Gradient

가중치 매개변수에 대한 손실함수의 기울기(Gradient)를 수치미분을 이용하여 구하면 계산 시간이 너무 오래 걸린다.

따라서 가중치 매개변수에 대한 기울기(Gradient)를 효율적으로 계산하는 Chain Rule을 이용한다.

Analytical Differentiation(해석적 미분)

$$f'(x) = \frac{df(x)}{dx} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} \tag{1}$$

Numerical Differentiation(수치 미분)

해석적 미분 방식으로 풀 수 없는 문제가 있을 때 수치적 접근을 통해 근사 값을 찾는 방식

컴퓨터에서는 $\lim_{\Delta x \to 0}$과 같이 무한소 형태로 계산하는 것이 불가능하다. 따라서 수치미분을 통해 근사치로 계산한다.

$$f'(x) \approx \frac{f(x+\Delta x) - f(x)}{\Delta x} \tag{2}$$

실제값과 근사치의 오차를 줄이기 위해서 중앙차분을 사용한다

$$f'(x) \approx \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} \tag{3}$$

> https://blog.naver.com/PostView.naver?blogId=mykepzzang&logNo=220069937244&parentCategoryNo=&categoryNo=16&viewDate=&isShowPopularPosts=false&from=postView

하지만 h에 매우 작은 값을 주어도 실제값과 오차가 생기며 hidden layer가 많아지게 되면 미분 횟수도 늘어나므로 오차가 누적될 수도 있다. 이 뿐만 아니라 컴퓨터가 소수를 저장하는 부동소수점방식에 의해서도 오차가 생기며 하나 하나 편미분으로 계산하여야 하기 때문에 연산 속도도 느리다.

그래서 다음과 같은 미분 방식이 나오게 된다.

Symbolic differentiation(기호 미분)

기호 미분은 인간에게 가장 익숙한 미분법이다. 간단한 함수라면 쉽게 미분할 수 있다. (sympy)

$$\frac{d}{dx}(\frac{x^2cos(x-7)}{sin(x)}) = \\ x^2sin(7-x)csc(x)+x^2(-cos(7-x))cot(x)csc(x)+2xcos(7-x)csc(x) \tag{4}$$

또한 매우 복잡한 수식의 경우 미분이 매우 어려우며 미분을 하였더라도 수식에 대한 간략화가 필요하며 이를 컴퓨터로 구현하기엔 너무 복잡하며 어렵다. 따라서 이 미분법을 머신러닝에선 주된 방법으로 사용하지 않는다.

Automatic Differentiation(AD- 자동 미분)

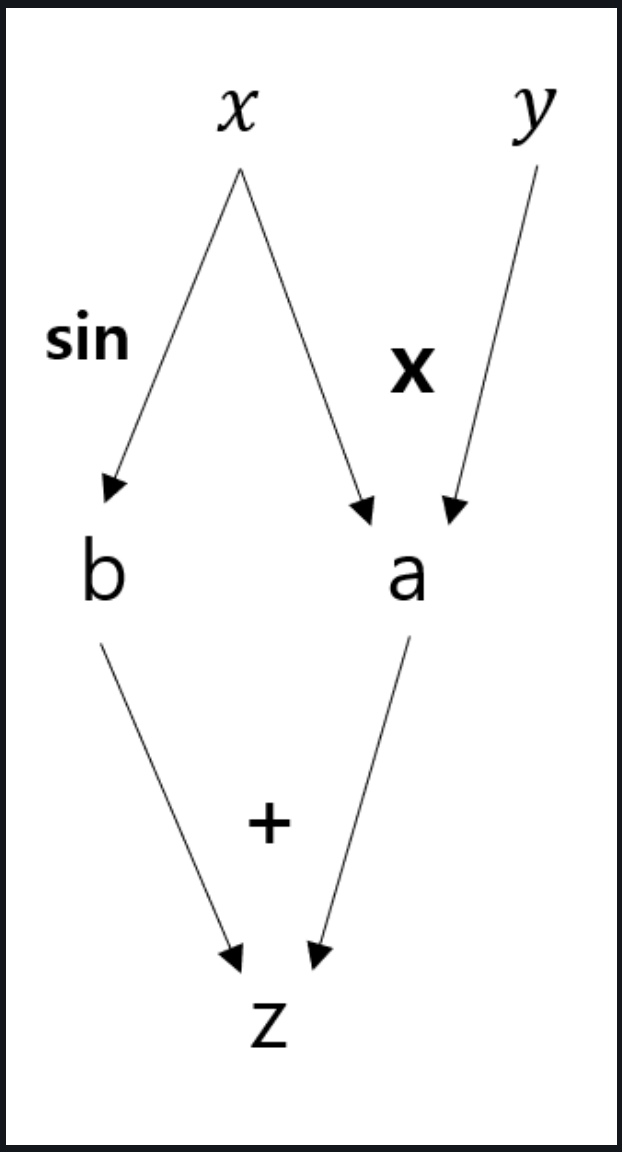
$$sin(a+b*c) = sin(add(mul(b,c),\ a)) \tag{5}$$

어떤 복잡한 함수의 개형이라도 연속적인 합성함수의 꼴로 나타낼 수 있다는 것을 아이디어로 도함수를 구하는 방법이다.

이해를 돕기위해 예를 들어 보겠다.

$$z = x \cdot y + \sin(x) \\ x = ?,\ y = ? \\ a = x \cdot y,\ b = \sin(x),\ \ z = a + b \tag{6}$$

그래프로 보면 아래와 같다.



< auto_diff >

임의의 변수 t에 대해서 편미분을 해본다.

$$
\begin{aligned}
\frac{\partial x}{\partial t} &= ?, \quad \frac{\partial y}{\partial t} = ? \\
\frac{\partial a}{\partial t} &= y \cdot \frac{\partial x}{\partial t} + x \cdot \frac{\partial y}{\partial t} \\
\frac{\partial b}{\partial t} &= \cos(x) \cdot \frac{\partial x}{\partial t} \\
\frac{\partial z}{\partial t} &= \frac{\partial a}{\partial t} + \frac{\partial b}{\partial t}
\end{aligned}
\tag{7}
$$

만약 여기서 $t = x?,\ t = y?$ 라면 어떻게 될까?

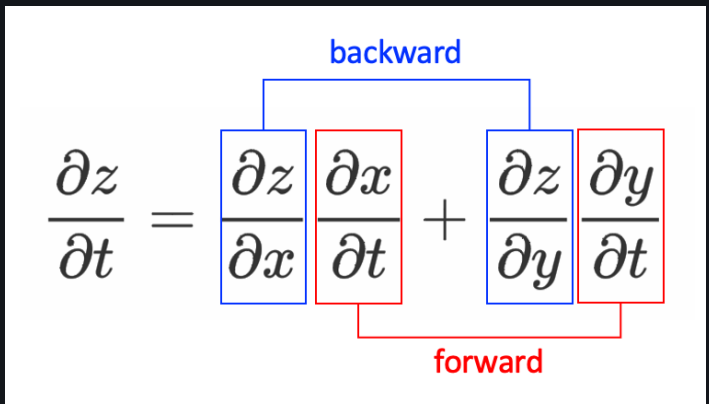
$$\frac{\partial z}{\partial t} = y \cdot \frac{\partial x}{\partial t} + x \cdot \frac{\partial y}{\partial t} + \cos(x) \cdot \frac{\partial x}{\partial t}$$

$$if \quad t = x \tag{8}$$
$$\frac{\partial z}{\partial x} = y + \cos(x) \tag{9}$$
$$\tag{10}$$
$$if \quad t = y \tag{11}$$
$$\frac{\partial z}{\partial y} = x \tag{12}$$

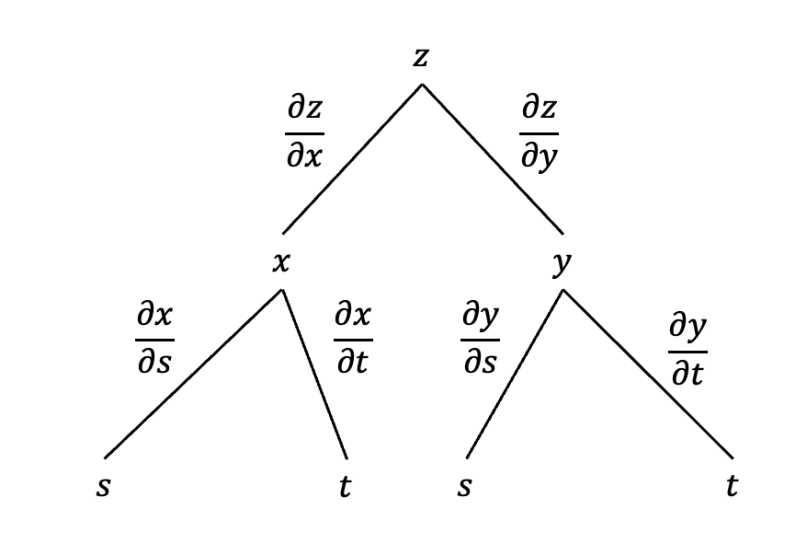
이렇듯 합성함수 미분법으로 하나하나 풀어 나갈 수 있다. 우리는 위 방법을 foward로 진행한 것이다.

$$z = f(x, y), \quad x = g(s, t), \quad y = h(s, t) \tag{13}$$

$$\frac{\partial z}{\partial t} = \frac{\partial z}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial z}{\partial y}\frac{\partial y}{\partial t} \tag{14}$$



< auto_diff >



< auto_diff >

위에서 보았듯이 미분 결과는 동일하다. 그럼 forward와 backward중 어떤게 더 좋은걸까?

정답은 backward의 경우 Input(입력)차원이 output(출력)차원보다 많은 경우 (n > m)

forward의 경우 Input(입력)차원이 output(출력)차원보다 적은경우이다. (n < m)

보통 최적화 문제에서는 (m < n) 인 경우가 더 많다.

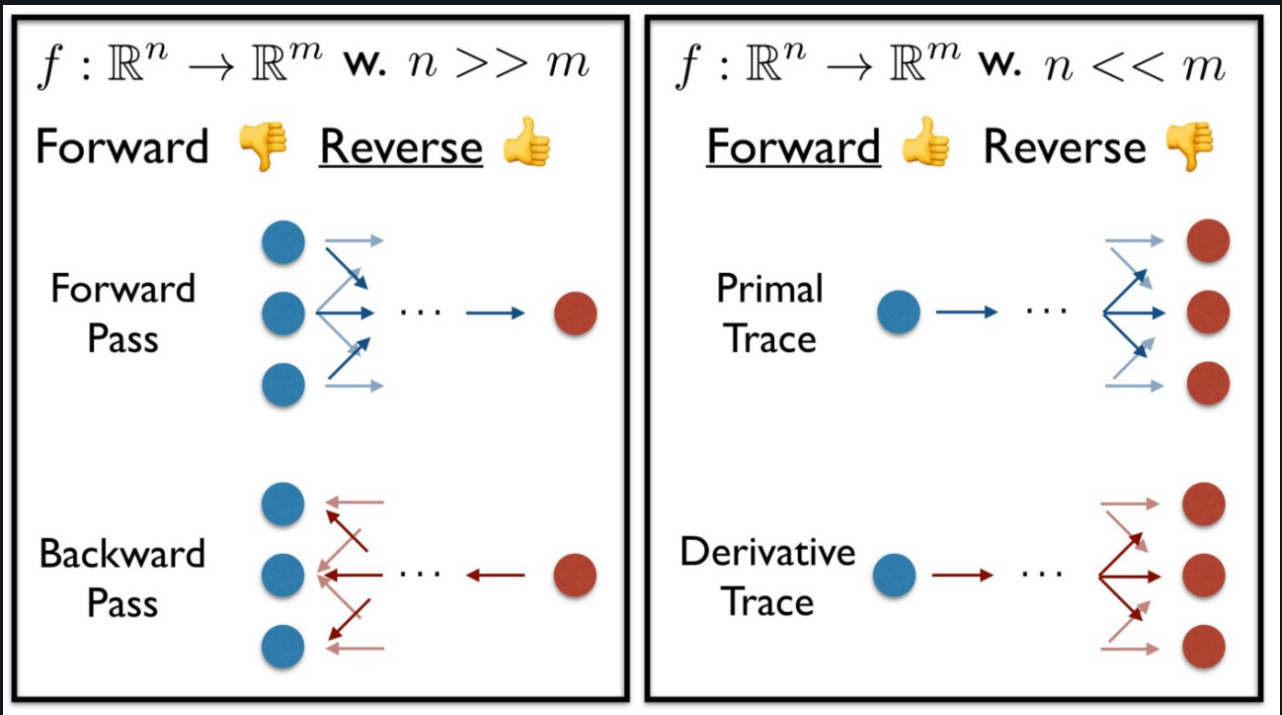
최적화라는 것이 어떤 특정 함수 f에 대해서 이 함수가 최대 또는 최소가 되는 독립변수(x, y)값을 구하는 것이 목적이기 때문이다.

$$f(x_1, x_2, \ldots, x_n) = (f_1, f_2, \ldots, f_m)$$

$$J_f = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

< auto_diff >

아래 그림을 보면 더 이해가 쉬울 것이다.



< auto_diff >

https://www.youtube.com/watch?v=XG73maPwDl8

https://towardsdatascience.com/forward-mode-automatic-differentiation-dual-numbers-8f47351064bf

Chain Rule (= 연쇄법칙 = 합성함수 미분법)

어떤 변수에 대한 다른 변수의 변화율을 알아내기 위해 쓰인다.

예를들어, 변수 y가 u에 의존하고, 변수 u가 x에 의존한다고 하면 x에 대한 y의 변화율은 u에 대한 y의 변화율과 x에 대한 u의 변화율을 곱함으로써 계산할 수 있다. 아래식을 참고하자.

$$
\begin{gathered}
y = f(g(x)), \quad y = f(u), \quad u = g(x) \\
\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx}
\end{gathered} \tag{15}
$$

> https://m.blog.naver.com/alwaysneoi/100171733834

일변수 미분

변수의 방향은 왼쪽과 오른쪽 두가지만 존재한다.

$$\lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} \tag{16}$$

다변수 방향 미분

여러 방향이 존재하므로 방향 벡터를 사용해야 한다.

$$D_{\vec{u}} f(\underline{x}) = \lim_{\Delta x \to 0} \frac{f(\underline{x}+\Delta x\vec{u}) - f(\underline{x})}{\Delta x} \tag{17}$$

편도함수[1]와 편미분

각 축에 대한 미분을 편미분이라고 하며 방향 미분 축에 해당하는 변수 외에는 상수취급한다.

$$
\begin{aligned}
D_{(1,0,\cdots,0,0)}f(\underline{x}) &= \frac{f(\underline{x}+\Delta x(1,0,\cdots,0,0))-f(\underline{x})}{\Delta x} \\
D_{(0,1,\cdots,0,0)}f(\underline{x}) &= \frac{f(\underline{x}+\Delta x(0,1,\cdots,0,0))-f(\underline{x})}{\Delta x} \\
&\vdots \\
D_{(0,0,\cdots,1,0)}f(\underline{x}) &= \frac{f(\underline{x}+\Delta x(0,0,\cdots,1,0))-f(\underline{x})}{\Delta x} \\
D_{(0,0\cdots,0,1)}f(\underline{x}) &= \frac{f(\underline{x}+\Delta x(0,0,\cdots,0,1))-f(\underline{x})}{\Delta x}
\end{aligned} \tag{18}
$$

Gradient

미분계수가[2] 커지는 가장 가파른 방향 의 순간변화율 혹은 기울기라고도 하며 모든 변수에 대한 편미분을 벡터로 정리한 것도 Gradient 라고 한다.

$$
\nabla f = (\frac{\partial f}{\partial w_0}, \frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial w_2}, \cdots, \frac{\partial f}{\partial w_n}) \tag{19}
$$

모든 방향벡터의 크기(Scalar)를 Gradient와 방향(단위벡터)를 내적하여 구할 수 있다.

$$
D_{\vec{u}}f(\underline{x}) = \nabla f(\underline{x}) \cdot \vec{u} = \| \nabla f(x) \| \| \vec{u} \| \cos\theta \tag{20}
$$

방향 미분 계수의 최대 최소

Gradient는 가장 가파른 방향의 기울기라고 하였다. 그럼 언제가 Gradient가 가장 가파른 기울기 일까?

(9)식의 크기가 최대가 되려면 $cos\theta$ 가 1일 때이다.

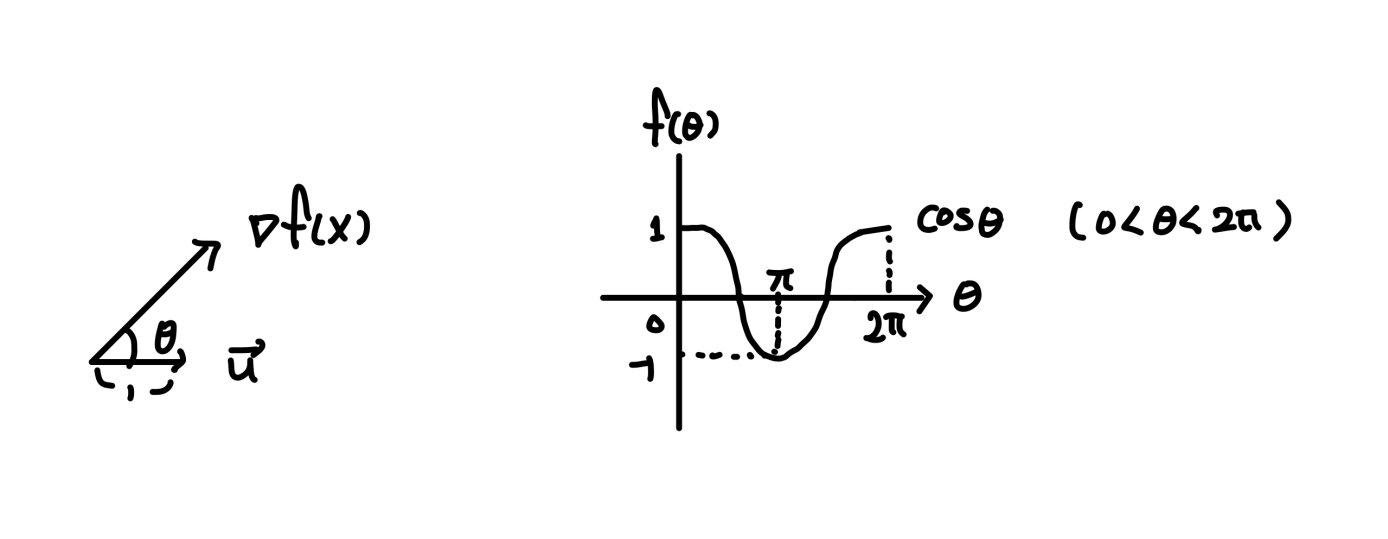
방향벡터가 Gradient와 같은 방향을 보고 있을 때가 최대가 된다.

즉 Gradient의 크기가 특정 지점에서의 미분계수가 가장 빠르게 커지는 방향의 기울기 값이다.

반대로 최소가 되려면 θ값이 π인 $\cos\theta$ 가 -1일 때이다.

이는 방향벡터가 Gradient와 반대 방향을 보고 있을 때 최소가 된다.

즉 Gradient의 크기에 `-`를 붙인 값이 특정 지점에서의 미분 계수가 가장 빠르게 작아지는 방향의 기울기 값이다.



< 방향미분계수의 최대최소 >

Gradient의 반대방향이 그 지점에서 손실함수의 값을 가장 빠르게 줄이는 방향(Local Minima)이지만 손실함수의 최솟값을 가르키는건 아닐 수 있다.(Global Minina)

Gradient Desent

경사 하강법은 Gradient의 반대 방향으로 한 발자국씩 내딛으면서 Loss 함수의 값을 낮추는 방법이다.

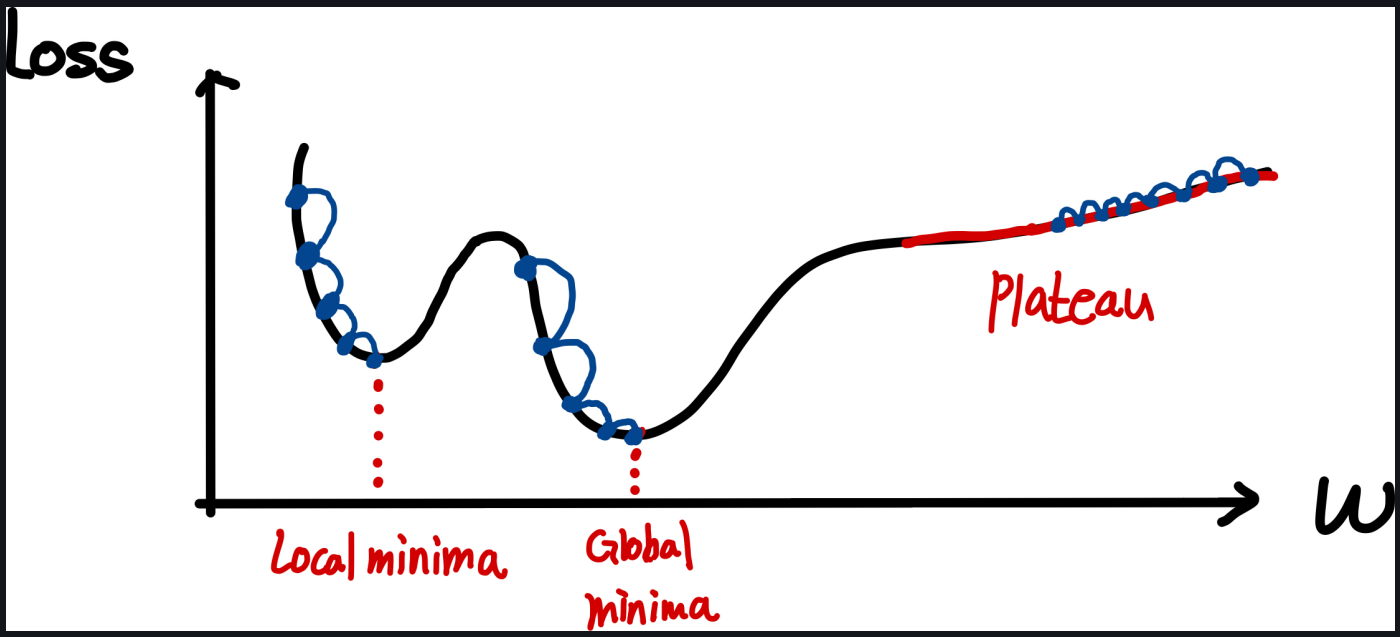
부호가 `-`인 이유는 Gradient의 반대 방향이기 때문이다.

$$W = W_0 - \eta \nabla f(\underline{x}) \tag{21}$$

경사 하강법의 문제점은 극소점이나 안장점[3] 근처에서는 Gradient의 크기가 작아서 보폭도 작아진다.

랜덤하게 결정된 초기 위치(시작점)에 따라 학습 결과가 달라질 수 있다.

아래 그림에서는 Global Minima에 도착했을 때가 가장 Loss값이 작아지는 지점으로 모델의 학습이 가장 잘 된 경우라 볼 수 있다.



< 방향미분계수의 최대최소 >

Weight Updating

실제 어떻게 가중치 업데이트가 이루어지는지 알아보겠다.

시작점은 $w_1 = 1, w_2 = 1$ 이고 $\eta = 1$이라 하자.

각 변수에 대한 Gradient를 $2w_1, w_2$라고 하면 아래와 같이 쓸 수 있다.

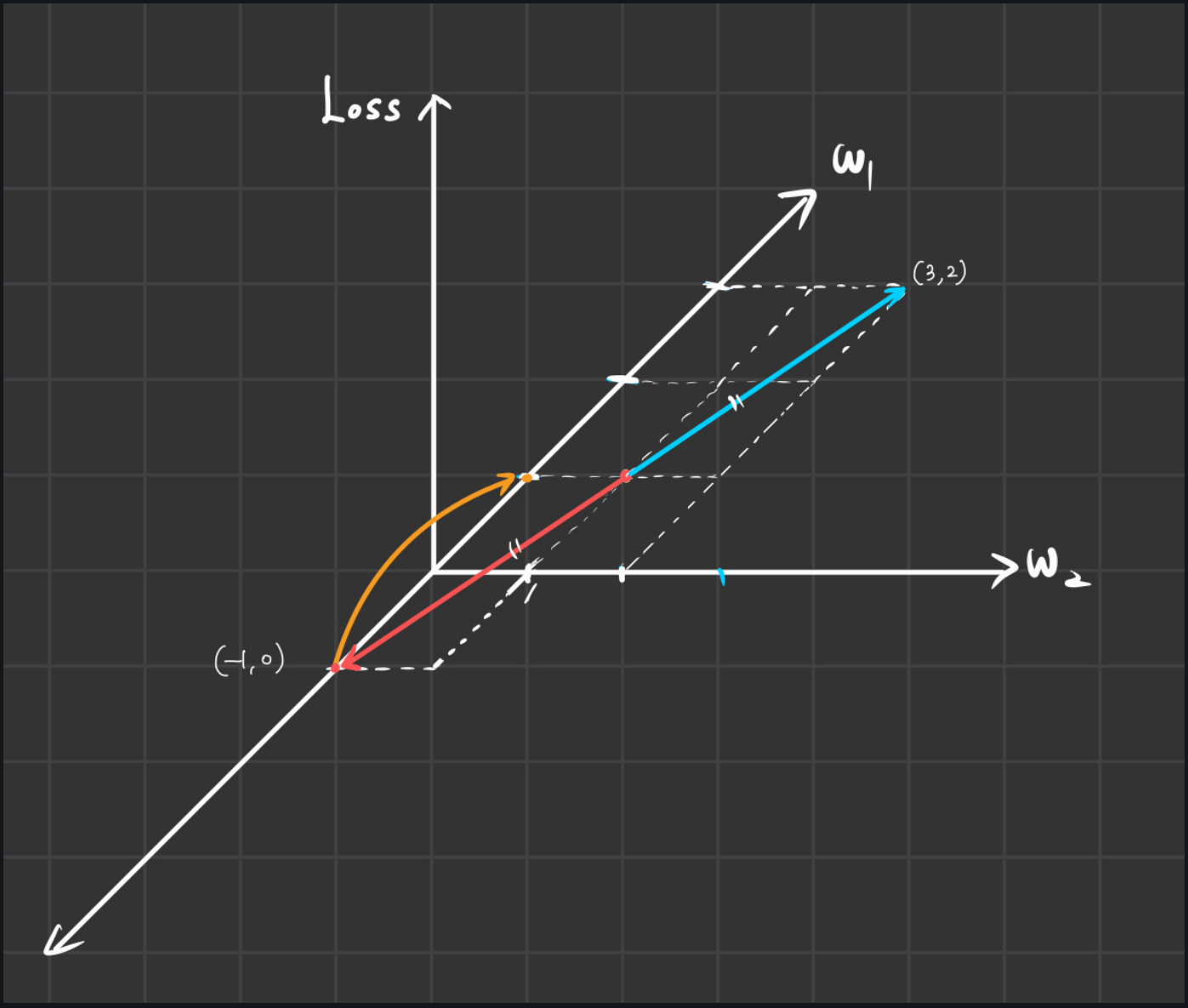
$$\nabla f = (\frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial w_2}) \tag{22}$$

$$\nabla f = (2w_1, w_2)$$

위 식에서 $w_1 = 1, w_2 = 1$을 대입하면 Gradient는 (2, 1)이 나온다. 즉 (-2, -1)방향의 기울기가 (1, 1)에서의 미분 계수가 가장 빠르게 작아지는 방향이다.

따라서 현 지점인 (1, 1)에서 (-2, -1) 방향으로 가중치 값들을 업데이트 시켜주면 $w_1, w_2 = (-1, 0)$점에 도착한다.

$$
\begin{aligned}
w_1 &= w_1 - \eta \frac{\partial f}{\partial w_1} = 1 - 2 = -1 \\
w_2 &= w_2 - \eta \frac{\partial f}{\partial w_2} = 1 - 1 = 0
\end{aligned}
\tag{23}
$$



< 방향미분계수의 최대최소 >

그럼 현 지점이 (1,1)에서 (-1, 0)으로 바뀌었다.

(-1, 0)에서의 Gradient는 `(11)` 식에 대입하면 $\nabla f = (-2, 0)$가 나온다.

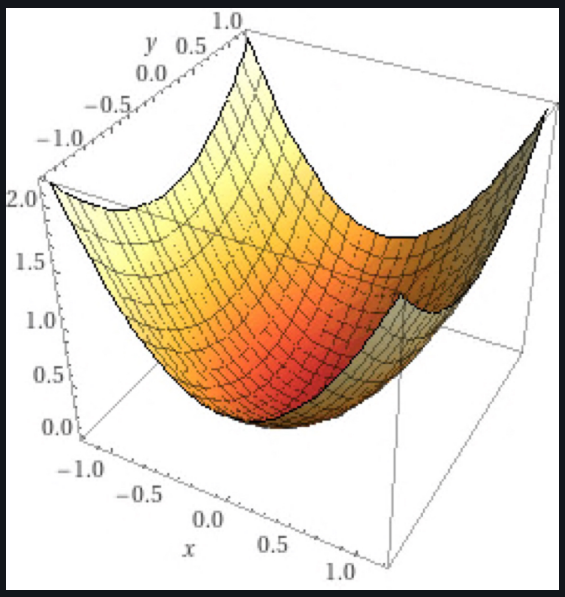
즉, (2, 0)의 방향의 기울기가 (-1, 0)에서 미분계수가 빠르게 작아지는 방향이다.

따라서 따라서 현 지점인 (-1, 0)에서 (2, 0) 방향으로 가중치 값들을 업데이트 시켜주면 $w_1, w_2 = (1, 0)$점에 도착한다.

이런 작업들을 반복하며 학습을 시켜주는 방법이 경사 하강법이다.

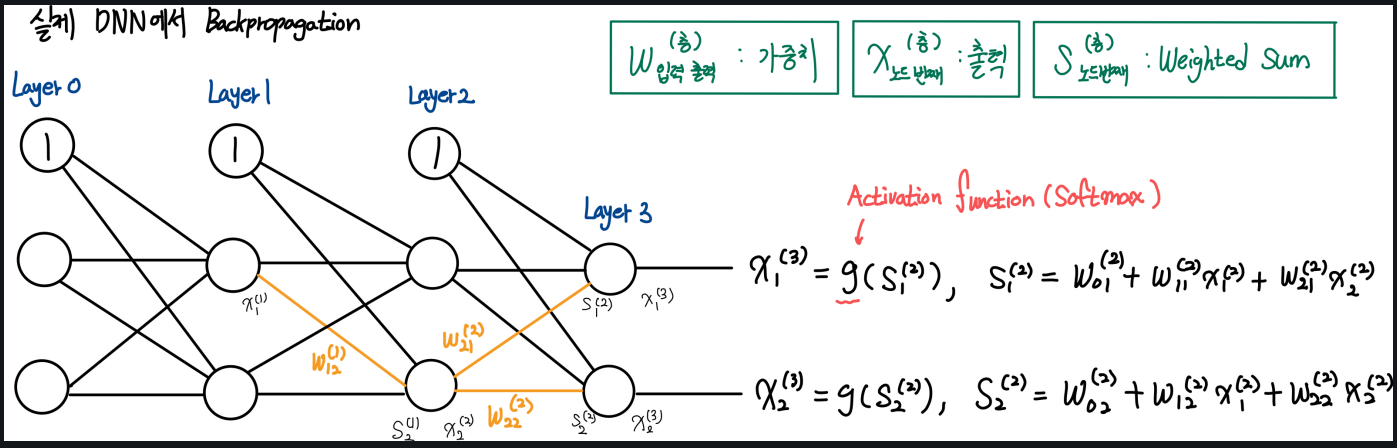
아래의 수식은 우리가 임의로 정한 Loss function이 이루는 그래프다.(실제 학습에서는 모델은 수백차원 또는 수만 차원의 깊은 층으로 이루어져 있기 때문에 그래프의 개형을 알 수 없다.)

$$y = w_1^2 + \frac{1}{2}w_2^2 \tag{24}$$



< graph_1 >

Back Propagation



<Back Propagation>

1. $w_{21}^{(2)}$에 대한 Loss function f의 미분

$$\frac{\partial f}{\partial w_{21}^{(2)}} = \frac{\partial f}{\partial x_1^{(3)}} \frac{\partial x_1^{(3)}}{\partial s_1^{(2)}} \frac{\partial s_1^{(2)}}{\partial w_{21}^{(3)}} \tag{25}$$

2. $w_{22}^{(2)}$에 대한 Loss function f의 미분

$$\frac{\partial f}{\partial w_{22}^{(2)}} = \frac{\partial f}{\partial x_2^{(3)}} \frac{\partial x_2^{(3)}}{\partial s_2^{(2)}} \frac{\partial s_2^{(2)}}{\partial w_{22}^{(2)}} \tag{26}$$

3. $w_{12}^{(1)}$에 대한 Loss function f의 미분

$$\frac{\partial f}{\partial w_{12}^{(1)}} = \frac{\partial f}{\partial x_1^{(3)}} \frac{\partial x_1^{(3)}}{\partial s_1^{(2)}} \frac{\partial s_1^{(2)}}{\partial x_2^{(2)}} \frac{\partial x_2^{(2)}}{\partial s_2^{(1)}} \frac{\partial s_2^{(1)}}{\partial w_{12}^{(1)}} + \frac{\partial f}{\partial x_2^{(3)}} \frac{\partial x_2^{(3)}}{\partial s_2^{(2)}} \frac{\partial s_2^{(2)}}{\partial x_2^{(2)}} \frac{\partial x_2^{(2)}}{\partial s_2^{(1)}} \frac{\partial s_2^{(1)}}{\partial w_{12}^{(1)}} \tag{27}$$

1. 어떤 다변수 함수가 있을 때, 그 중 한 변수만을 기준으로 잡고 다른 변수는 상수로 간주하여 미분하였을 때 얻는 도함수 ↩

2. 계수는 y=f(x) 의 x=a에서의 접선의 기울기 를 의미 ↩

3. 다변수함수의 그래프에서 나타난다. 어떤 점에서 어느 한 축에서는 극대에 이르면서 다른 축에서는 극소에 이르는 점. ↩