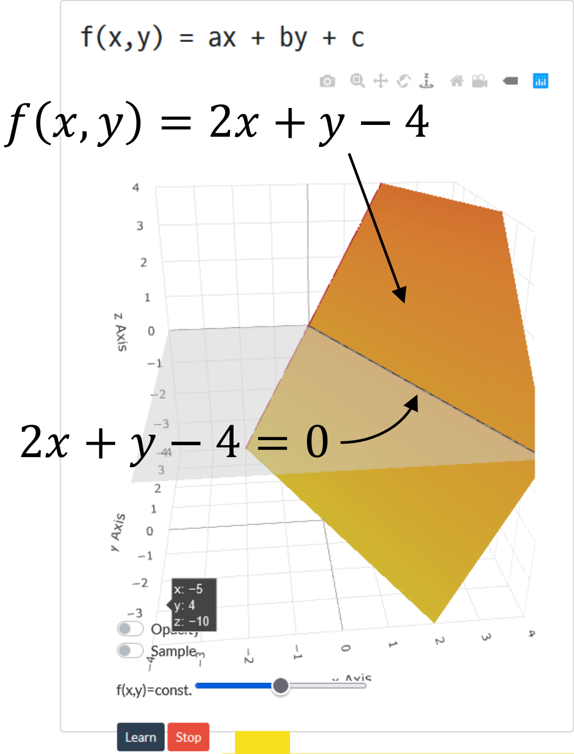
평면: 이변수 스칼라 함수

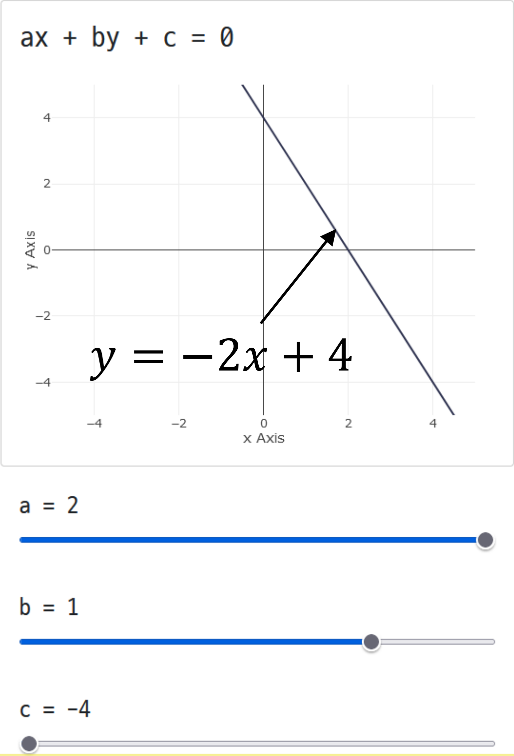
$f(x,y) = 2x + y - 4$

직선: 우변이 0으로 고정

$2x + y - 4 = 0$

$y = -2x + 4$

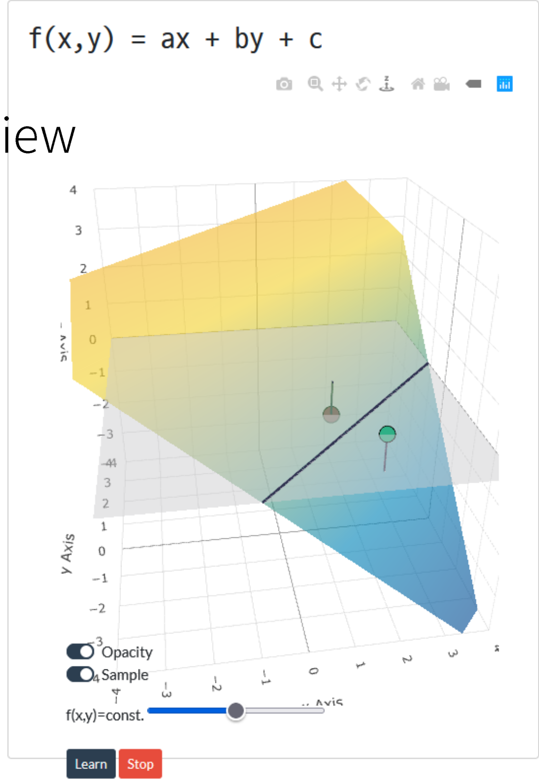


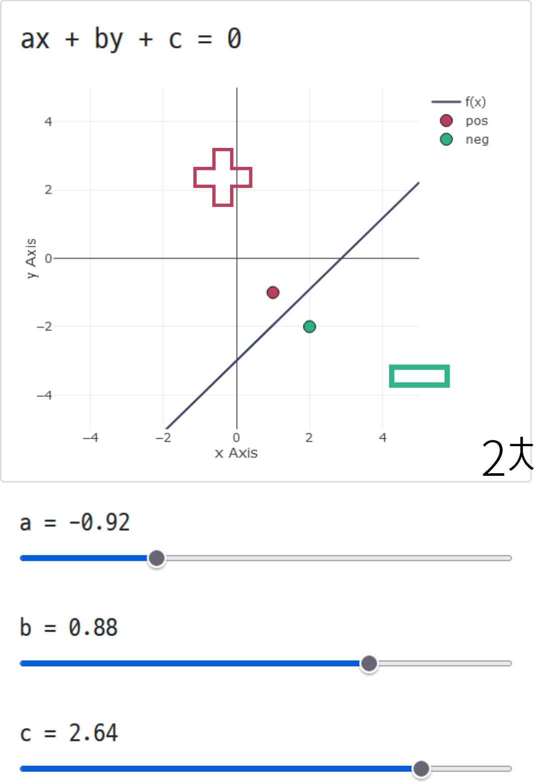


JS https://metamath1.github.io/noviceml/plane.html

👀: 평면을 이용한 분류

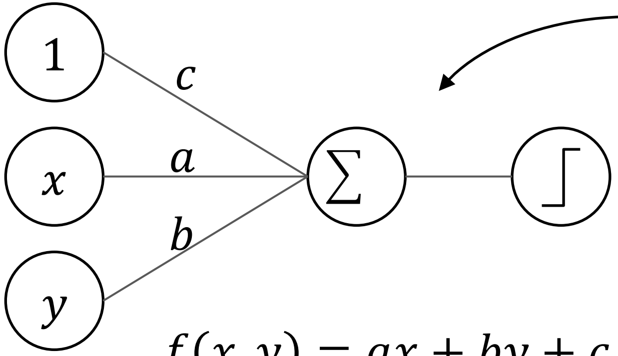
3차원 view



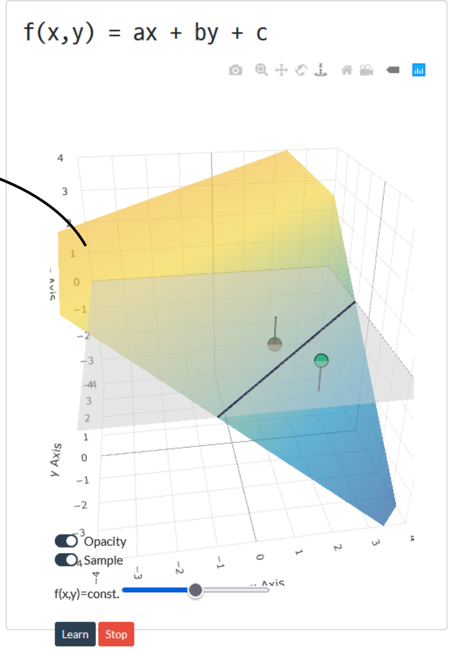


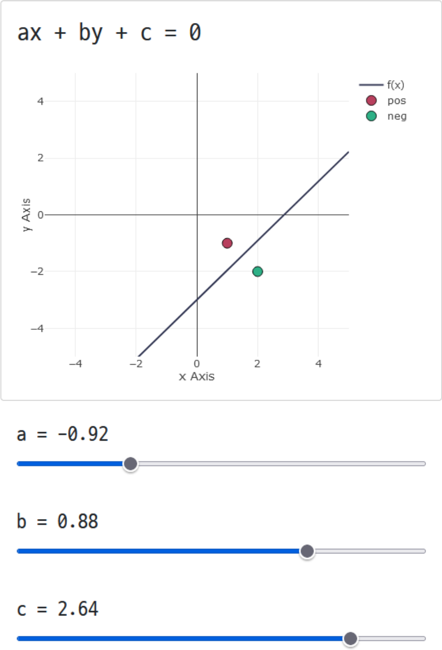
2차원 view

👀: 퍼셉트론preceptron

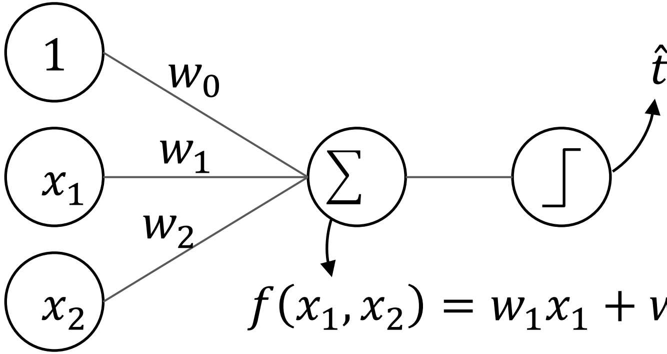


$$f(x, y) = ax + by + c$$





👀: 퍼셉트론 학습규칙



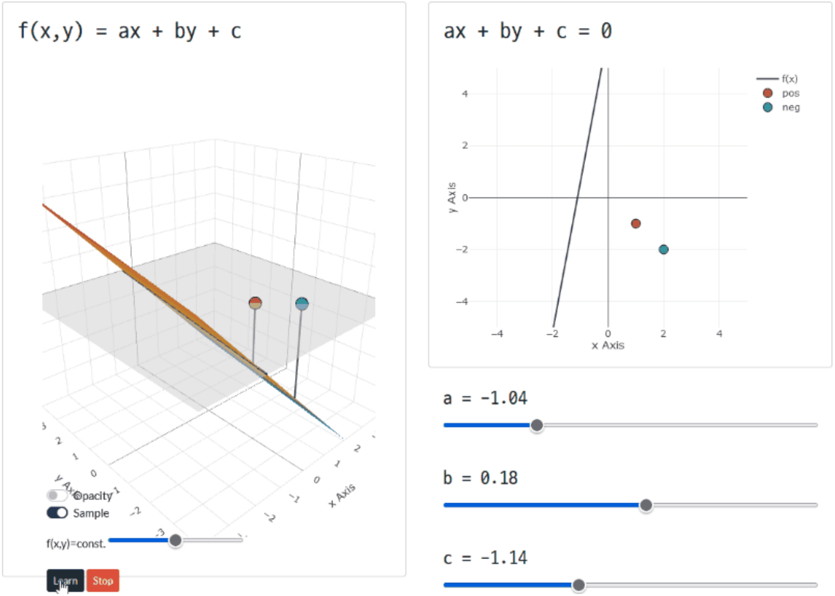
$$\hat{t} = \begin{cases} 1 & \text{if } f(x_1, x_2) \geq 0 \\ -1 & \text{else} \end{cases}$$

$$f(x_1, x_2) = w_1 x_1 + w_2 x_2 + w_o$$

Update rule

$$\Delta w_j = \eta(t - \hat{t})x_j$$

$$w_j = w_j + \Delta w_j$$



https://metamath1.github.io/2017/05/12/perceptron-conv-theorem.html

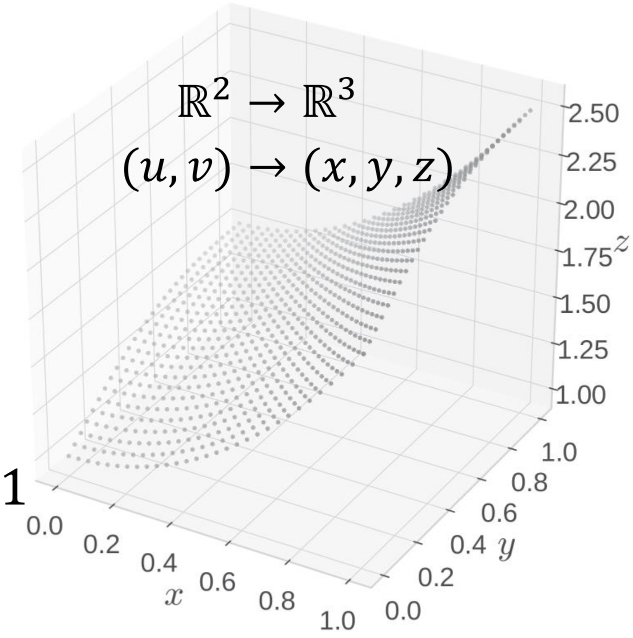
다변수 벡터 함수

- $\mathbf{F}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})\right)^T, \mathrm{x} \in \mathbb{R}^m$

- $f_i(\mathbf{x})\colon \mathbb{R}^m \to \mathbb{R}$

- 입력: 벡터, 출력 : 벡터
  - 입력이 숫자 여러 개가 들어가고
    출력으로 숫자 여러 개가 나온다

- $\mathbf{s}(u, v) = \left(u, v, 1 + u^2 + \frac{v}{1+v^2}\right)^{\mathrm{T}}, 0 \le u, v \le 1$



다변수-벡터함수의 활용

- $softmax: \mathbb{R}^K \rightarrow [0,1]^K$: 실수 k개가 0에서 1사이의 숫자 k개로 매핑

Scores	Softmax	Probabilities
$\boldsymbol{z}$		$\boldsymbol{s}$
1.0 →		→ 0.0641
2.5 →	$softmax_j(\mathbf{z}) = \frac{\exp(z_j)}{\sum_{k=1}^{K} \exp(z_k)}$	→ 0.2875
3.0 →		→ 0.474
2.0 →		→ 0.1744
		+
		1.0

함수의 합성Composite Function

- 정의 : 함수 $f: X \to Y$의 공역과 함수 $g: Y \to Z$의 정의역이 같다고 할 때 다음과 같이 정의된 함수 $g \circ f$를 두 함수 f와 g의 합성이라고 한다.

- 표기법 : $g(f(x))$: 직관적, $g \circ f$: 덜 직관적

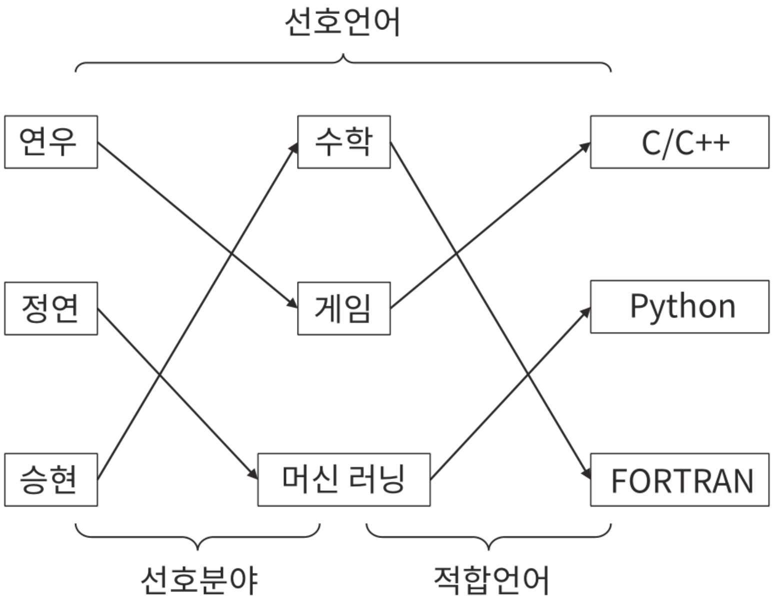
- 입출력 관점 : f의 출력이 g의 입력으로 들어감

"선호분야"와 "적합언어 " 가
합성된 새로운 관계

선호언어 적합언어 아이 이름

$$g \circ f = g(f(x))$$

선호분야

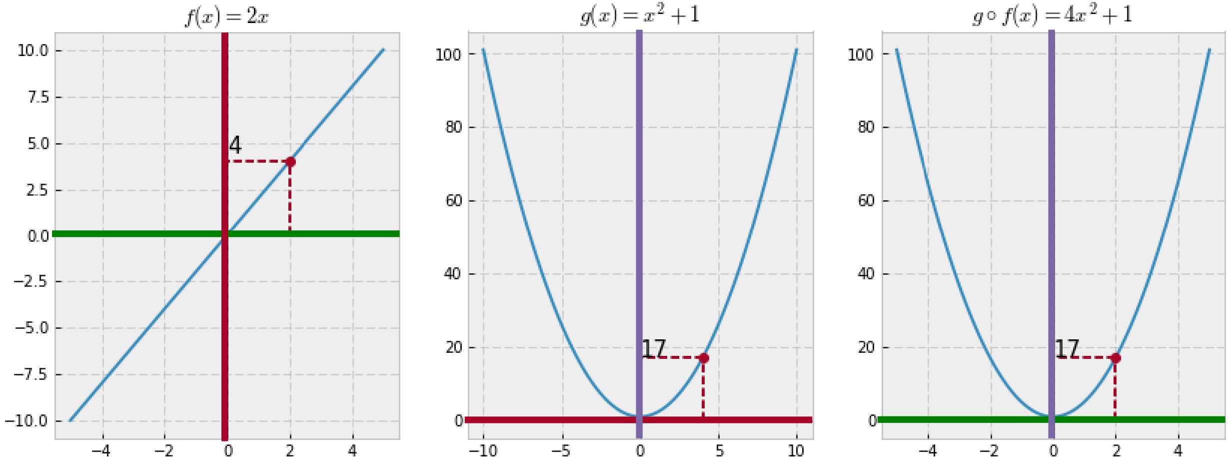


함수의 합성Composite Function

- 정의 : 함수 $f: X \rightarrow Y$의 공역과 함수 $g: Y \rightarrow Z$의 정의역이 같다고 할 때 다음과 같이 정의된 함수 $g \circ f$를 두 함수 f와 g의 합성이라고 한다.

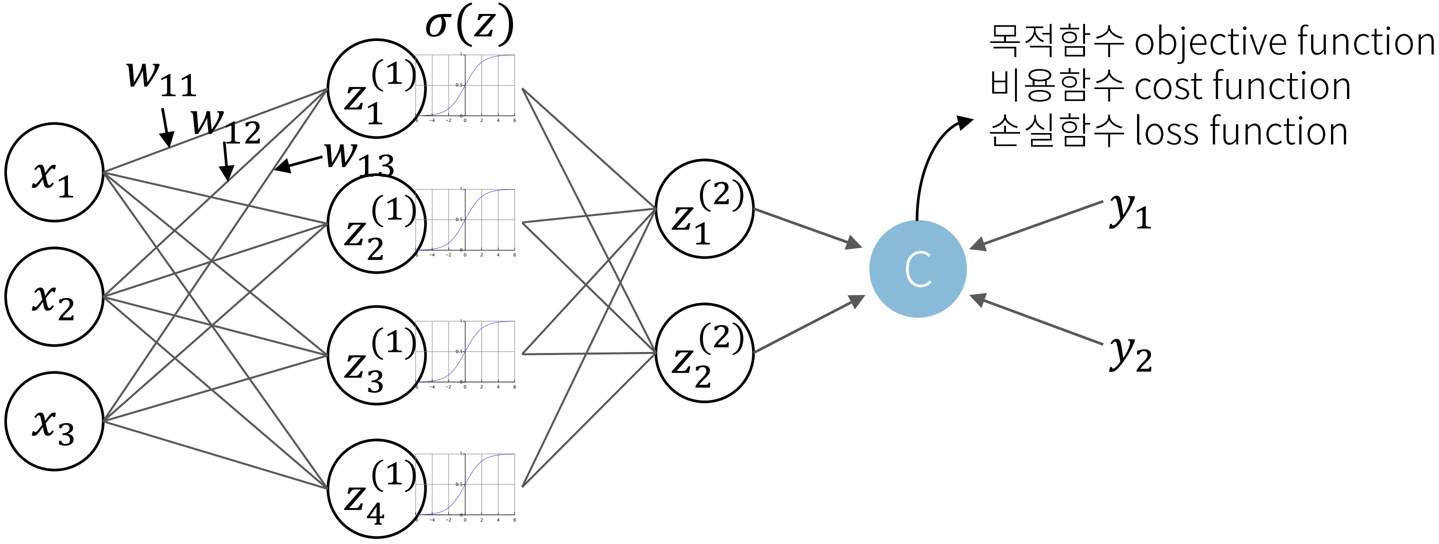
- 표기법 : $g(f(x))$: 직관적, $g \circ f$: 덜 직관적

- 입출력 관점 : f의 출력이 g의 입력으로 들어감



👀: 인공신경망은 합성함수

- 신경망은 매우 많은 함수가 다음처럼 겹겹이 합성된 것이라 할 수 있다.

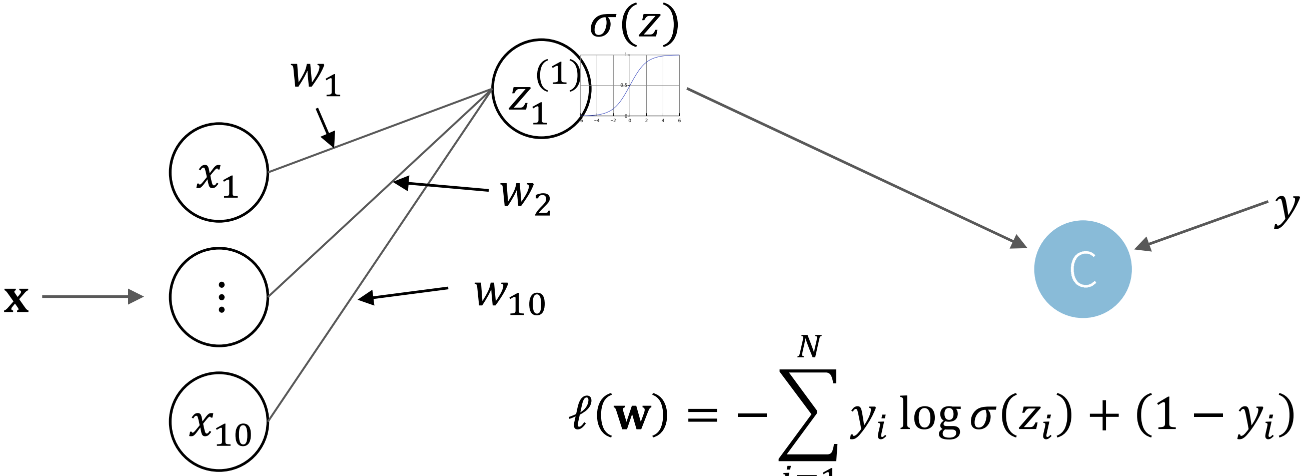


$f_1 : \mathbb{R}^3 \to \mathbb{R}^4$ $\quad$ $f_2 : \mathbb{R}^4 \to \mathbb{R}^2$ $\quad$ $f_3 : \mathbb{R}^2 \to \mathbb{R}$

$$f : \mathbb{R}^3 \longrightarrow \mathbb{R}^2$$

가장 간단한 형태

- 로지스틱 회귀



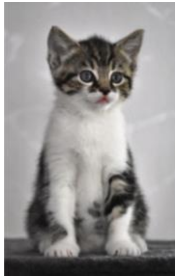
$$\ell(\mathbf{w}) = -\sum_{i=1}^{N} y_i \log \sigma(z_i) + (1 - y_i) \log(1 - \sigma(z_i))$$

$f_1: \mathbb{R}^{10} \to \mathbb{R}$

$f_2: \mathbb{R} \to \mathbb{R}$

초간단 분류기: 지수,로그함수활용

- 머신러닝 분류 문제에 있어서 평가 함수로 사용



0

1

$\mathbf{x} \in \mathbb{R}^n$

w_1
w_2
w_3
w_4
w_5
w_6
w_7
w_8
w_9
w_{10}

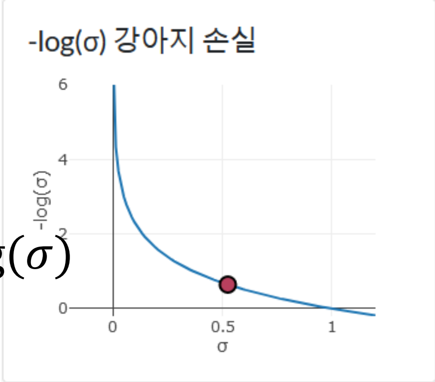
$\mathbf{w}$

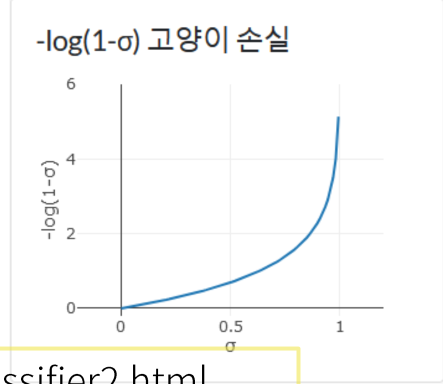
$z: \mathbb{R}^{10} \to \mathbb{R}$ ⇒ $\sigma(z) \in (0,1)$ ⇒ $-\log(\sigma)$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

강아지

강아지 확률:0.525, 고양이 확률: 0.475

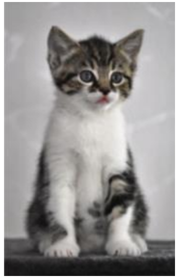




JS metamath1.github.io/noviceml/toyclassifier2.html

초간단 분류기: 지수,로그함수활용

- 머신러닝 분류 문제에 있어서 평가 함수로 사용



0

1

$\mathbf{x} \in \mathbb{R}^n$

w_1, $\boxed{w_2 = 2}$, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10}

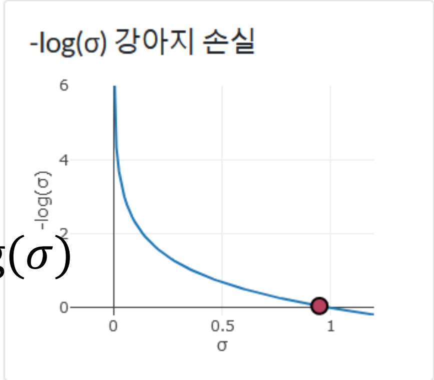
$\mathbf{w}$

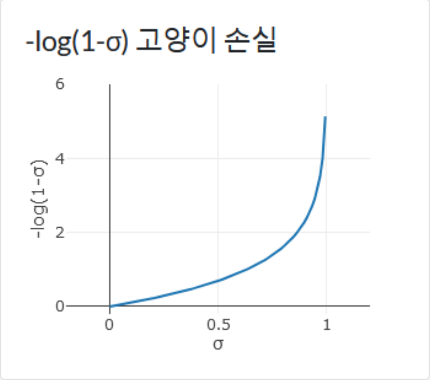
$z: \mathbb{R}^n \to \mathbb{R} \Rightarrow \sigma(z) \in (0,1) \Rightarrow -\log(\sigma)$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

강아지

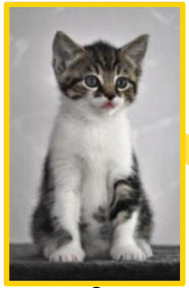
강아지 확률:0.949, 고양이 확률: 0.051





초간단 분류기: 지수,로그함수활용

- 머신러닝 분류 문제에 있어서 평가 함수로 사용



0

1

$\mathbf{x} \in \mathbb{R}^n$

w_1, $\boxed{w_2 = 2}$, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10}

$\mathbf{w}$

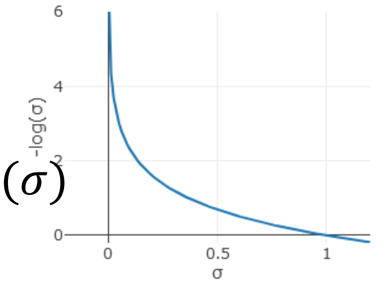
$z: \mathbb{R}^n \rightarrow \mathbb{R}$ ⇒ $\sigma(z) \in (0,1)$ ⇒ $-\log(\sigma)$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

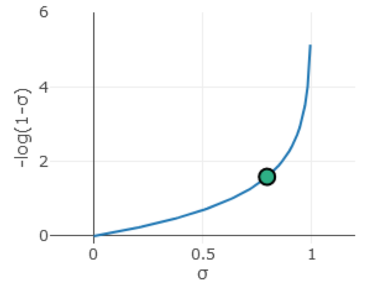
강아지

강아지 확률:0.796, 고양이 확률: 0.204

-log(σ) 강아지 손실

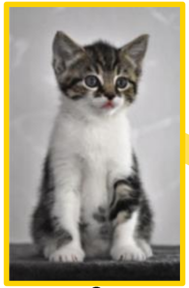


-log(1-σ) 고양이 손실



초간단 분류기: 지수,로그함수활용

- 머신러닝 분류 문제에 있어서 평가 함수로 사용



0

1

$\mathbf{x} \in \mathbb{R}^n$

w_1

$\boxed{w_2 = 2}$

w_3

w_4

w_5

w_6

w_7

$\boxed{w_8 = -4}$

w_9

w_{10}

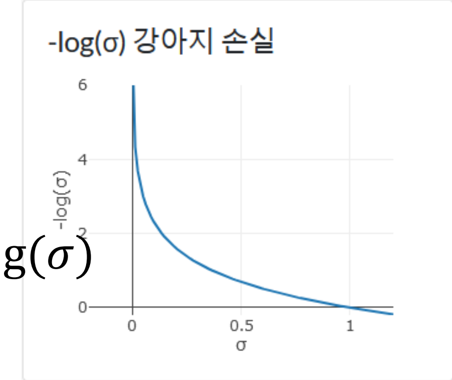
$\mathbf{w}$

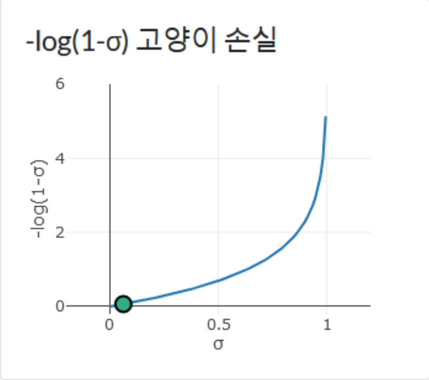
$z: \mathbb{R}^n \to \mathbb{R}$ ⇒ $\sigma(z) \in (0,1)$ ⇒ $-\log(\sigma)$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

강아지

강아지 확률:0.949, 고양이 확률: 0.051





인공지능 이산수학

행렬

조준우
metamath@gmail.com

행렬Matrix

- 사각 괄호로 둘러 쌓인 숫자들의 배열(Rectangular array of numbers written between square brackets)

$$\begin{bmatrix} 0.3 & 1 & -5 \\ 0 & -0.2 & 16 \end{bmatrix}$$

요소entries, elements

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

행rows

열columns

행렬Matrix

- 볼드 대문자 표시
- 행렬의 차원(크기) : 행 개수 × 열 개수

$$\mathbf{A} = [a_{jk}] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

- 보기의 경우 $m \times n$ 행렬
- 요소의 첫 번째 인덱스가 행 번호, 두 번째 인덱스가 열 번호

행렬의 덧셈과 스칼라 곱셈

- 행렬 $\mathbf{A}$, $\mathbf{B}$의 크기가 같을 때 같은 위치의 요소의 덧셈

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} \qquad \mathbf{A} + \mathbf{B} = \begin{bmatrix} 11 & 22 \\ 33 & 44 \end{bmatrix}$$

- 크기가 다르면 덧셈 불가능

- 행렬에 대한 스칼라 곱은 각 요소에 스칼라를 곱하는 것

$$\mathbf{B} = \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} \qquad \mathbf{B} \times \mathbf{3} = \begin{bmatrix} 30 & 60 \\ 90 & 120 \end{bmatrix}$$

행렬-행렬의 곱셈

- 일반적인 설명: 행과 열의 요소를 곱해서 더함.

$$C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \boxed{a_{i1} \quad a_{i2} \quad \cdots \quad a_{in}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1j} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2j} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nj} & \cdots & b_{np} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1j} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2j} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{i1} & c_{i2} & \cdots & \boxed{c_{ij}} & \cdots & c_{ip} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mj} & \cdots & c_{mp} \end{bmatrix}$$

$m \times n$ matrix $\qquad$ $n \times p$ matrix $\qquad$ $m \times p$ matrix

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj}$$

행렬의 전치|Transpose

- 주어진 행렬의 행을 열로, 열을 행으로 이동
- $\mathbf{A}^{\mathrm{T}}$로 표시
- $(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \qquad A^{\mathrm{T}} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

```
import numpy as np

A = np.arange(1, 7).reshape(2, 3)
A
#>>> array([[1, 2, 3],
            [4, 5, 6]])
```

```
A.transpose(1, 0)
#>>> array([[1, 4],
            [2, 5],
            [3, 6]])

A.T #❸
#>>> array([[1, 4],
            [2, 5],
            [3, 6]])
```

특별한 행렬

- 단위행렬unit, identity matrix
  - 대각 성분이 모두 1인 $n \times n$ 정사각 행렬, $\mathbf{I}_\mathrm{n}, \mathbf{I}$
  - $\mathbf{AI} = \mathbf{IA} = \mathbf{A}$

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- 대각행렬diagonal matrix
  - 대각 성분만 0이 아닌 성분을 가진 $n \times n$ 정사각 행렬

$$\mathbf{D} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

특별한 행렬

- 대칭행렬symmetric matrix
  - 정사각 행렬에 대해서 $\mathbf{S}^{\mathrm{T}} = \mathbf{S}$, 따라서 $s_{kj} = s_{jk}$인 행렬

$$\mathbf{S} = \begin{bmatrix} 10 & 20 & 100 \\ 20 & 10 & 120 \\ 100 & 120 & 20 \end{bmatrix}$$

- 직교행렬orthogonal matrix
  - 전치된 것이 자신의 역행렬 $\mathbf{A}^{\mathrm{T}} = \mathbf{A}^{-1}$

$$\mathbf{A} = \begin{bmatrix} 1/3 & 2/3 & -2/3 \\ -2/3 & 2/3 & 1/3 \\ 2/3 & 1/3 & 2/3 \end{bmatrix}$$

행렬을 이용한 데이터 표현
테이블형 데이터

- 신체 특징 데이터: 키, 몸무게, 혈압, ⋯
- 한 명 데이터를 벡터로 표시: $\mathbf{x} = (\text{키}, \text{몸무게}, \text{혈압}, \dots)^T$
- 개별 데이터를 행 벡터로 저장
- 데이터 행렬 $\mathbf{X}$:(N,D), N: 데이터 개수, D: 데이터 차원

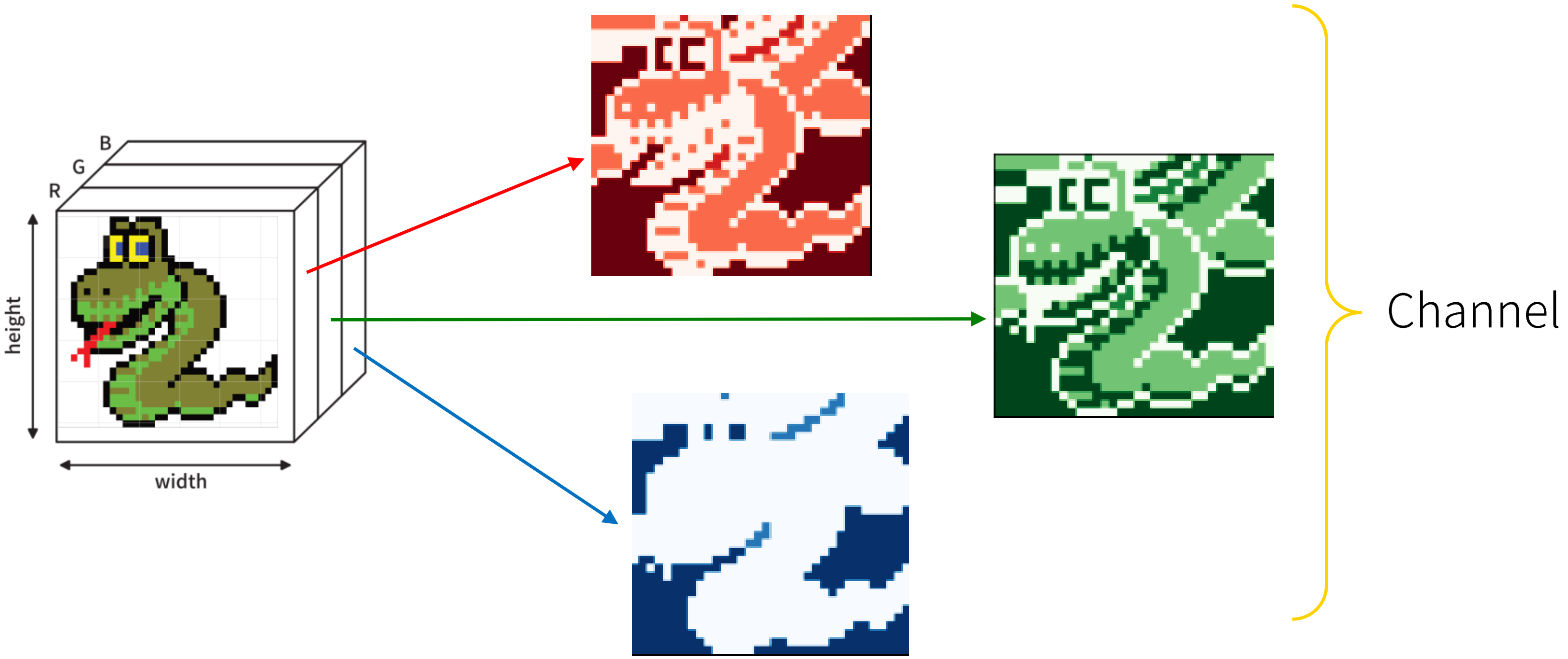
	키	몸무게	혈압	⋯
$\mathbf{x}_1$	170	64	90/100	...
$\mathbf{x}_2$	172	70	120/130	...
...	...	...	...	...
$\mathbf{x}_N$	182	82	100/120	...

개별 데이터

키 벡터

: 행렬을 이용한 데이터 표현: 이미지

- Channel first: (C, H, W), Channel last: (H, W, C)



: 행렬 곱셈을 이해하는 필살기

- 행렬 곱은
  - 뒤에서 곱하는 행렬의 각 열의 요소를 계수로 앞 행렬의 모든 열을 선형결합linear combination
  - 앞에서 곱하는 행렬의 각 행의 요소를 계수로 뒤 행렬의 모든 행을 선형결합
  - 앞에서 곱하는 행렬의 열과 뒤에서 곱하는 행렬의 행을 외적한 것들의 합

행렬 곱셈: 열 결합

- 뒤에서 곱하는 벡터의 요소를 계수로 한 모든 열의 선형결합linear combination

$$\mathbf{Ab} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_{11}b_1 + a_{12}b_2 + a_{13}b_3 \\ a_{21}b_1 + a_{22}b_2 + a_{23}b_3 \\ a_{31}b_1 + a_{32}b_2 + a_{33}b_3 \\ a_{41}b_1 + a_{42}b_2 + a_{43}b_3 \end{bmatrix}$$

$\times \quad \times \quad \times \quad +$

$$\begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 b_1 + \mathbf{a}_2 b_2 + \mathbf{a}_3 b_3 \end{bmatrix}$$

다시 말해 앞 행렬의 열을 벡터 요소로 가지는 1 행 짜리 행렬과 벡터의 곱으로 생각

행렬 곱셈: 열 결합

- 뒤에서 행렬을 곱해도 똑같은 방식이 성립

요소가 열 벡터인 1행 짜리 행렬

결과 행렬의 둘째 요소는
A행렬 열벡터의 선형결합;
계수 뒷 행렬 둘째 열 요소들

$$\mathbf{AB} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{a}_1 b_{11} + \mathbf{a}_2 b_{21} + \mathbf{a}_3 b_{31} & \mathbf{a}_1 b_{12} + \mathbf{a}_2 b_{22} + \mathbf{a}_3 b_{32} \end{bmatrix}$$

결과 행렬의 첫 요소는
A행렬 열벡터의 선형결합; 계수 뒷 행렬 첫 열 요소들

행렬 곱셈: 행 결합

- 앞에서 곱하는 행렬의 행 요소를 계수로
- 뒷 형렬의 행을 조합하여 결과의 행을 계산
- 열 결합 방식을 전치시키면 성립

$$(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} = \begin{bmatrix} b_{11} & b_{21} & b_{31} \\ b_{12} & b_{22} & b_{32} \end{bmatrix} \begin{bmatrix} \mathbf{a}_1^{\mathrm{T}} \\ \mathbf{a}_2^{\mathrm{T}} \\ \mathbf{a}_3^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} b_{11}\mathbf{a}_1^{\mathrm{T}} + b_{21}\mathbf{a}_2^{\mathrm{T}} + b_{31}\mathbf{a}_3^{\mathrm{T}} \\ b_{12}\mathbf{a}_1^{\mathrm{T}} + b_{22}\mathbf{a}_2^{\mathrm{T}} + b_{32}\mathbf{a}_3^{\mathrm{T}} \end{bmatrix}$$

matrix.ipynb

행렬 곱셈: 외적 합

- 앞, 뒤 행렬을 모두 벡터 요소로 표현하면

$$\mathbf{AB} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix} \begin{bmatrix} \mathbf{b}_1^{\mathrm{T}} \\ \mathbf{b}_2^{\mathrm{T}} \\ \mathbf{b}_3^{\mathrm{T}} \end{bmatrix}$$

$$= \mathbf{a}_1\mathbf{b}_1^{\mathrm{T}} + \mathbf{a}_2\mathbf{b}_2^{\mathrm{T}} + \mathbf{a}_3\mathbf{b}_3^{\mathrm{T}}$$

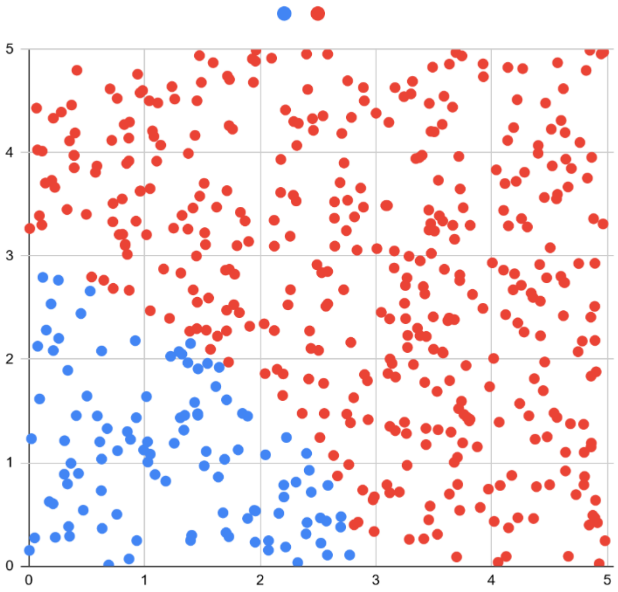
앞 행렬의 열벡터와
뒷 행렬의 행벡터들 간의 곱의 합

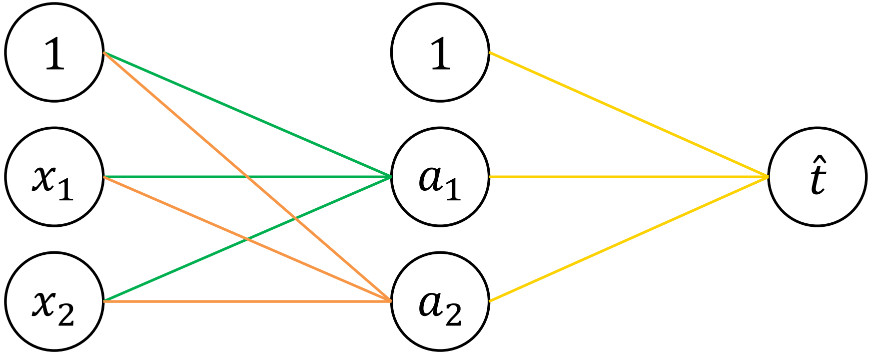
벡터 외적outer product

$$\mathbf{a}_1\mathbf{b}_1^{\mathrm{T}} = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{41} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} \\ a_{21}b_{11} & a_{21}b_{12} \\ a_{31}b_{11} & a_{31}b_{12} \\ a_{41}b_{11} & a_{41}b_{12} \end{bmatrix}$$

matrix.ipynb

인공신경망의 행렬 표현





$$W_1 = \begin{bmatrix} -15 & 3 & 5 \\ 18 & -6 & -3 \end{bmatrix} \quad W_2 = \begin{bmatrix} 4 & 10 & -9 \end{bmatrix}$$

simple_nn.gsheet matrix.ipynb

인공지능 이산수학

그래프

조준우
metamath@gmail.com

그래프의 정의

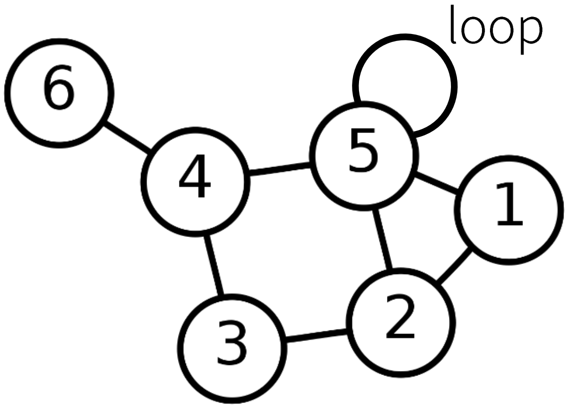
- 정점vertex의 집합 V와 서로 다른 정점쌍 (v_i, v_j)를 연결하는 모서리edge의 집합 E로 구성된 구조*

$G = (V, E)$ 정점vertex의 집합 V와 모서리edge의 집합 E로 구성된 그래프

$V = \{v_1, v_2, \dots, v_n\}$ 정점의 집합 V

$E = \{e_1, e_2, \dots, e_m\}$ 모서리의 집합 E

$= \{(v_i, v_j), \dots\}$ 모서리가 있는 정점의 쌍으로 표현



*박주미, 컴퓨팅 사고력을 키우는 이산수학, 한빛미디어, p.374

Public domain, https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)

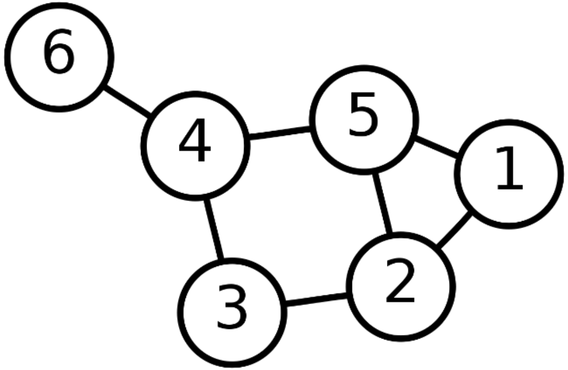
그래프 예

- 정점vertex의 집합 V와 서로 다른 정점쌍 (v_i, v_j)를 연결하는 모서리edge의 집합 E로 구성된 구조*

$G = (V, E)$ 정점vertex의 집합 V와 모서리edge의 집합 E로 구성된 그래프

$V = \{1,2,3,4,5,6\}$

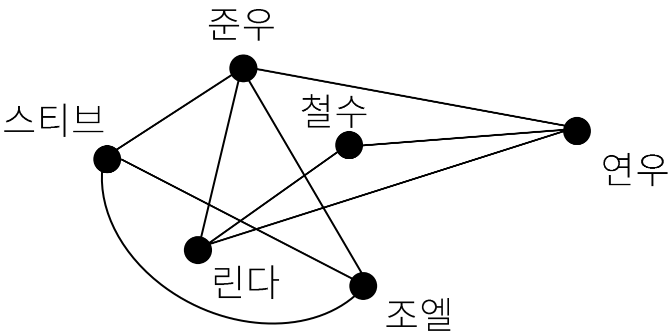
$E = \{(1,5), (1,2), (5,2), (4,5), (3,2), (4,3), (6,4)\}$

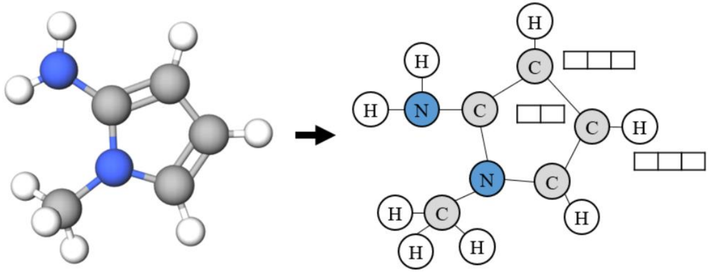


Public domain, https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)

그래프의 활용

- 소셜 네트워크
- 대진표
- 웹 사이트 링크 관계도
- 연구 인용 관계도
- 분자 구조



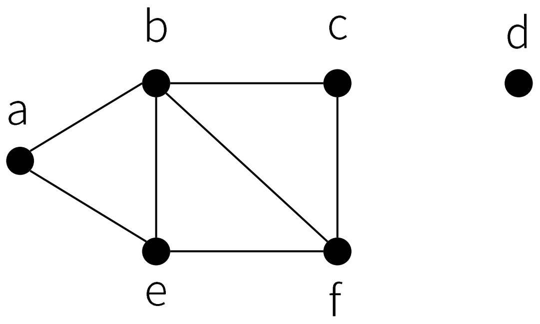


그래프의 용어

- 인접adjacent
  - 정점 v_i 과 v_j를 연결하는 모서리가 존재: 정점 v_i 과 v_j는 인접한다.*
  - (v_i, v_j), $v_i \sim v_j$
  - v_i는 v_j의 이웃neighbor이다.
- 근접incident
  - 모서리 e_k는 정점 v_i와 v_j를 연결: 모서리 e_k는 정점 v_i 과 v_j에 근접한다.*
- $N(v)$: 정점 v의 이웃 정점 집합
- 차수degree
  - $d(v)$: 정점 v에 근접하는 모서리의 개수*
  - 루프는 두 번으로 계산

*박주미, 컴퓨팅 사고력을 키우는 이산수학, 한빛미디어, p.374

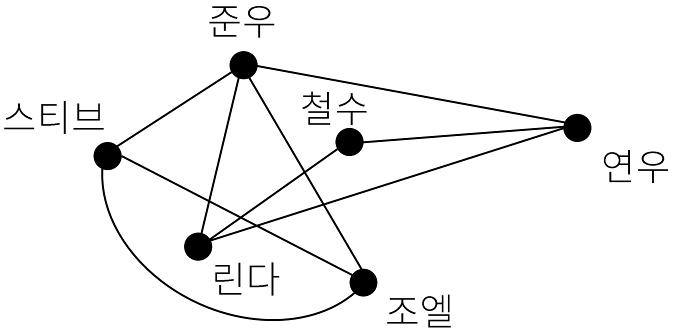
그래프의 용어 예



- $d(a) = 2, d(b) = 4, d(c) = 2,$
  $d(d) = 0, d(e) = 3, d(f) = 3$
- $N(a) = \{b, e\}, N(b) = \{a, e, f, c\}, N(c) = \{b, f\},$
  $N(d) = \emptyset, N(e) = \{a, b, f\}, N(f) = \{e, b, c\}$
- 정점 a와 b는 인접한다.

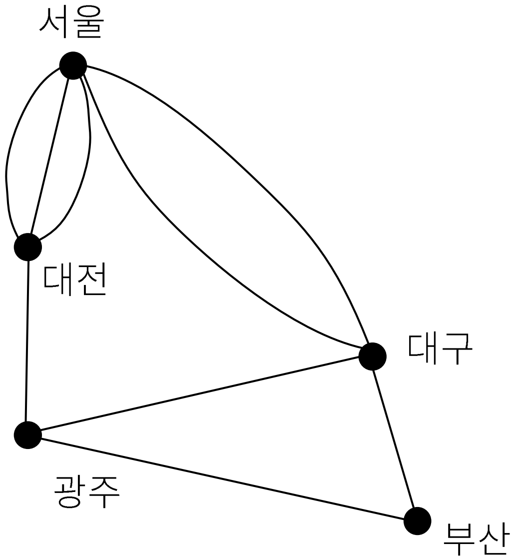
그래프의 형태

- 단순그래프simple graph
  - 두 정점 사이에 오직 하나의 모서리가 있는 그래프
- 다중그래프
- 의사그래프
- 가중치 그래프



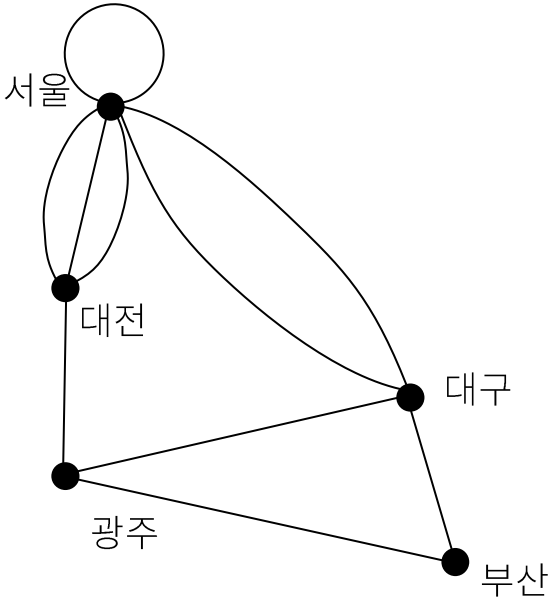
그래프의 형태

- 단순그래프simple graph
  - 두 정점 사이에 오직 하나의 모서리가 있는 그래프
- 다중그래프multi graph
  - 두 정점 사이에 둘 이상의 모서리가 있는 그래프
- 의사그래프
- 가중치 그래프



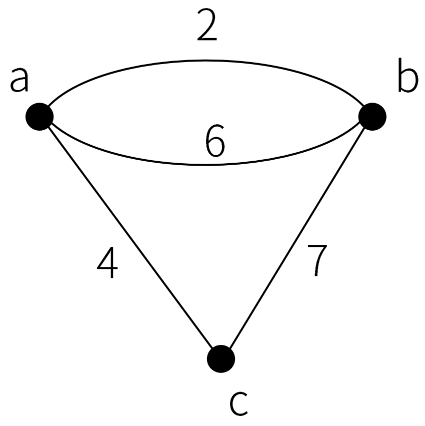
그래프의 형태

- 단순그래프simple graph
  - 두 정점 사이에 오직 하나의 모서리가 있는 그래프
- 다중그래프multi graph
  - 두 정점 사이에 둘 이상의 모서리가 있는 그래프
- 의사그래프pseudo graph
  - 루프를 허용하는 그래프
- 가중치 그래프



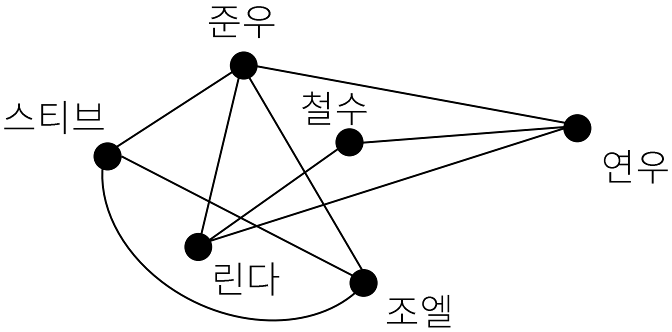
그래프의 형태

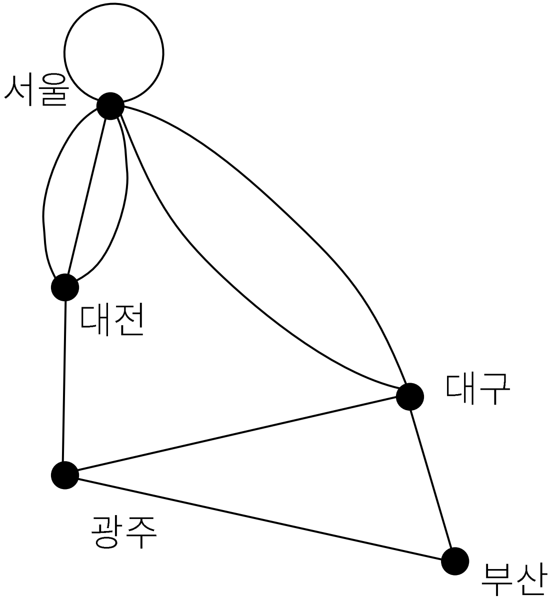
- 단순그래프simple graph
  - 두 정점 사이에 오직 하나의 모서리가 있는 그래프
- 다중그래프multi graph
  - 두 정점 사이에 둘 이상의 모서리가 있는 그래프
- 의사그래프pseudo graph
  - 루프를 허용하는 그래프
- 가중치 그래프weighted graph
  - 모서리에 가중치가 부여된 그래프

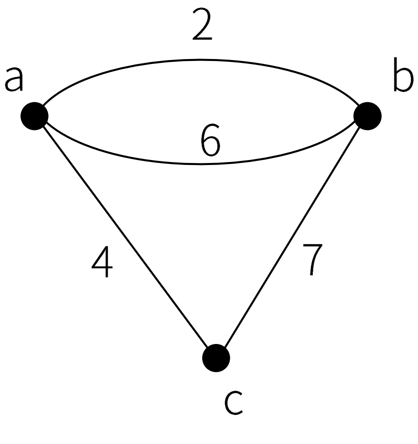


그래프의 형태

- 비방향(무향) 그래프undirected graph

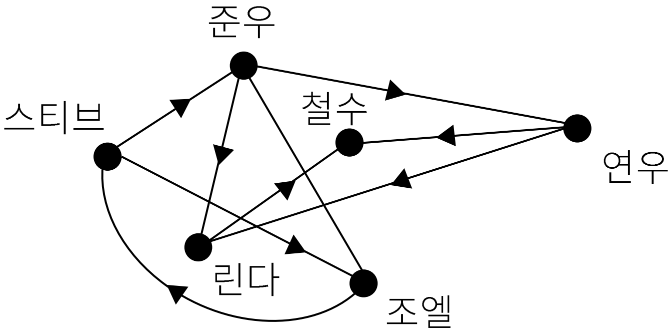


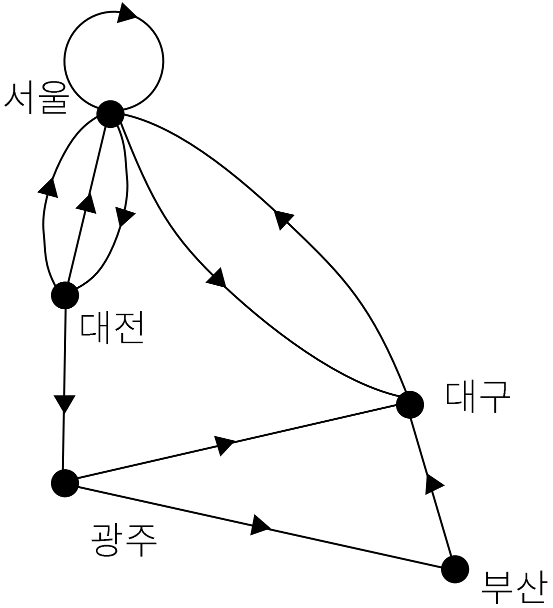


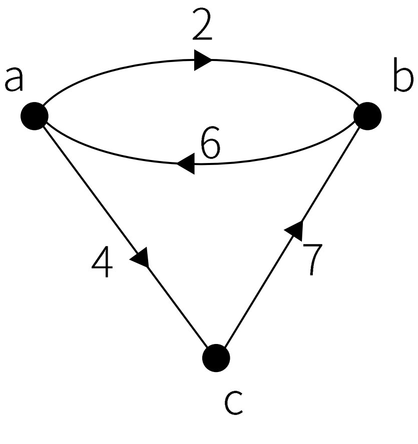


그래프의 형태

- 방향(유향) 그래프directed graph

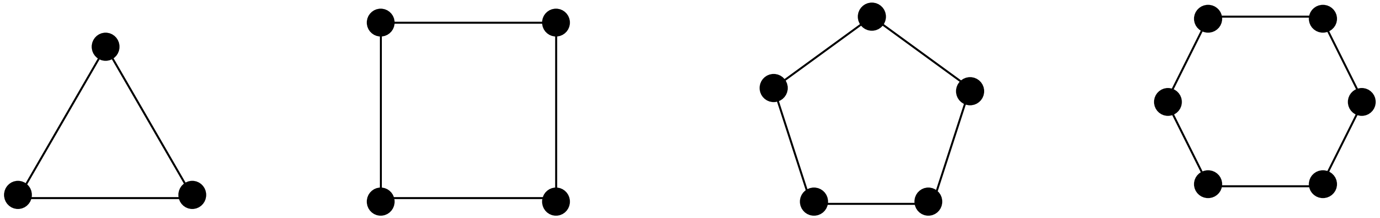






특별한 그래프

- 순환 Cycle, C_n
  - $n \geq 3$ 인 n개 정점과 $(v_1, v_2), (v_2, v_3), \cdots, \ (v_n, v_1)$인 모서리로 이루어진 그래프



C_3 C_4 C_5 C_6

그래프 개념 확인

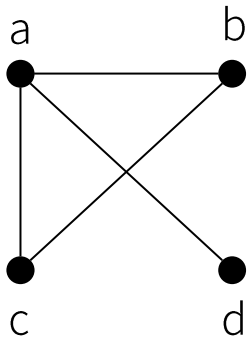
- 집합 $A_1, A_2, \cdots, A_n$ 들의 교집합 그래프는 두 집합 간에 공집합이 아닌 교집합이 있는 경우, 두 집합을 연결하는 변을 가지는 그래프입니다.
  다음 집합들에 대해서 교집합 그래프를 그리세요.*

$$A_1 = \{\dots, -4, -3, -2, -1, 0\},$$
$$A_2 = \{\dots, -2, -1, 0, 1, 2, \dots\},$$
$$A_3 = \{\dots, -6, -4, -2, 0, 2, 4, 6, \dots\},$$
$$A_4 = \{\dots, -5, -3, -1, 1, 3, 5, \dots\},$$
$$A_5 = \{\dots, -6, -3, 0, 3, 6, \dots\}$$

*Kenneth H. Rosen, Discrete Mathematics and Its Applications 8 ed., p.683

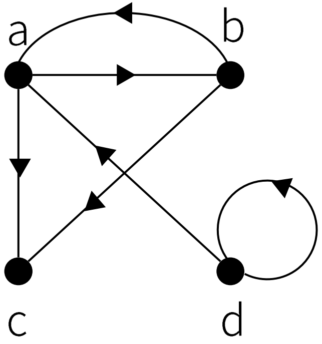
그래프의 표현

- 인접리스트adjacency list
  - 각 정점에 인접하는 정점들을 리스트로 표현



Vertex	Adjacent Vertices
a	b, c, d
b	a, c
c	b, a
d	a

방향 다중 그래프
- 출발에서 도착으로



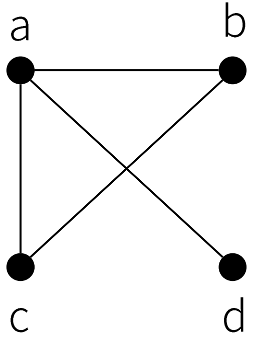
Initial Vertex	Terminal Vertices
a	b, c
b	a, c
c	
d	a, d

그래프의 표현

- 인접행렬adjacency matrix: A
  - 그래프 $G = (V, E)$에서 $|V| = n$ 일 때, $n \times n$ 행렬 $A = [a_{ij}]$

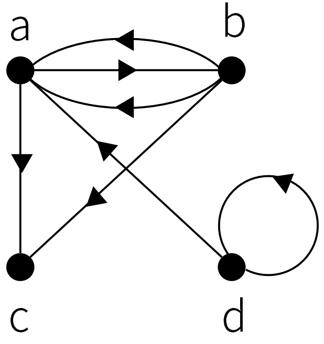
$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G \\ 0 & \text{otherwise.} \end{cases}$$

방향 다중 그래프
- 모서리 수를 직접 표시
- 행: 출발 정점, 열: 도착 정점



$$\begin{array}{cc} & \begin{array}{cccc} a & b & c & d \end{array} \\ A = & \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{array}{c} a \\ b \\ c \\ d \end{array} \end{array}$$

Symmetry

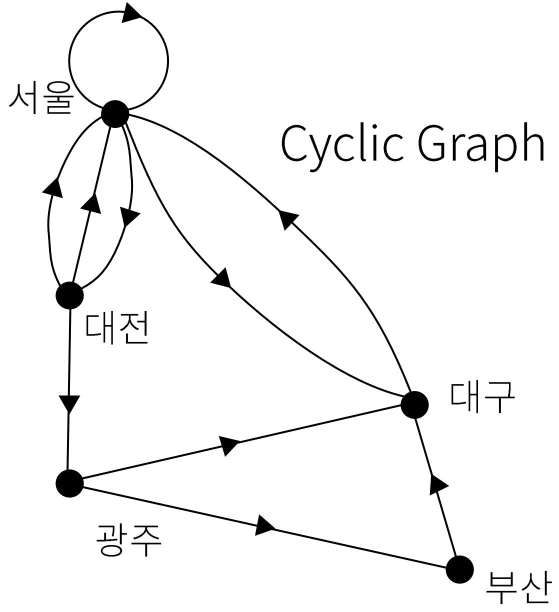


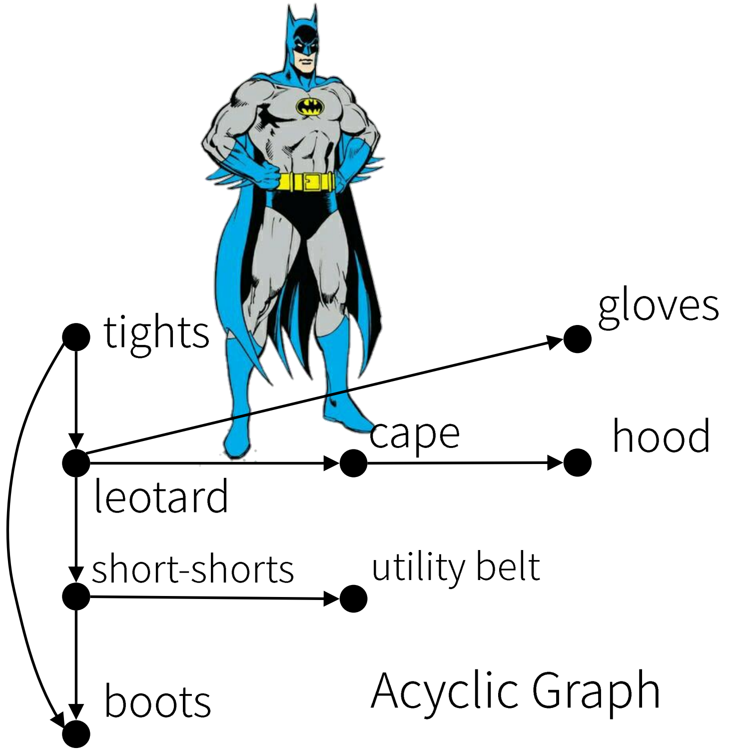
$$\begin{array}{cc} & \begin{array}{cccc} a & b & c & d \end{array} \\ A = & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{array}{c} a \\ b \\ c \\ d \end{array} \end{array}$$

Asymmetry

DAG: Directed Acyclic Graph

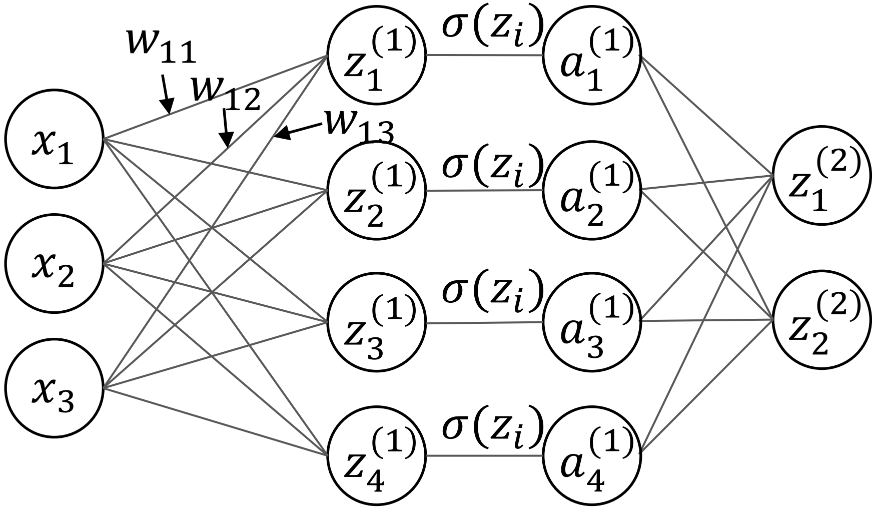
- 방향 비순환 그래프DAG
  - 순환이 존재하지 않는 방향 그래프





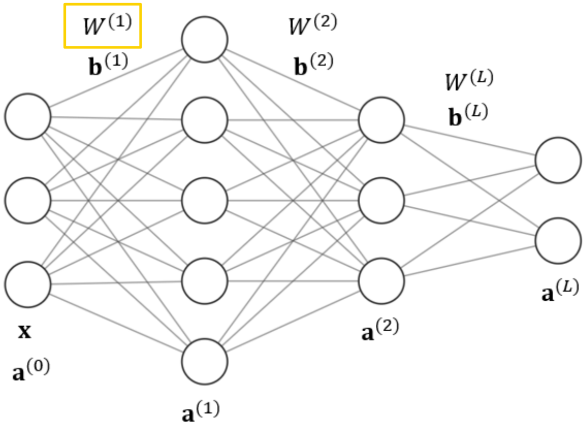
👀: 뉴럴넷과 그래프

- 가중 방향 비순환 그래프Weighted Directed Acyclic Graph



가중치의 표현

- w_{ij}: j 입력 인덱스, i 출력 인덱스입력



Weight

$$W^{(1)} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \\ w_{51} & w_{52} & w_{53} \end{bmatrix}$$

Bias

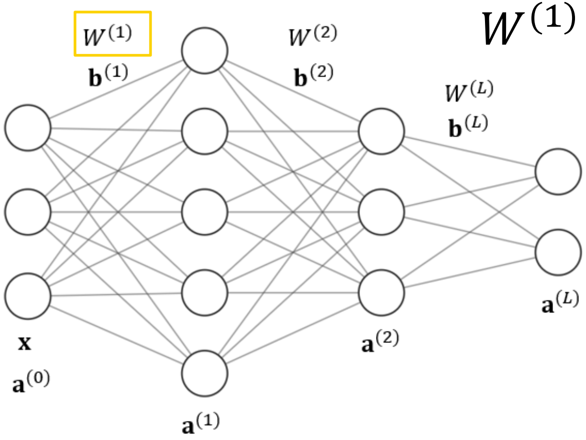
$$\mathbf{b}^{(1)} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\overbrace{\mathbf{a}^{(1)}}^{(5,1)} = f\left(\underbrace{W^{(1)}}_{(5,3)}\overbrace{\mathbf{x}}^{(3,1)} + \underbrace{\mathbf{b}^{(1)}}_{(5,1)}\right)$$

가중치의 표현

- w_{ij}: i 입력 인덱스, j 출력 인덱스입력



Weight

$$W^{(1)} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ w_{21} & w_{22} & w_{23} & w_{24} & w_{25} \\ w_{31} & w_{32} & w_{33} & w_{34} & w_{35} \end{bmatrix}$$

Bias

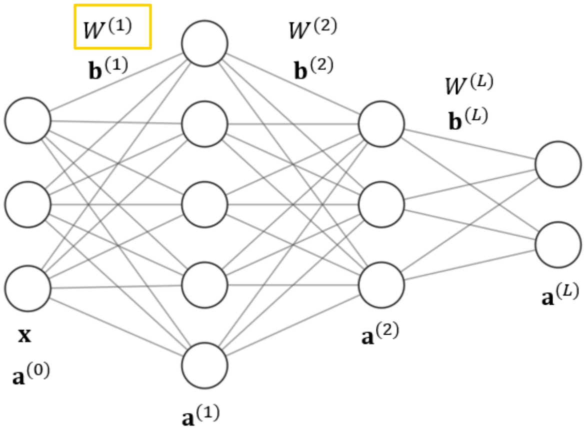
$$\mathbf{b}^{(1)^T} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix}^T$$

$$\mathbf{x}^T = [x_1 \quad x_2 \quad x_3]$$

$$\overset{(1,5)}{\mathbf{a}^{(1)^T}} = f\left(\overset{(1,3)}{\mathbf{x}^T}\underset{(3,5)}{W^{(1)}} + \underset{(1,5)}{\mathbf{b}^{(1)^T}}\right)$$

가중치의 표현

- 뒤죽 박죽



Weight

$$W^{(1)} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \\ w_{51} & w_{52} & w_{53} \end{bmatrix}$$

Bias

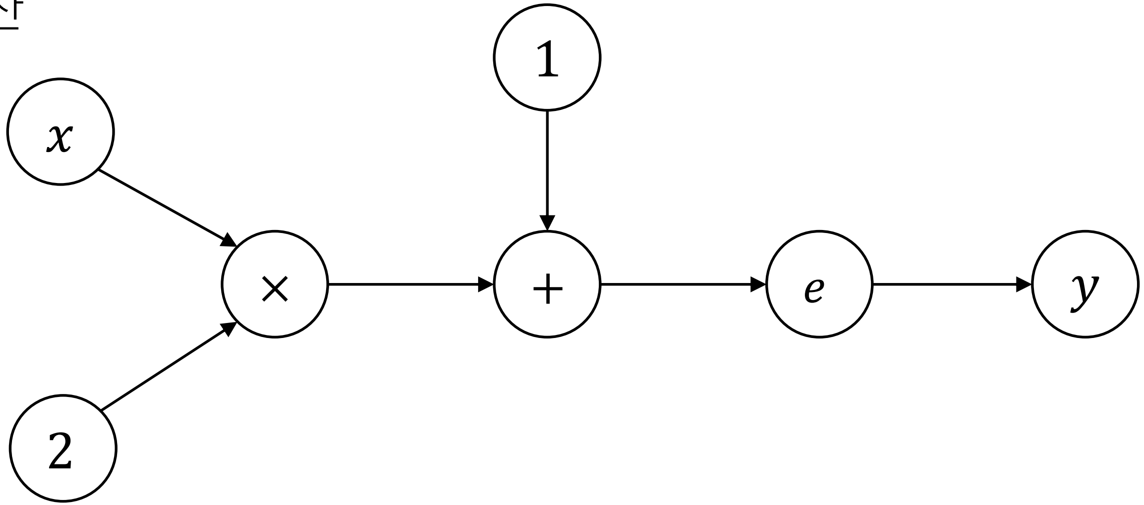
$$\mathbf{b}^{(1)} = [b_1 \quad \cdots \quad b_5]$$

$$\mathbf{x} = [x_1 \quad x_2 \quad x_3]$$

$$\overset{(1,5)}{\mathbf{a}^{(1)}} = f\left(\overset{(1,3)}{\mathbf{x}}\underset{(3,5)}{W^{(1)^T}} + \underset{(1,5)}{\mathbf{b}^{(1)}}\right)$$

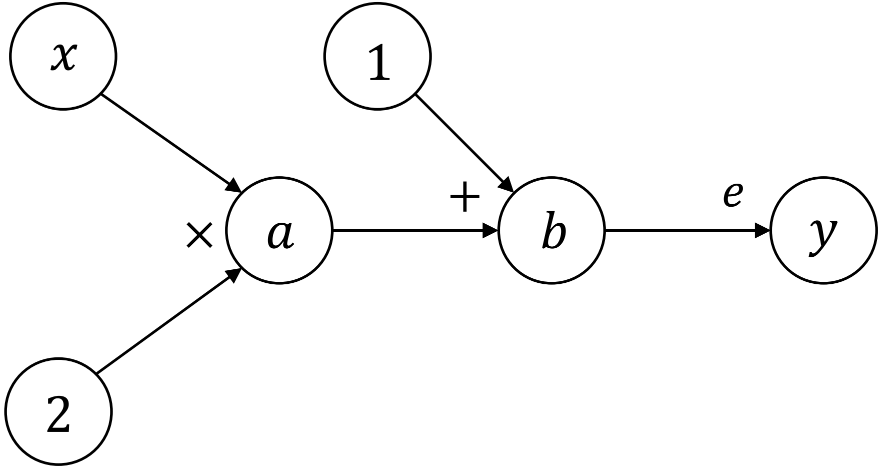
👀: 계산 그래프

- $y = e^{2x+1}$
- Directed Acyclic Graph
- Vertex: 변수, 연산



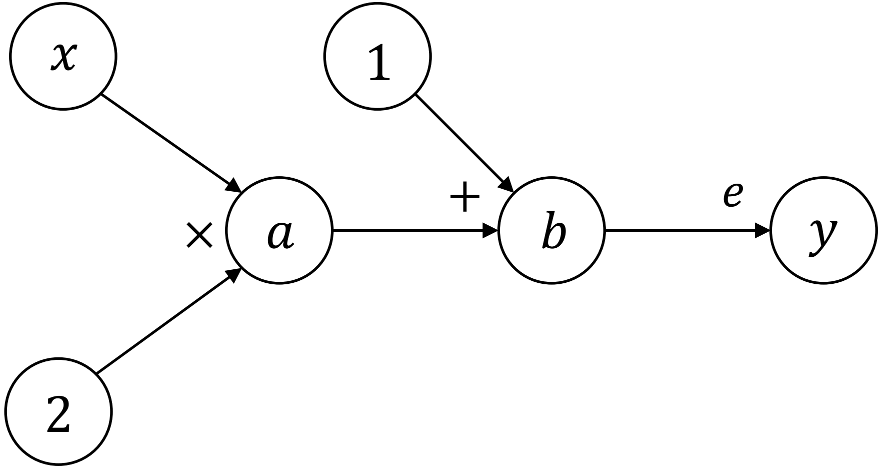
계산 그래프

- $y = e^{2x+1}$
- Directed Acyclic Graph
- Vertex: 변수
- Edge: 연산



계산 그래프의 포워드 연산

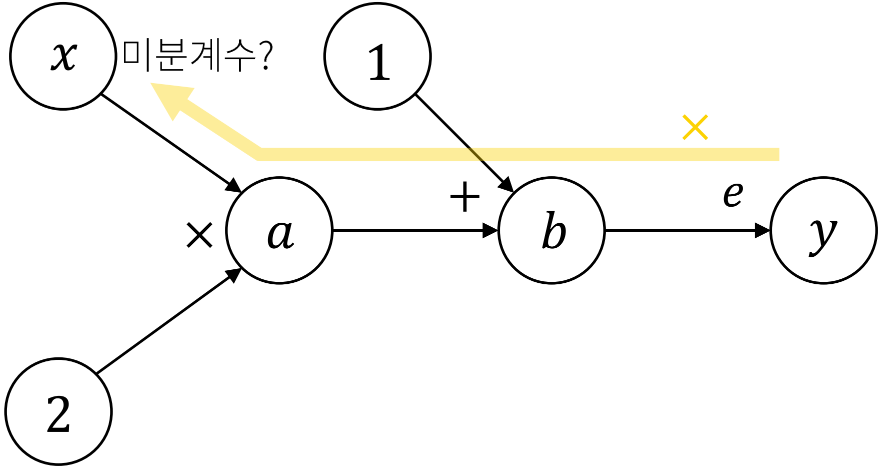
- $y = e^{2x+1}$
- 모서리의 방향을 따라 함숫값을 계산
- Vertex: 변수
- Edge: 연산



computational_graph.ipynb

계산 그래프의 백워드 연산

- $y = e^{2x+1}$
- 모서리의 역방향을 따라 미분계수를 계산
- Vertex: 변수
- Edge: 연산



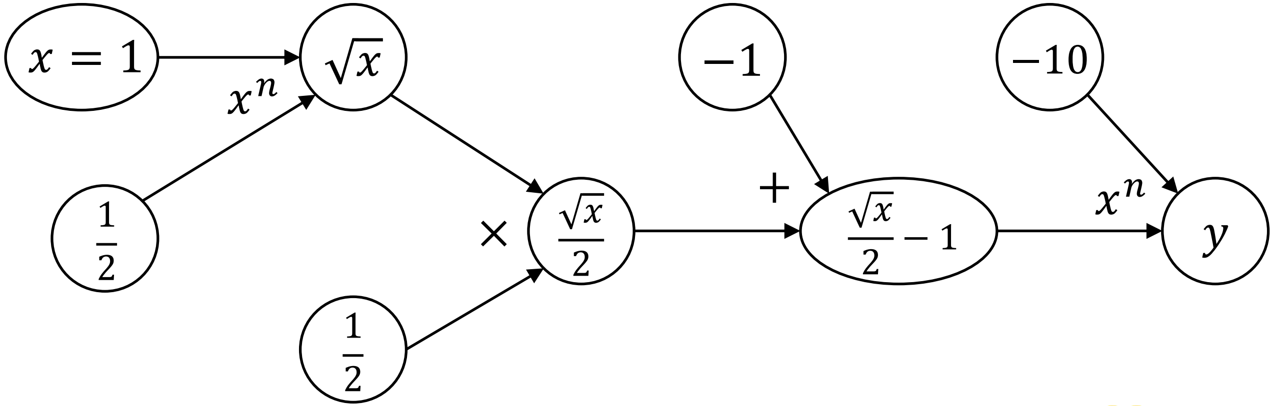
$$\frac{d}{dx} ax =$$

$$\frac{d}{dx}(a + x) =$$

$$\frac{d}{dx} e^x =$$

연습: 포워드 연산

$$y = \left(\frac{\sqrt{x}}{2} - 1\right)^{-10}$$



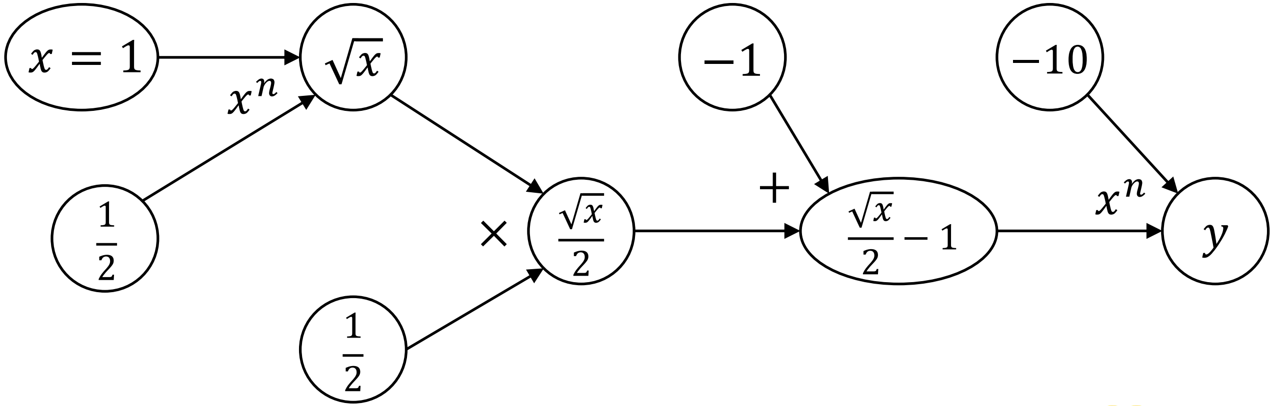
computational_graph.ipynb

연습: 백워드 연산

$$y = \left(\frac{\sqrt{x}}{2} - 1\right)^{-10}$$

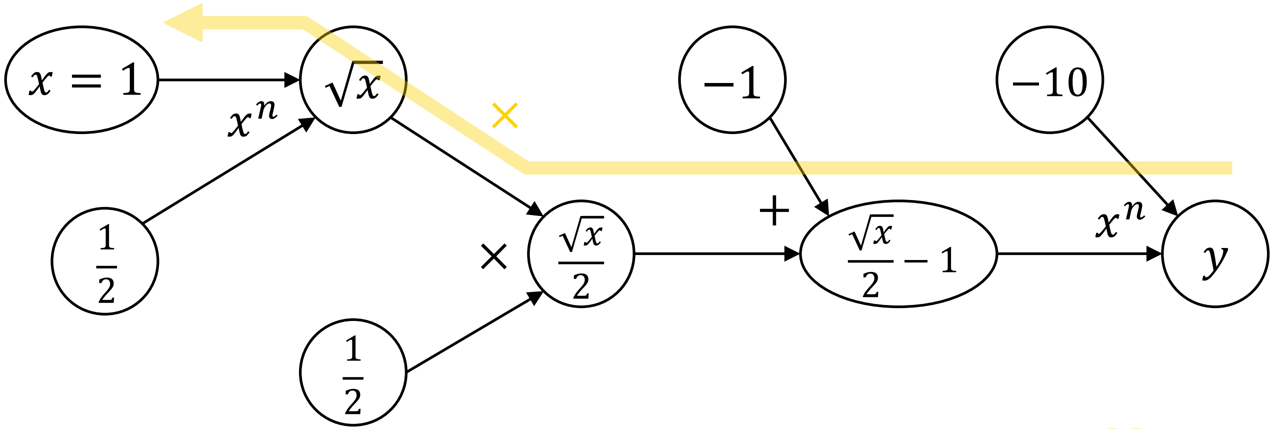
$$\frac{d}{dx}ax = \qquad \frac{d}{dx}e^{x} =$$

$$\frac{d}{dx}(a+x) = \qquad \frac{d}{dx}x^{n} =$$



연습: 백워드 연산

$$y = \left(\frac{\sqrt{x}}{2} - 1\right)^{-10} \qquad \frac{dy}{dx} = -\frac{5}{2\sqrt{x}}\left(\frac{\sqrt{x}}{2} - 1\right)^{-11}$$



[Automatic differentiation in machine learning: a survey]

computational_graph.ipynb