선형대수학팀

3팀
이재현
김규범
김민지
이정우
조혜현

INDEX

1

주성분 분석(PCA)

차원의 저주 & 차원의 축소

차원

설명변수의 개수

Input 데이터에서 X변수의 개수

차원 증가

↓

X변수가 늘어남

↓

데이터를 더욱 잘 설명

차원의 저주 & 차원의 축소

차원이 **높으면** 항상 좋은 걸까?

차원

Input 데이터에서 X변수의 개수

차원의 저주 & 차원의 축소

차원

설명변수의 개수

Input 데이터에서 X변수의 개수

No!

차원 증가

“ 차원의 저주 ”

X변수가 늘어남

데이터를 더욱 잘 설명

차원의 저주 & 차원의 축소

차원의 저주

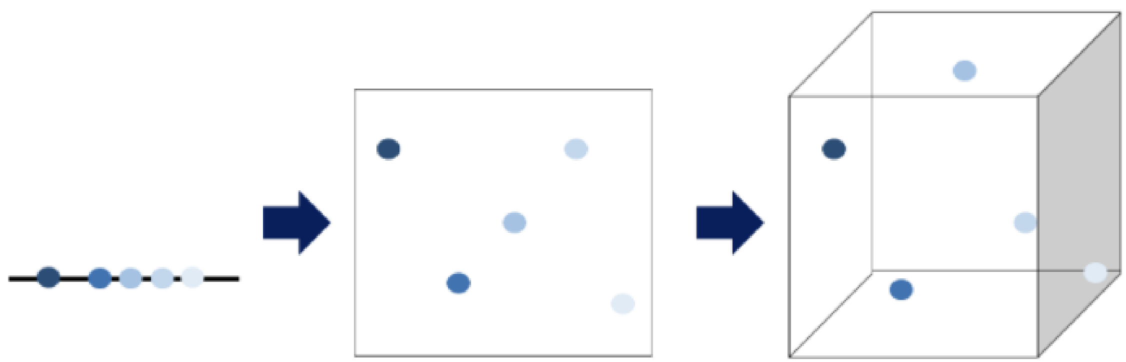
차원이 어느 정도 이상 증가하면
성능이 저하되는 현상

- 연산량 증가 → 데이터의 학습 속도 저하
- 데이터의 밀도 감소 → 성능 저하

차원의 저주 & 차원의 축소

차원의 저주

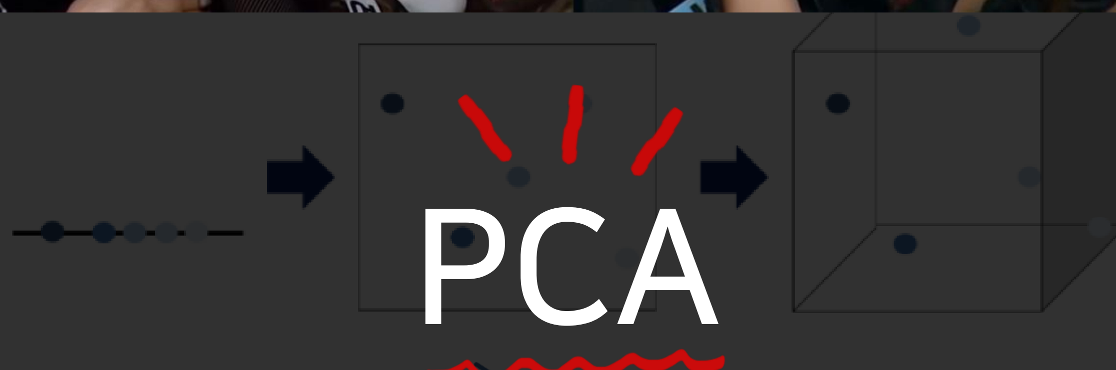
차원이 어느 정도 이상 증가하면
성능이 저하되는 현상



차원 증가 → 밀도 줄어들어 성능 저하

주성분 분석(PCA)

차원의 저



차원 증가 → 밀도 줄어들어 성능 저하

공분산에 대한 선형대수에서의 해석

공분산

두 확률변수의 선형관계에 대한 정보를 주는 척도

$$Cov(X,Y) = E(XY) - E(X)E(Y)$$

X = [x_1 $x_2 \cdots$ x_n]$^\top$으로 구성된 행렬이며, x_i 가 각각 벡터일 때,

$$\text{공분산 행렬 } K_{XX} = \begin{bmatrix} Cov(X_1,X_1) & Cov(X_1,X_2) & \cdots & Cov(X_1,X_n) \\ Cov(X_2,X_1) & Cov(X_2,X_2) & \cdots & Cov(X_2,X_n) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_n,X_1) & Cov(X_n,X_2) & \cdots & Cov(X_n,X_n) \end{bmatrix}$$

$$= V(X_n)$$

공분산에 대한 선형대수에서의 해석

공분산 행렬을 통한 선형변환

$$\begin{bmatrix} V(X) & Cov(X,Y) \\ Cov(Y,X) & V(Y) \end{bmatrix}$$

분산 = 변수들이 퍼져 있는 정도

공분산 = 변수들의 공동적 움직임

공분산에 대한 선형대수에서의 해석

공분산 행렬을 통한 선형변환

$$\begin{bmatrix} V(X) & Cov(X,Y) \\ Cov(Y,X) & V(Y) \end{bmatrix}$$

분산 = 변수들이 퍼져 있는 정도

공분산 = 변수들의 공동적 움직임

분산과 공분산만큼 공간이 변화

변수 간의 연관성 설명

변수들이 어떻게 분포됐는지 표현

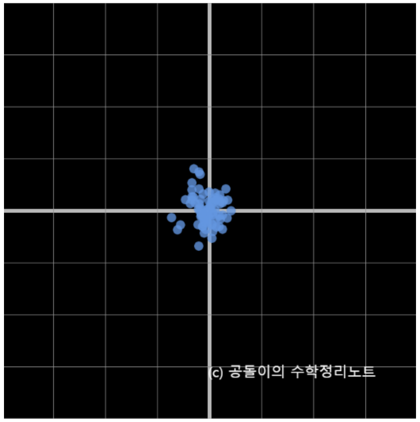
공분산에 대한 선형대수에서의 해석

공분산 행렬을 통한 선형변환

$$\begin{bmatrix} V(X) & Cov(X,Y) \\ Cov(Y,X) & V(Y) \end{bmatrix}$$

분산 = 변수들이 퍼져 있는 정도

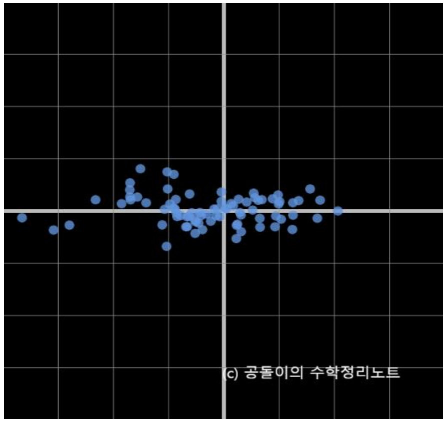
공분산 = 변수들의 공동적 움직임



X 방향으로 퍼진 정도

$$\begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}$$

Y 방향으로 퍼진 정도



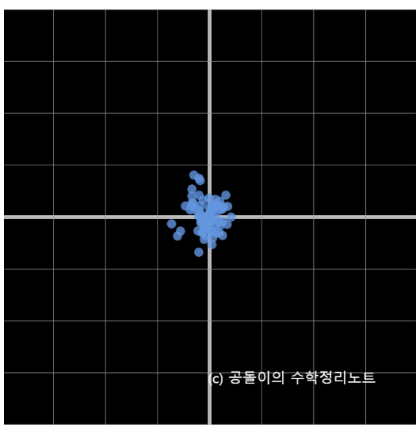
공분산에 대한 선형대수에서의 해석

공분산 행렬을 통한 선형변환

$$\begin{bmatrix} V(X) & Cov(X,Y) \\ Cov(Y,X) & V(Y) \end{bmatrix}$$

분산 = 변수들이 퍼져 있는 정도

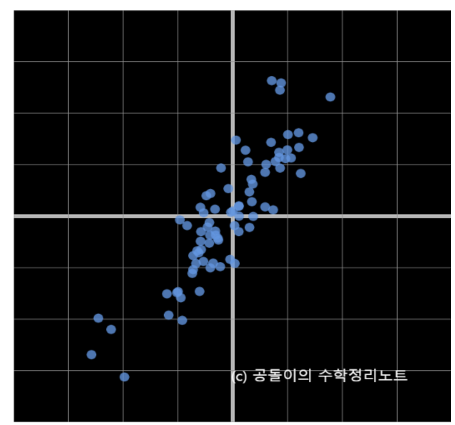
공분산 = 변수들의 공동적 움직임



$$\begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix}$$

X, Y 방향으로

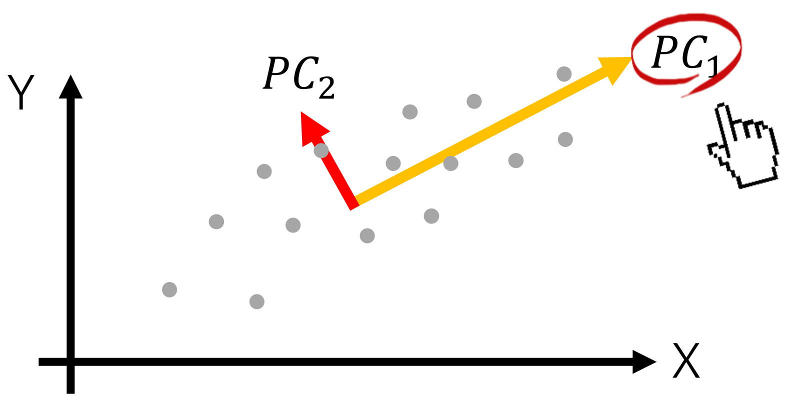
함께 퍼진 정도



주성분 분석 (PCA)

주성분 분석

데이터를 가장 잘 설명하는 주성분을 찾아 주성분이 이루는 공간으로 데이터를 정사영시켜 차원을 축소



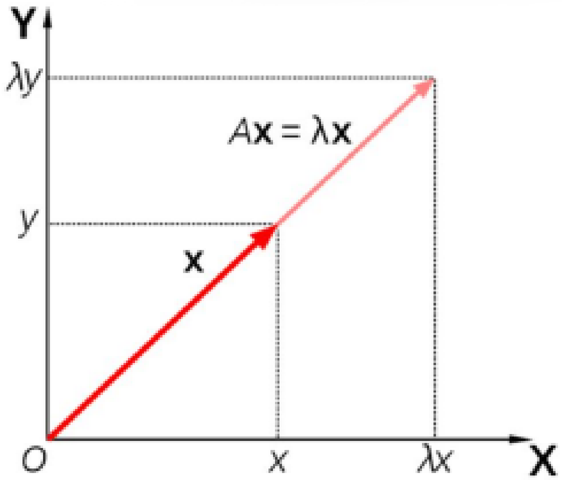
PC_1 이 PC_2 보다 데이터가 더 많이 퍼져 있음

분산이 더 큰 주성분을 선택

주성분 찾기

How?

공분산행렬
- 고유벡터 = PC
- 고유값 = 중요성 , 데이터가 퍼져 있는 정도



X의 방향을 그대로 두고

크기만을 변화시키는 선형 변환

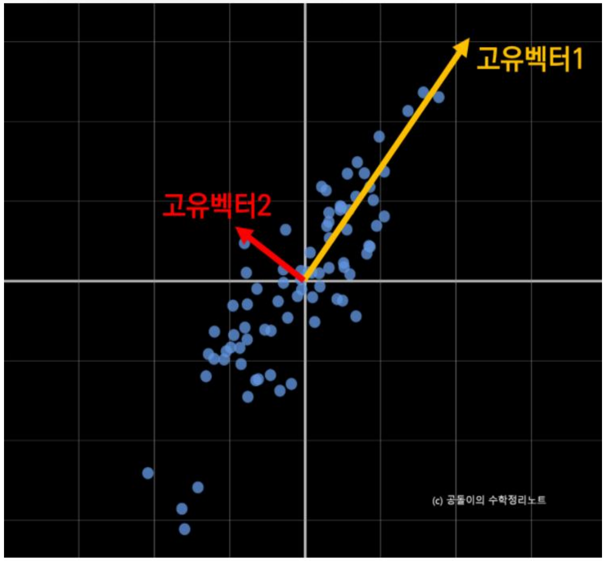
- **고유벡터** → 선형변환의 고정된 축
- **고유값** → 크기 변화 정도

주성분 찾기

How?

공분산행렬
- 고유벡터 = PC
- 고유값 = 중요성 , 데이터가 퍼져 있는 정도

공분산 행렬	$\begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix}$	
고유값	5.56	1.44
고유벡터	$\begin{pmatrix} 0.62 \\ 0.78 \end{pmatrix}$	$\begin{pmatrix} -0.78 \\ 0.62 \end{pmatrix}$
	고유벡터1	고유벡터2



주성분 … 기

고유벡터1의 고유값은 5.56으로 고유벡터2보다

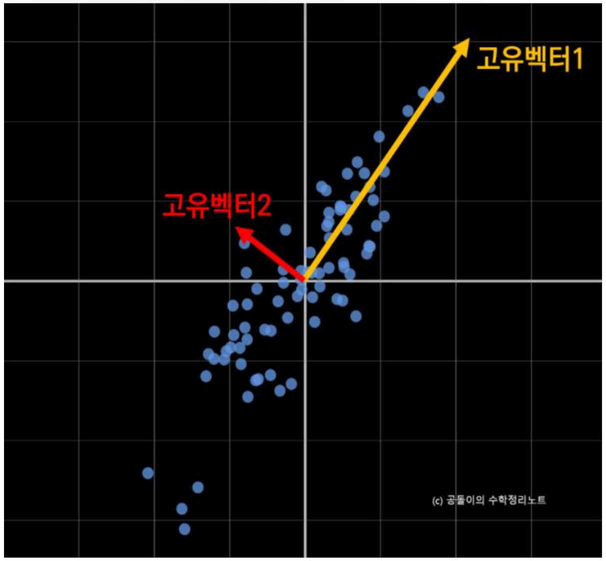
많은 데이터에 대해 분포

공분산행렬 — 고유벡터 = PC

고유값 = 중요성 , 데이터가 퍼져 있는 정도

✓ 고유벡터1이 고유벡터2보다 더 중요!!

공분산 행렬	$\begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix}$	
고유값	5.56	1.44
고유벡터	$\begin{pmatrix} 0.62 \\ 0.78 \end{pmatrix}$ 고유벡터1	$\begin{pmatrix} -0.78 \\ 0.62 \end{pmatrix}$ 고유벡터2



주성분 선택

PCA결과의 설명되는 누적 분산량을 근거로 주성분의 개수 선택

	PC1	PC2	PC3	PC4
Standard Deviation	2.3925	0.44457	0.2522	0.0892
Proportion of Variance	0.8548	0.03294	0.0106	0.0034
Cumulative Proportion	0.8548	0.91273	0.9883	1.0000

누적비율로 데이터의 변동을 얼마나 설명하는지 파악 가능

주성분 선택

PCA결과의 설명되는 누적 분산량을 근거로 주성분의 개수 선택

	PC1	PC2	PC3	PC4
Standard Deviation	2.3925	0.44457	0.2522	0.0892
Proportion of Variance	0.8548	0.03294	0.0106	0.0034
Cumulative Proportion	0.8548	0.91273	0.9883	1.0000

일반적으로, 누적비율이 90%이상 되도록 선택

4차원 데이터 → *2차원 데이터*

2

벡터 미적분학

편미분의 개념

편미분
(Partial Differentiation)

- 어떤 다변수함수에 대해 **하나의 변수**에 주목
  나머지 변수의 값 고정시켜 놓고 **그 변수로 미분**
- 다변수함수에서 1개의 변수 값이 변화할 때의
  변화율을 알기 위해 사용됨
- 편미분 기호로는 ∂를 사용

→ $f(x, y)$가 있을때 y를 상수로 보고
이것을 x로 미분하는 일을 **"x로 편미분한다"**고 함

편미분 수식

$$\frac{\partial f}{\partial x_1} = \lim_{h \to 0} \frac{f(x_1 + h, x_2, \dots, x_n) - f(x)}{h}$$

$$\frac{\partial f}{\partial x_2} = \lim_{h \to 0} \frac{f(x_1, x_2 + h, \dots, x_n) - f(x)}{h}$$

$$\vdots$$

$$\frac{\partial f}{\partial x_n} = \lim_{h \to 0} \frac{f(x_1, x_2, \dots, x_n + h) - f(x)}{h}$$

하나의 변수에 주목한다는 점 제외하고는

일반적인 미분 정의와 동일

편미분 수식

$$\frac{\partial f}{\partial x_1} = \lim_{h \to 0} \frac{f(x_1 + h, x_2, \dots, x_n) - f(x)}{h}$$

$$\frac{\partial f}{\partial x_2} = \lim_{h \to 0} \frac{f(x_1, x_2 + h, \dots, x_n) - f(x)}{h}$$

$$\vdots$$

$$\frac{\partial f}{\partial x_n} = \lim_{h \to 0} \frac{f(x_1, x_2, \dots, x_n + h) - f(x)}{h}$$

하나의 변수에 주목한다는 점 제외하고는

일반적인 미분 정의와 동일

편미분 계산

미분하는 변수 외에 다른 변수는 상수 취급하고

미분을 진행함

$f(x,y) = x^2 + xy$의 경우,

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

미분과 Chain Rule

Case

다변수함수 f를 구성하는 $x, y, z\ \cdots$가 다시 다른 변수 t에 대해 정의되어 있음

↓

변수 t에 대한 f의 도함수는 편미분을 사용해 다음과 같이 정의

$x(t), y(t), z(t) \cdots$

$$\frac{df(x, y, z, \ldots)}{dt} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial f}{\partial z}\frac{\partial z}{\partial t} + \cdots$$

미분과 Chain Rule

각각의 항에

“Chain Rule” 적용

Case

다변수함수 f를 구성하는 $x, y, z \cdots$가 다시 다른 변수 t에 대해 정의되어 있음

변수 t에 대한 f의 도함수는 편미분을 사용해 다음과 같이 정의

$(x)t, y(t), z(t) \cdots$

$$\frac{df(x, y, z, \ldots)}{dt} = \boxed{\frac{\partial f}{\partial x}\frac{\partial x}{\partial t}} + \boxed{\frac{\partial f}{\partial y}\frac{\partial y}{\partial t}} + \boxed{\frac{\partial f}{\partial z}\frac{\partial z}{\partial t}} + \cdots$$

미분과 Chain Rule

$$\frac{df}{dt} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial x_1(t)}{\partial t} \\ \frac{\partial x_2(t)}{\partial t} \end{bmatrix} = \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial t}$$

함수 f가 $x_1(t)$, $x_2(t)$로 구성되어 있을 때, t에 관한 f의 도함수

f를 구성하는 x_1, x_2가 한 개의 변수 t로 이루어졌을 때의 예

미분과 Chain Rule

$$\frac{\partial f}{\partial s} = \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial s} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial s}$$

$$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial t}$$

함수 f가 $x_1(s,t)$, $x_2(s,t)$로 구성되어 있을 때,

s,t에 대한 f의 도함수

미분과 Chain Rule

앞선 도함수를 행렬로 정리하면,

$$\begin{bmatrix} \frac{\partial f}{\partial s} & \frac{\partial f}{\partial t} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix}}_{\frac{\partial f}{\partial \vec{x}}} \underbrace{\begin{bmatrix} \frac{\partial x_1}{\partial s} & \frac{\partial x_1}{\partial t} \\ \frac{\partial x_2}{\partial s} & \frac{\partial x_2}{\partial t} \end{bmatrix}}_{\frac{\partial f}{\partial (s,t)}}$$

$$\boxed{\begin{bmatrix} \frac{\partial f}{\partial s} & \frac{\partial f}{\partial t} \end{bmatrix} = \frac{\partial f}{\partial \vec{x}} \frac{\partial f}{\partial (s,t)}}$$

Gradient란?

Gradient

- 어떤 함수 f가 다중 변수 $x_1, x_2, x_3, \ldots$로 구성되어 있을 때,
- f의 gradient 각 변수에 대하여 f를 편미분 한 것을 행(또는 열)으로 나열한 것으로 표현
- 이때, gradient는 $\boldsymbol{\nabla f}$혹은 $\boldsymbol{grad\ f}$로 표기

Gradient란?

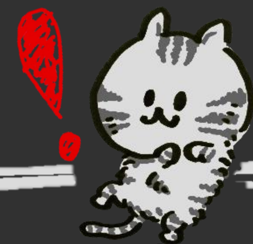
Gradient

$x_1, x_2, x_3, \dots$로 구성된 어떤 함수 f의 gradient

- 어떤 함수 f가 다중 변수 $x_1, x_2, x_3, \dots$로 구성되어 있을 때,
- f의 gradient 각 변수에 대하여 f를 편미분 한 것을 행(또는 열)으로 나열한 것으로 표현
- 이때, gradient는 $\boldsymbol{\nabla f}$ 혹은 $\boldsymbol{grad\ f}$로 표기

“ $$\nabla f = grad\ f = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} & \dfrac{\partial f}{\partial x_2} & \dfrac{\partial f}{\partial x_3} & \cdots \end{bmatrix}$$ ”

Gradient란?



Gradient

어떤 스칼라 함수 f에 대해서

gradient ∇f의 방향은

f가 가장 빨리 증가하는

방향을 가리킴

어떤 함수 f가 다중 변수 $x_1, x_2, \ldots$로 구성되어 있을 때,

f의 gradient 각 변수에 대하여 f를 편미분 한 것을 행(또는 열)으로 나열한 것으로 표현

이때, gradient는 ∇f 혹은 $grad\ f$로 표기

Gradient란?

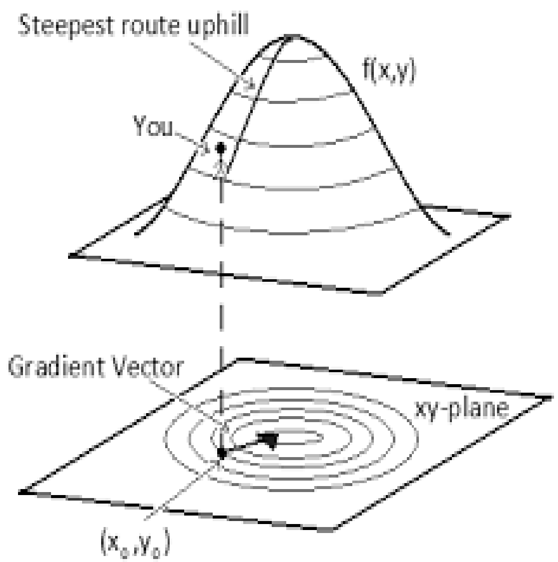
f에 대한 등고선을 그렸을 때, 각 등고선에서 어떤 지점의 접선이

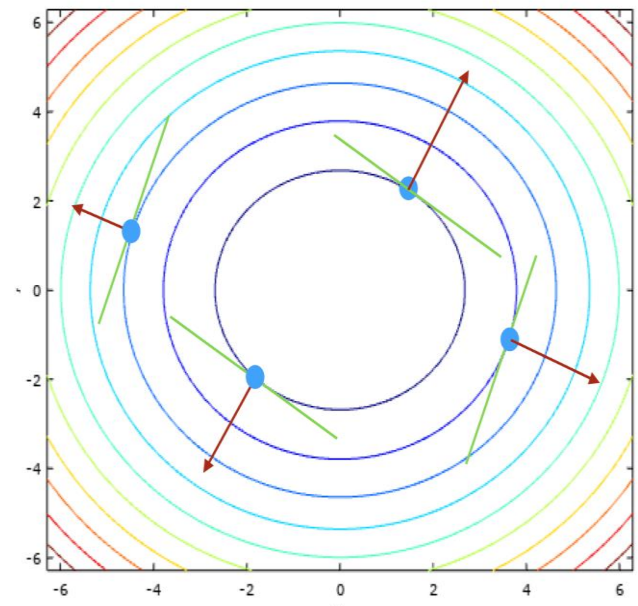
그 지점에서의 gradient와 반드시 **수직**을 이루게 됨

3차원 이상으로 확장했을 때,

한 점에서의 gradient는 **그 점에서의 접평면과 수직**을 이루게 됨

등고선과 Gradient





어떤 함수 f의 등고선과 gradient는 **수직**
즉, **접선**을 이루는 모습

직교 관계 증명

$w = f(x, y, z)$ 라는 3변수 함수가 있다고 하자.

등고선: 어떤 함수에 대해 함수 값이
동일한 것을 모아 선으로 그려 놓은 것

등고선을 구하기 위한 어떤 level c에 대한 교점의 집합

$f(x, y, z) = c$ (c는 상수)

"Level Surface"

직교 관계 증명

어떤 특정 Level Surface 값(이하 c)에 대한 함수를 살펴보자.

값이 c인 Level surface위의 점들을 모아 놓은

curve를 $r(t) = < x(t), y(t), z(t) >$라 하고,

$g(t) = f(x(t), y(t), z(t)) = c$라 하자.

즉, r은 값이 c를 가리키는 어떤 등고선 위의 점들을 나타낸 일종의 함수

직교 관계 증명

$g(t) = c$를 양변에 대해 미분하게 되면,

Chain Rule에 의해

$$\frac{dg}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} + \frac{\partial f}{\partial z}\frac{dz}{dt} = 0$$

Vector Form으로 정리하면

$$< \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} > \cdot < \frac{dx}{dt}, \frac{dy}{dt}, \frac{dz}{dt} > = 0$$

직교 관계 증명

앞선 식의 표현을 바꿔보면,

$$\nabla f \cdot r'(t) = 0$$

c에서의 Level surface위의 curve를 뜻하는

함수 r의 접선 기울기

등고선에서의 접선의 기울기

직교 관계 증명

앞선 식의 표현을 바꿔보면,

$$\nabla f \cdot \underline{r'}(t) = 0$$

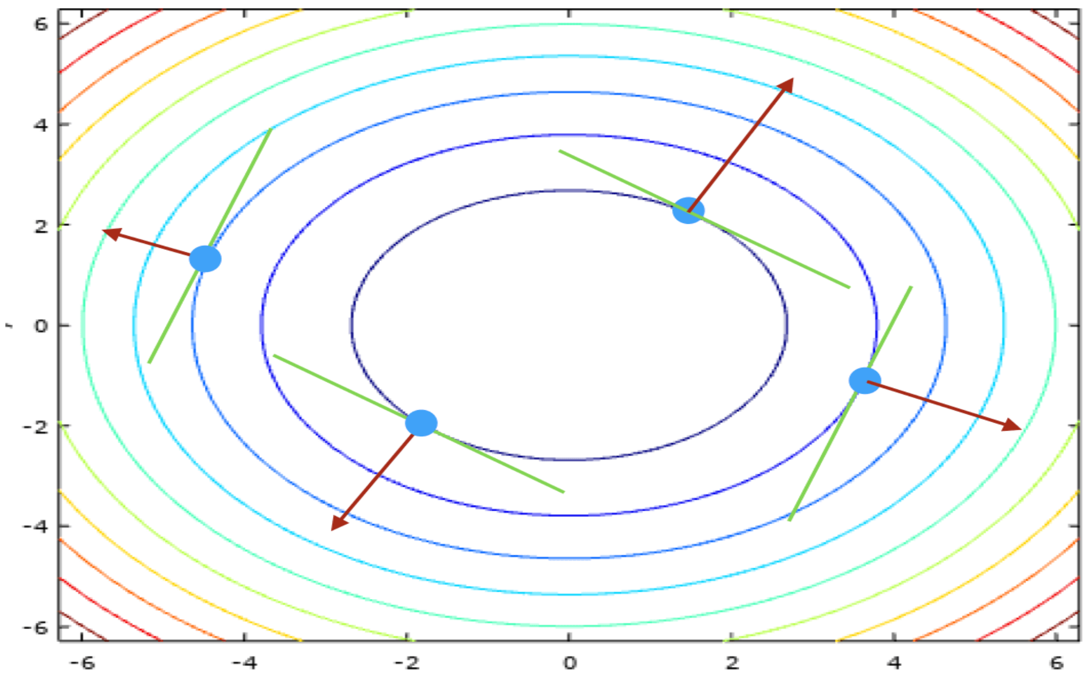
f의 gradient와 등고선에서의 접선의 기울기의

내적 값이 **0**이 되는 것

↓

f의 gradient와 등고선의 접선은 직교 관계를 이룸

직교 관계 증명



등고선의 접선과 f의 gradient가 **직교**를 이룬다는 말은

특정 점에서 **가장 크게 증가하는 방향**으로 **기울기 벡터의 방향이 설정**된다는 것!

벡터함수란?

벡터함수
(Vector-Valued Functions)

$\boldsymbol{f}$: $\mathbb{R}^n \rightarrow \mathbb{R}^m$과 벡터 **x** = [x$_1$, x$_2$, ⋯, x$_n$]^T 로 주어졌다고 할 때,

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix} \in \mathbb{R}^m$$ 으로 정의

벡터함수란?

벡터함수
(Vector-Valued Functions)

$\boldsymbol{f}$는 함수 $f_1, f_2, \dots, f_m$ 들의 벡터로 $\begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix}$ 로 표현되며

각 f_i는 $f_i\colon\ \mathbb{R}^n \rightarrow \mathbb{R}$로 일반적인 실수 함수

벡터함수의 Gradient

실수함수 $f_1, f_2, \ldots$에 대해 $\boldsymbol{f} = \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix}$, 벡터 $\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$로 주어졌다고 하자.

벡터함수 f를 특정 $\boldsymbol{x_i}$에 대해 편미분하면

$$\frac{\partial f}{\partial x_i} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_i} \\ \vdots \\ \dfrac{\partial f_m}{\partial x_i} \end{bmatrix}$$

벡터함수의 Gradient

실수함수 $f_1, f_2, \ldots$에 대해 $\boldsymbol{f} = \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix}$, 벡터 $\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$로 주어졌다고 하자.

벡터함수 f를 벡터 $\boldsymbol{x}$에 대해 편미분하면

$$\frac{\partial f(x)}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} & \cdots & \frac{\partial f(x)}{\partial x_n} \end{bmatrix}$$

벡터함수의 Gradient

앞선 식은 최종적으로 다음과 같이 전개됨

$$\frac{\partial f(x)}{\partial x} = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} & \cdots & \frac{\partial f(x)}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(x)}{\partial x_1} & \cdots & \frac{\partial f_m(x)}{\partial x_n} \end{bmatrix}$$

바로 위의 식을 벡터함수 $\boldsymbol{f}$의 gradient라고 정의하며,

$\nabla_x \boldsymbol{f}$ 로 표기

3

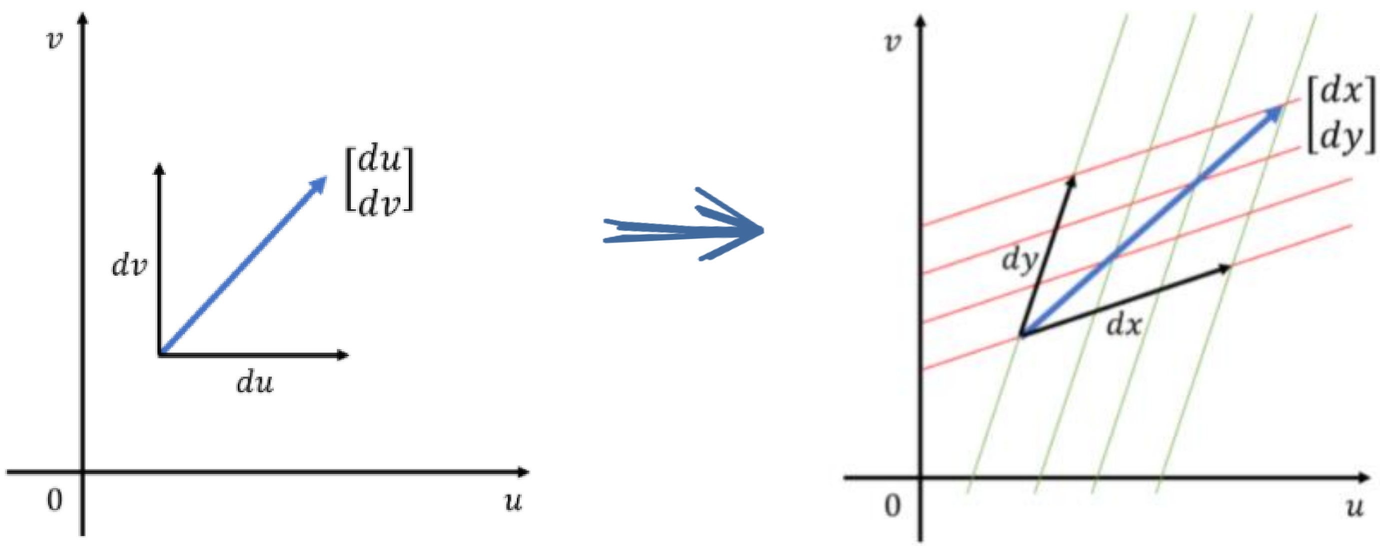
벡터 미적분학의 응용

자코비안(Jacobian)

자코비안

벡터함수 F의 gradient

기하학적으로 **축 변환**의 역할을 수행



자코비안(Jacobian)

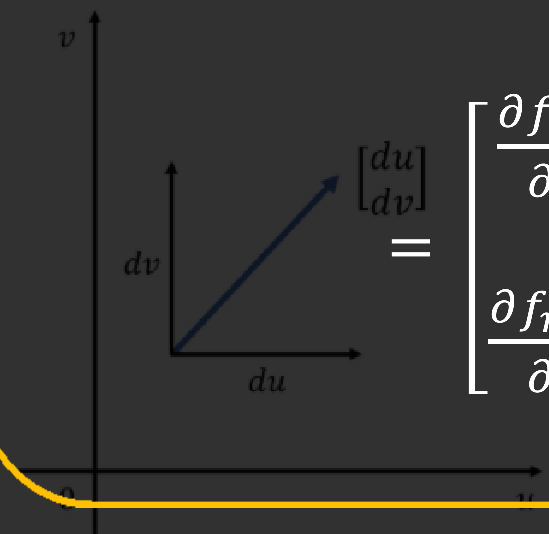
자코비안의 수식

자코비안

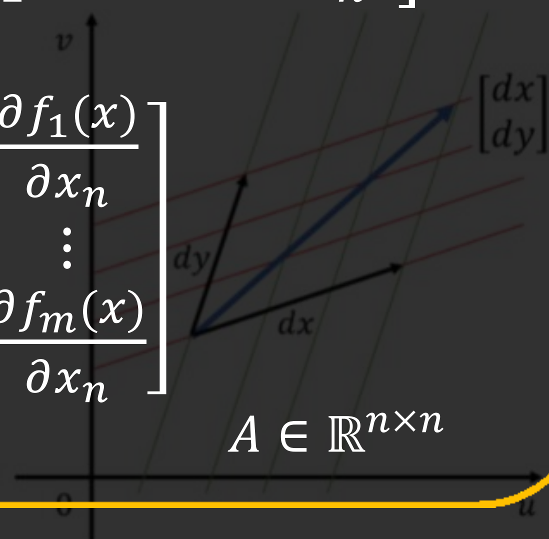
벡터함수 F의 gradient

기하학적으로 축 변환의 역할을 수행

$$\boldsymbol{J} = \nabla_{\mathrm{x}}\,\mathbf{f} = \frac{\partial f(x)}{\partial x} = \left[\frac{\partial f(x)}{\partial x_1} \cdots \frac{\partial f(x)}{\partial x_n}\right]$$



$$= \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(x)}{\partial x_1} & \cdots & \frac{\partial f_m(x)}{\partial x_n} \end{bmatrix}$$



$A \in \mathbb{R}^{n \times n}$

자코비안(Jacobian)

Example

$$\mathrm{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

x에서 y로의 축 변환

$y_1 = -2x_1 + x_2, y_2 = x_1 + x_2$의 관계를 가정

$J = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 1 & 1 \end{bmatrix}$를 x에 곱함으로써 y로 축 변환을 수행

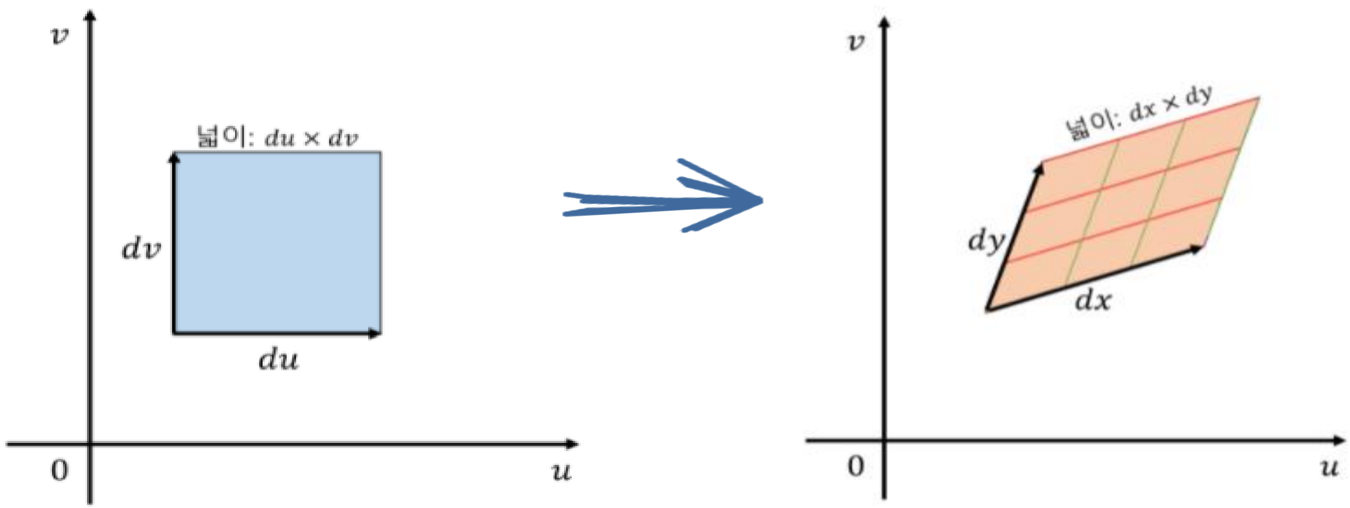
$x = (1,1)$은 $\begin{bmatrix} -2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$이므로 $y = (-1, 2)$로 변환됨

자코비안의 활용

수리통계학에서 축 변환을 위해 자코비안 행렬의 **행렬식** 사용

자코비안의 행렬식은 **좌표축을 변환시켰을 때의 크기 변화**를 의미

$$f_{y_1,y_2}(y_1,y_2) = |J| f_{x_1,x_2}(g(y_1,y_2), h(y_1,y_2))$$



자코비안의 활용

수리통계학에서 축 변환을 위해 자코비안 행렬의 **행렬식** 사용

자코비안의 행렬식은 **좌표축을 변환시켰을 때의 크기 변화**를 의미

$$f_{y_1,y_2}(y_1, y_2) = |J| f_{x_1,x_2}(g(y_1, y_2), h(y_1, y_2))$$

X의 pdf를 통해 Y의 pdf를 구해보자

독립변수 X_1, X_2에 대해 $Y_1 = X_1, Y_2 = X_1 + X_2$일 때,

$J = \begin{bmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_1}{\partial y_2} \\ \frac{\partial x_2}{\partial y_1} & \frac{\partial x_2}{\partial y_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$이 되어, $|J| = \begin{vmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_1}{\partial y_2} \\ \frac{\partial x_2}{\partial y_1} & \frac{\partial x_2}{\partial y_2} \end{vmatrix} = \begin{vmatrix} 1 & 0 \\ -1 & 1 \end{vmatrix} = \mathbf{1}$

자코비안의 활용

수리통계학에서 축 변환을 위해 자코비안 행렬의 **행렬식** 사용

자코비안의 행렬식은 **좌표축을 변환시켰을 때의 크기 변화**를 의미

$$f_{y_1,y_2}(y_1,y_2)=|J|f_{x_1,x_2}(g(y_1,y_2),h(y_1,y_2))$$

X의 pdf를 통해 Y의 pdf를 구해보자

독립변수 X_1, X_2에 대해 $Y_1 = X_1, Y_2 = X_1 + X_2$일 때,

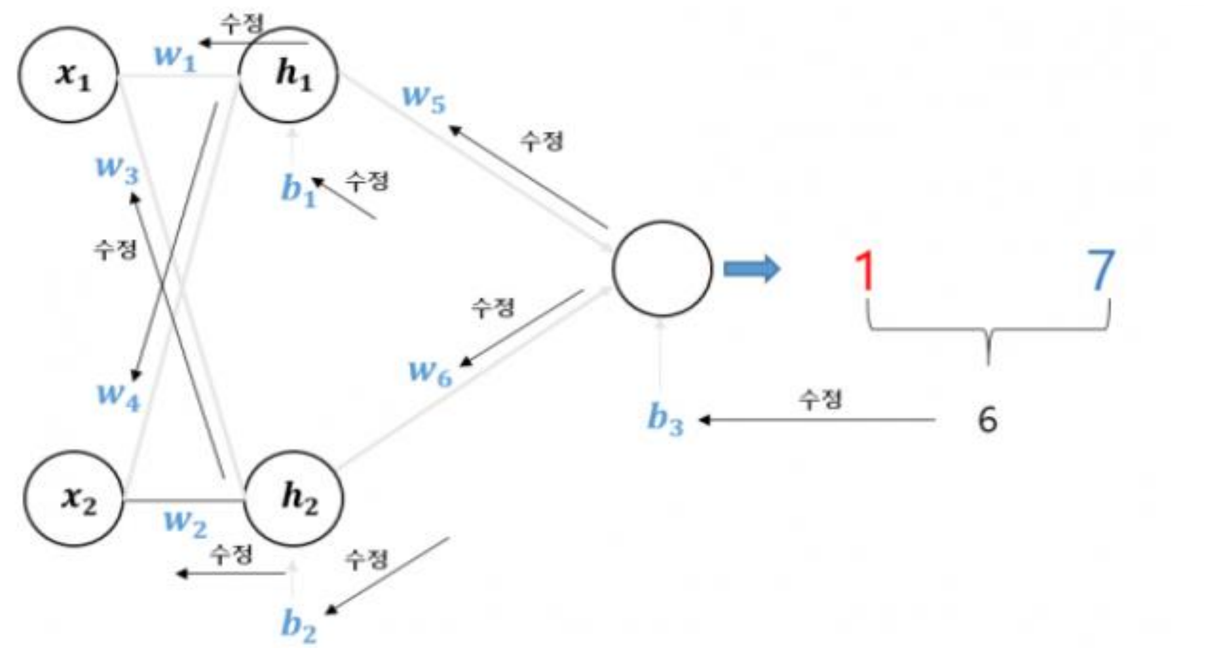
$f_{y_1,y_2}(y_1,y_2)=|J|f_{x_1,x_2}(g(y_1,y_2),h(y_1,y_2))$식에 대입

$$\boldsymbol{f_{y_1,y_2}(y_1,y_2)}=|J|f_{x_1,x_2}(y_1,-y_1+y_2)=\boldsymbol{f_{x_1,x_2}(y_1,-y_1+y_2)}$$

Back Propagation

Back Propagation

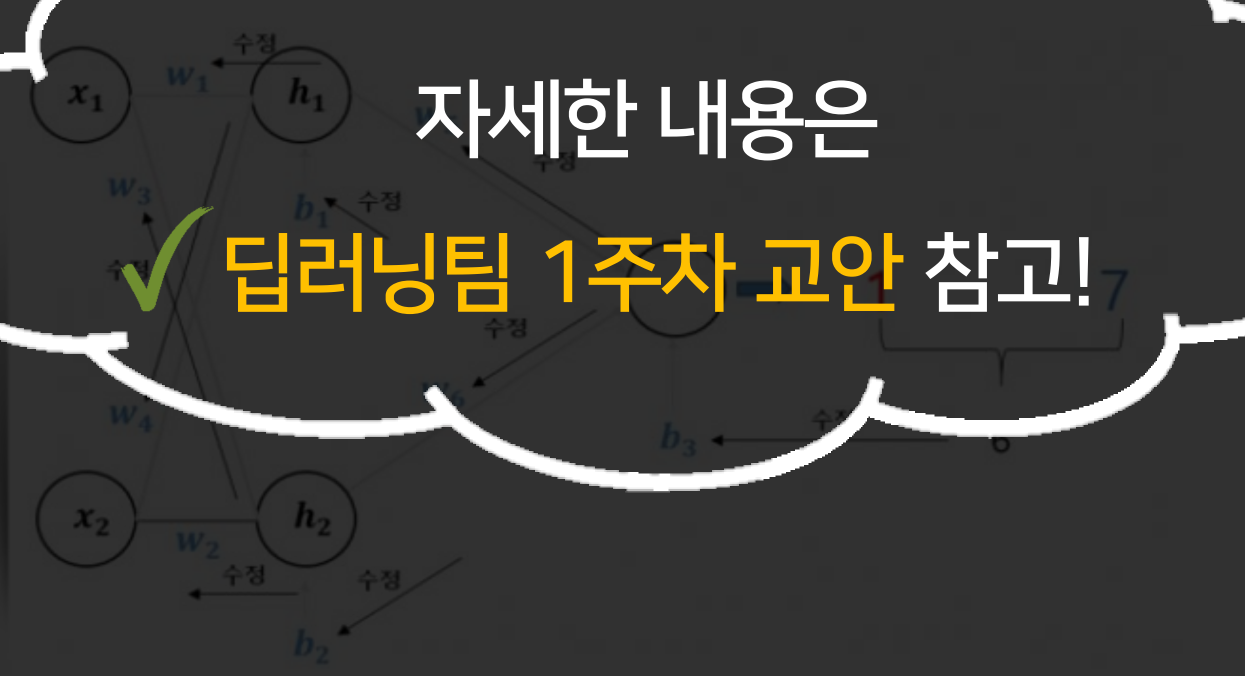
Back Propagation(역전파) 과정에서 각 파라미터에 대한 gradient를 구하게 됨



Back Propagation

Back Propagation

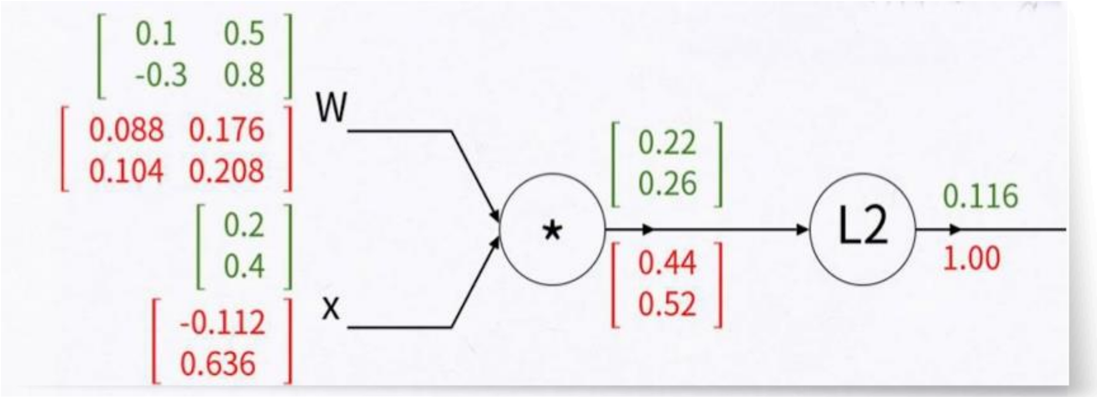
Back Propagation(역전파) 과정에서 파라미터에 대한 gradient를 구하게 됨



자세한 내용은

✓ 딥러닝팀 1주차 교안 참고!

Back Propagation



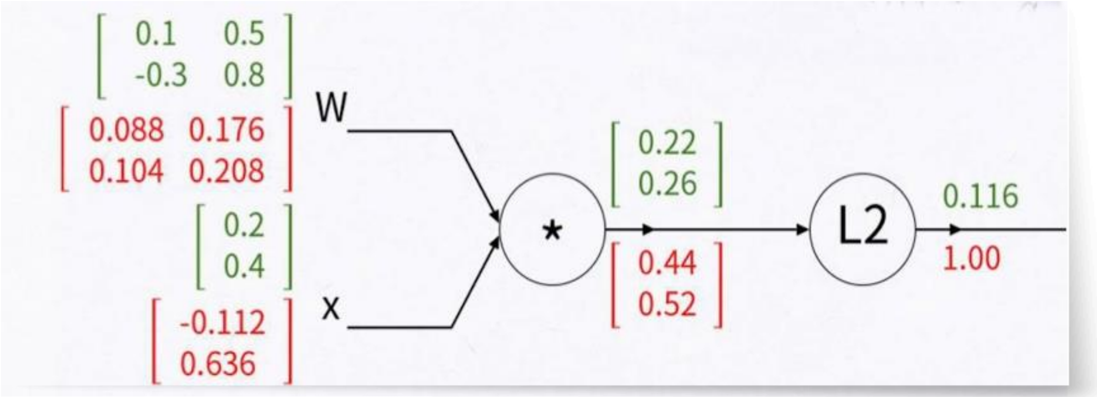
*은 행렬의 곱 연산, $L2$는 Input 값을 제곱하는 함수를 의미

*을 이용해 연산하는 중간 결과 식 $q = W \cdot x$

최종 결과 식 $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^{n} (W \cdot x)_i^2$

역전파 과정에서 $\nabla_W f, \nabla_x f, \nabla_q f$를 구하게 됨

Back Propagation



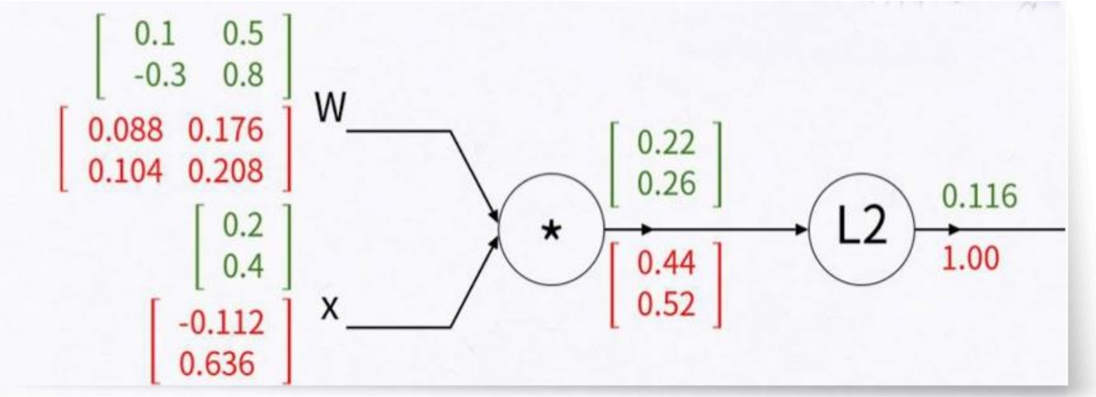
1) $\nabla_q f$ 구하기

$f(q) = ||q||^2 = {q_1}^2 + {q_2}^2 + \dots + {q_n}^2$이므로 $\frac{\partial f}{\partial q_i} = 2q_i$

즉, 모든 i에 대해서 위의 등식이 성립하므로 $\boldsymbol{\nabla_q f = 2q}$

따라서, q가 $\begin{bmatrix} 0.22 \\ 0.26 \end{bmatrix}$일 때, $\nabla_q f = \begin{bmatrix} 0.44 \\ 0.52 \end{bmatrix}$

Back Propagation

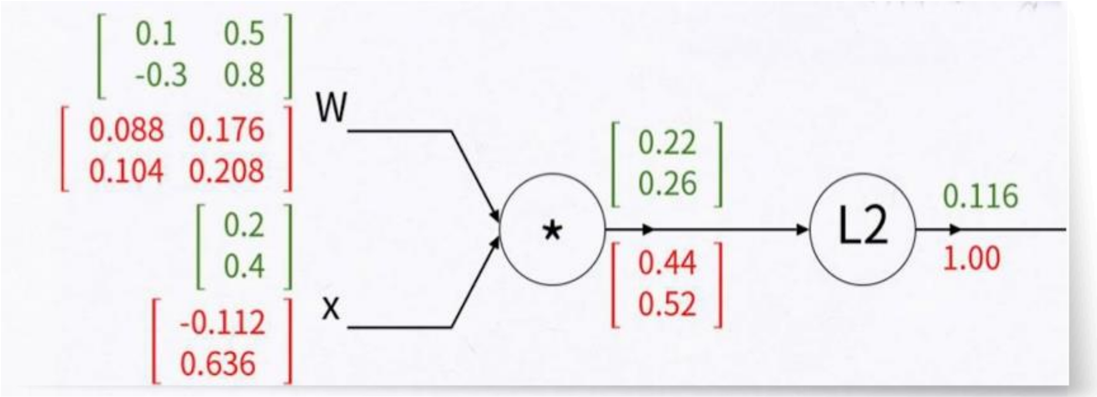


2) $\nabla_w f$ 구하기

$$\frac{\partial f}{\partial W_{ij}} = \sum_k \frac{\partial f}{\partial q_k} \cdot \frac{\partial q_k}{\partial W_{ij}}$$ 에서 $q = W \cdot x$이므로 $\frac{\partial q_k}{\partial W_{ij}} = 1_{k=i} x_j$

$$(\because\ q = W \cdot x = \begin{bmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{bmatrix})$$

Back Propagation



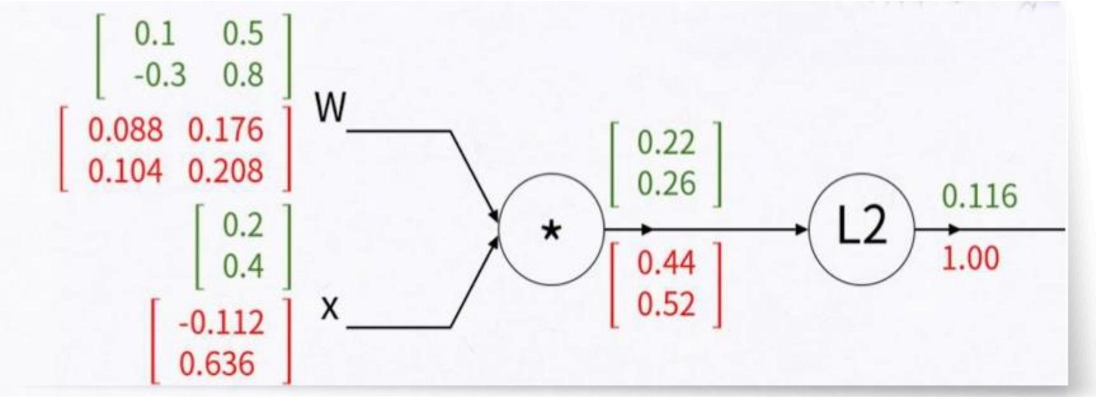
2) $\nabla_w f$ 구하기

따라서, $\frac{\partial f}{\partial W_{ij}} = \sum_k \frac{\partial f}{\partial q_k} \cdot \frac{\partial q_k}{\partial W_{ij}} = \sum_k 2q_k \cdot (1_{k=i} x_j) = 2q_i x_j$이므로 $\boldsymbol{\nabla_W f = 2q \cdot x^T}$

즉, q와 x가 각각 $\begin{bmatrix} 0.22 \\ 0.26 \end{bmatrix}$, $\begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix}$로 주어졌을 때

$$\nabla_W f = \begin{bmatrix} 0.44 \\ 0.52 \end{bmatrix} [0.2 \quad 0.4] = \begin{bmatrix} 0.088 & 0.176 \\ 0.104 & 0.208 \end{bmatrix}$$

Back Propagation

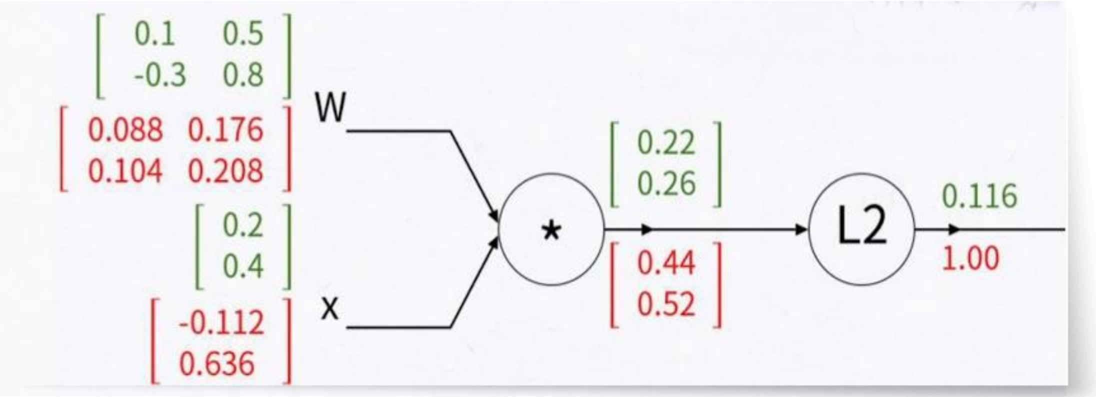


3) $\nabla_x f$ 구하기

$$\frac{\partial f}{\partial x_i} = \sum_k \frac{\partial f}{\partial q_k} \cdot \frac{\partial q_k}{\partial x_i}\text{에서 } q = W \cdot x\text{이므로 } \frac{\partial q_k}{\partial x_i} = W_{k,i}$$

$$(\because\ q = W \cdot x = \begin{bmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{bmatrix})$$

Back Propagation



3) $\nabla_x f$ 구하기

따라서, $\frac{\partial f}{\partial x_i} = \sum_k \frac{\partial f}{\partial q_k} \cdot \frac{\partial q_k}{\partial x_i} = \sum_k 2q_k \cdot W_{k,i}$이므로 $\boldsymbol{\nabla_x f = 2W^T \cdot q}$

즉, q와 W가 각각 $\begin{bmatrix} 0.22 \\ 0.26 \end{bmatrix}, \begin{bmatrix} 0.1 & 0.5 \\ -0.3 & 0.8 \end{bmatrix}$로 주어졌을 때,

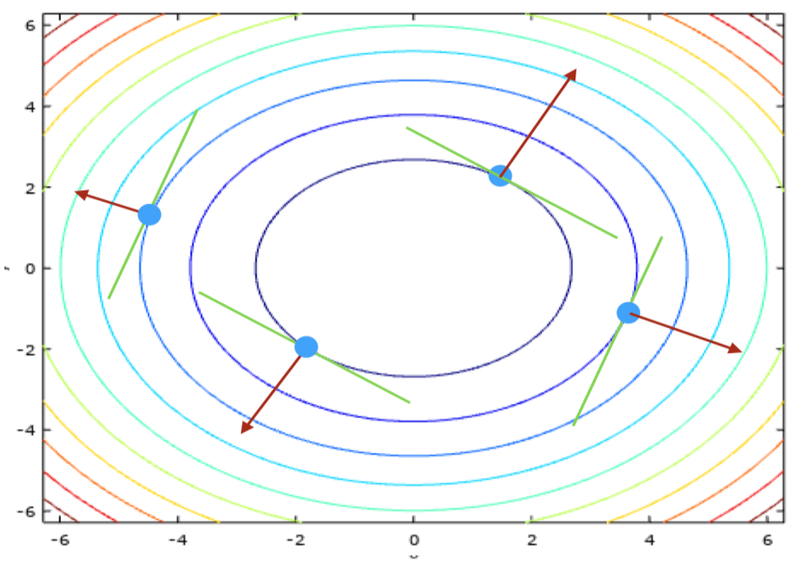
$$\nabla_x f = \begin{bmatrix} 0.2 & -0.6 \\ 1.0 & 0.8 \end{bmatrix} \begin{bmatrix} 0.22 \\ 0.26 \end{bmatrix} = \begin{bmatrix} -0.112 \\ 0.636 \end{bmatrix}$$

Gradient Descent

Gradient Ascent

등고선에서 가장 빠르게 **증가**하는 방향으로 이동

gradient가 가리키는 방향으로 움직여 최대값을 찾는 과정

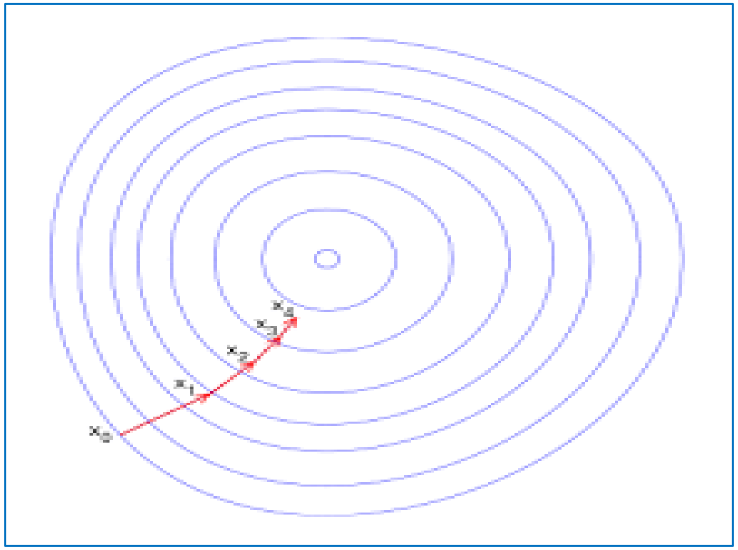


Gradient Descent

Gradient Descent

등고선에서 가장 빠르게 감소하는 방향으로 이동

Loss값을 최소화하기 위해 gradient에 (−1)을 곱해서 알고리즘 운영



Gradient Descent

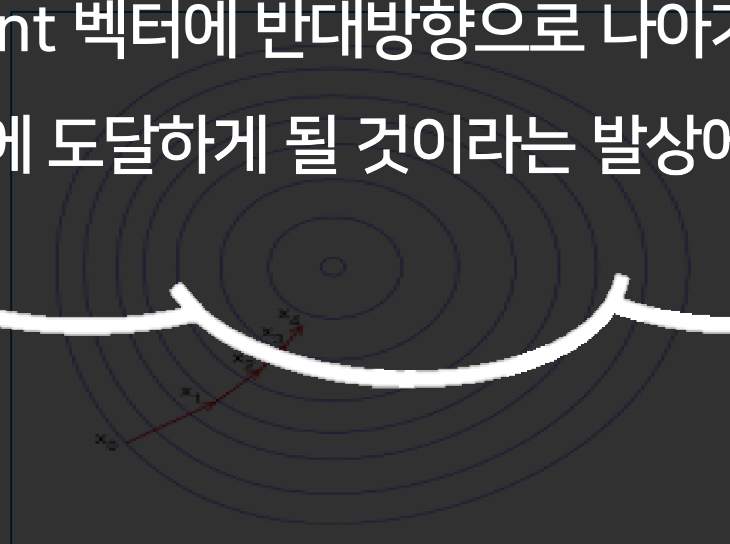
Gradient Descent

등고선에서 가장 빠르게 감소하는 방향으로 이동

Loss값을 최소화하기 위해 그라디언트에 (−1)을 곱해서 알고리즘 운영

gradient 벡터에 반대방향으로 나아가다 보면

최소값에 도달하게 될 것이라는 발상에서 출발

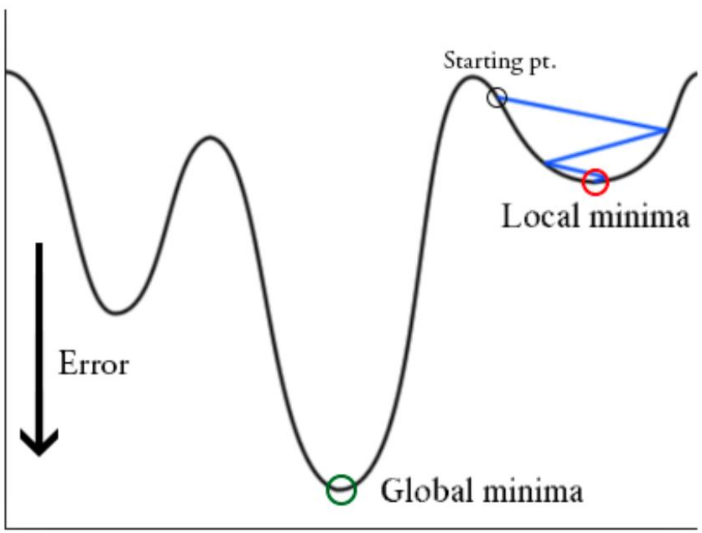


Gradient Descent

Gradient Descent

등고선에서 가장 빠르게 **감소**하는 방향으로 이동

Loss값을 최소화하기 위해 gradient에 (-1)을 곱해서 알고리즘 운영



gradient의 값이

0이 되는 지점에서 멈춤

→ 하지만 Gradient Descent 방법은

local minimum에 빠질 수 있고

속도가 느리다는 단점 존재

Gradient Descent

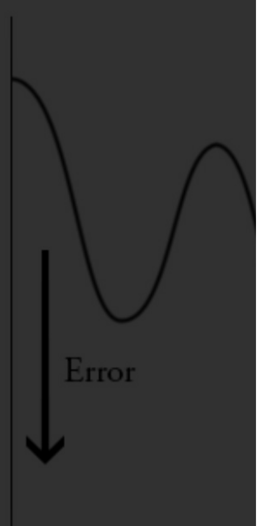
Gradient Descent의 단점을

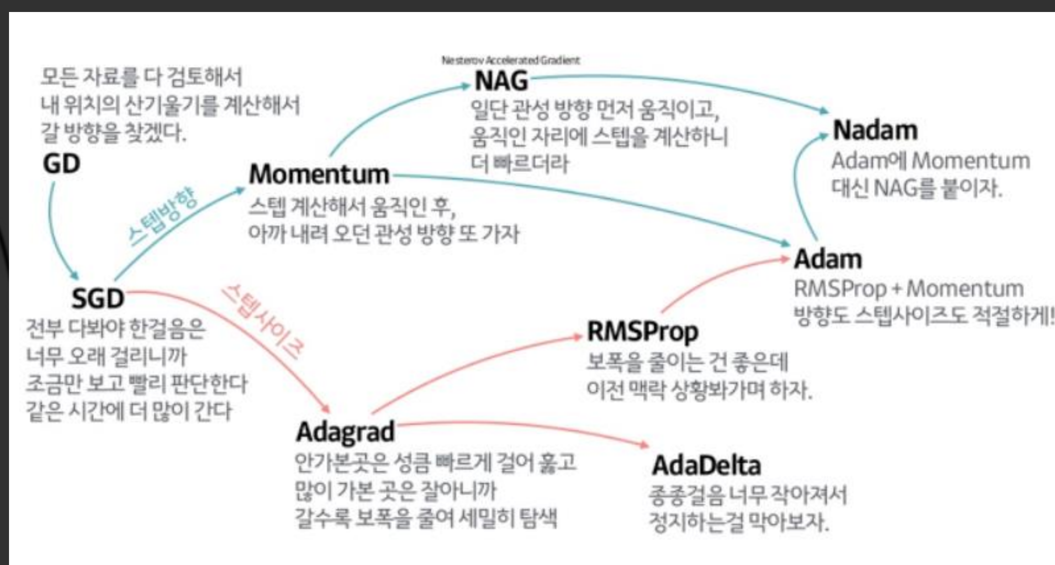
보완하기 위해 다양한 Optimizer가 등장

등고선에서 가장 빠르게 감소하는 방향으로 이동

✓ 딥러닝팀 1주차 교안 참고!

Loss값을 최소화하기 위해 그라디언트에 (−1)을 곱해서 알고리즘 운영





이

멈춤

cent 방법은

질 수 있고

점 존재

Thank You