

Optimization

모두의연구소
박은수 Research Director

지난시간 돌아보기 ...

- **#lec_tools**
 - Slack, Git
 - 아직 익숙하지 않으신 분들을 위해 **#slack_tutorial**
 - Git



지난시간 돌아보기 ...

- #lec_math

Branch: master ▾ Lectures / 00_math /

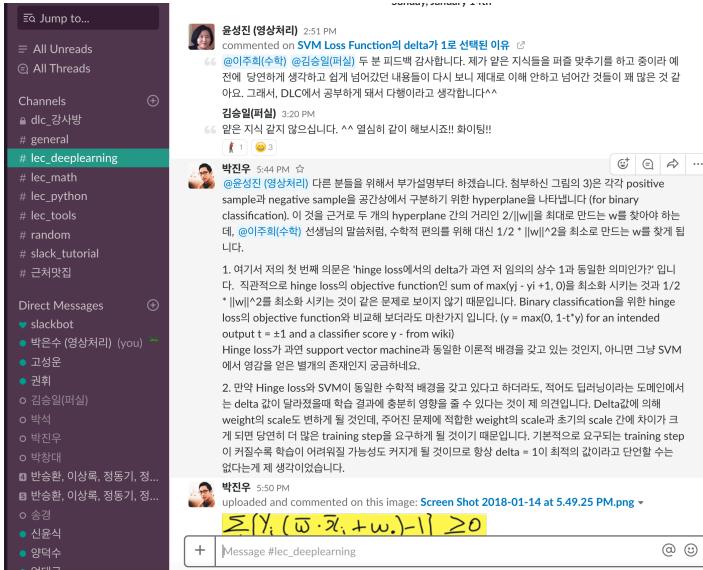
Create new file Upload files Find file History

...	Latest commit d465df6 2 days ago
cookie2016 Add files via upload	
..	
README.md	Update README.md 6 days ago
week1 introduction.pdf	Add files via upload 2 days ago
딥러닝과 수학_1강_LA.pdf	Add files via upload 6 days ago

강의 슬라이드 뿐만아니라 그 설명도 볼 수 있습니다

지난시간 돌아보기 ...

- #lec_deeplearning



엄청난 논의들

환영합니다

지난시간 돌아보기 ...

- #lec_deeplearning

Distance 를 측정

L1 distance: $d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$

test image				training image				pixel-wise absolute value differences			
56	32	10	18	10	20	24	17	46	12	14	1
90	23	128	133	8	10	89	100	82	13	39	33
24	26	178	200	12	16	178	170	12	10	0	30
2	0	255	220	4	32	233	112	2	32	22	108

- = add → 456

가장 간단한
Distance 측정

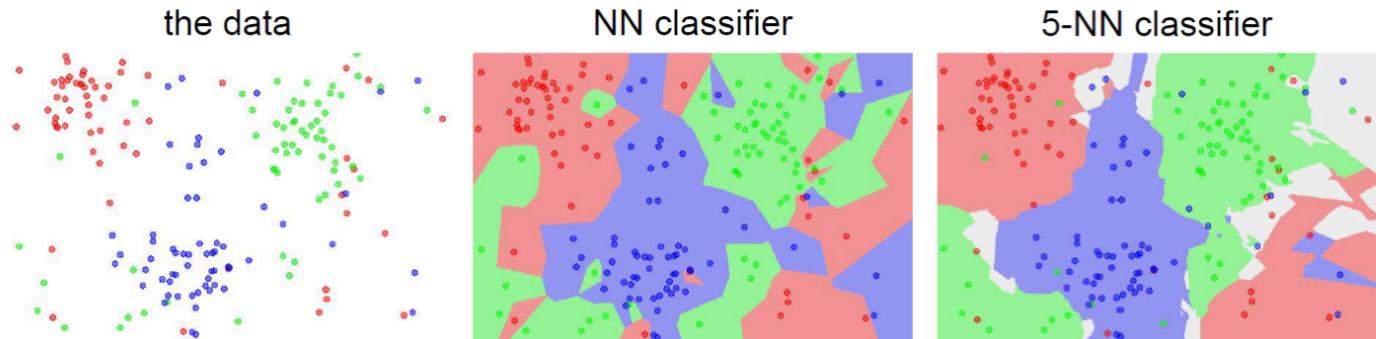
지난시간 돌아보기 ...

- **#lec_deeplearning**

k-Nearest Neighbor은 무엇인가?

k개의 Nearest Neighbor를 찾고, k개 중 많이 나온 레이블을 선택 (voting)

처음 만난
이미지 분류기



지난시간 돌아보기 ...

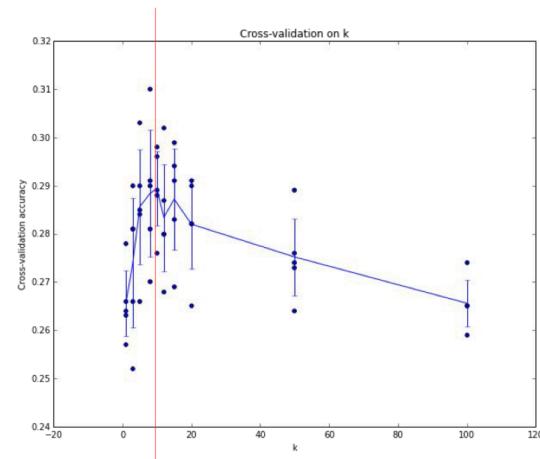
- ## • #lec_deeplearning



트레이닝(training)에 사용 할 데이터의 일정 부분을 떼어 내어 하이퍼파라미터를 튜닝하자

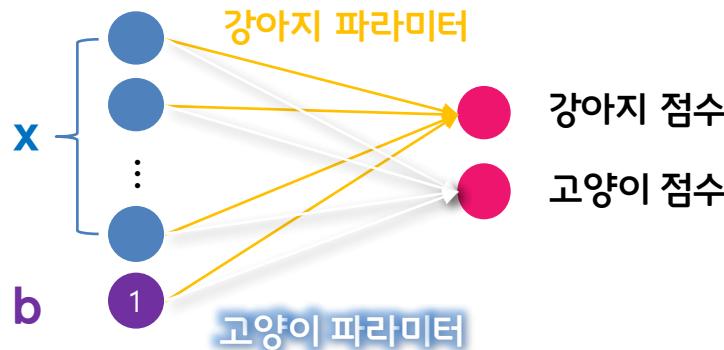
즉, fold1, fold2, fold3, fold4를 트레이닝에 사용하고 테스트를 fold5에 수행해본다

Cross validation



음.. $k=7$ 이 좋겠군

지난시간 돌아보기 ...



선형분류기

강아지 파라미터
고양이 파라미터

0.2	0.1	...	0.3	0.7
0.7	0.8	...	0.9	0.4

W

32x32x3 영상 =
3,072 픽셀



$$f(x, W) = Wx + b$$

2×1 $3,072 \times 1$
 $2 \times 3,072$ 2×1

155
200
...
110
78

x

0.1
0.2

b

0.3
0.7

강아지 점수
고양이 점수

지난시간 돌아보기 ...

- #lec_deeplearning

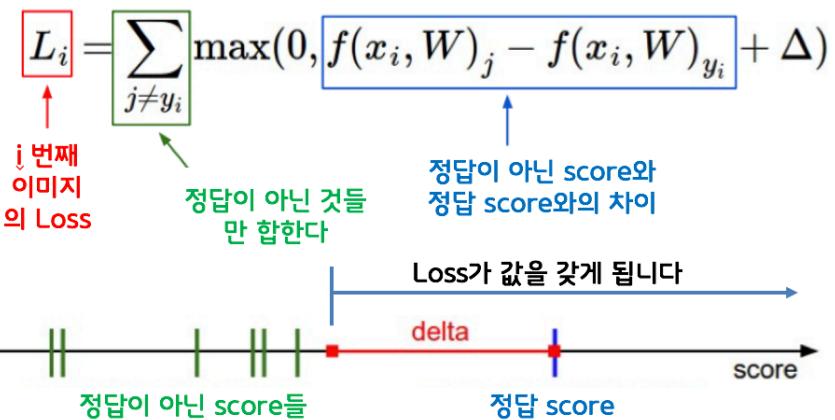
Suppose: 3 training examples, 3 classes.

With some W the scores $f(x, W) = Wx$ are:



cat	3.2	1.3	2.2
car	5.1	4.9	2.5
frog	-1.7	2.0	-3.1
Losses:	2.9	0	12.9

SVM loss



- 최소한 Loss가 0이 아닌 값을 가지려면 정답이 아닌 녀석들이 저 붉은 delta 위치에는 오거나 정답 Score보다 크면 되겠네요

지난시간 돌아보기 ...

- #lec_deeplearning



강아지 점수	2.3	\exp	9.97
고양이 점수	-1.2		0.3

전부다 양수가 됨

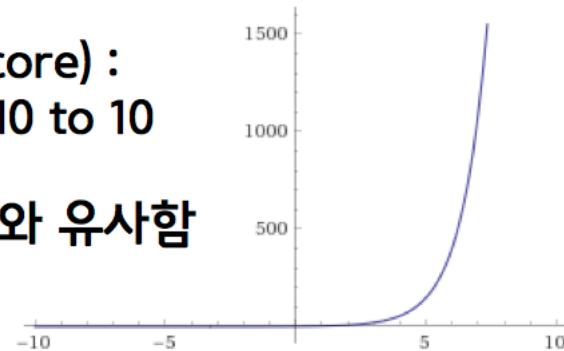
Softmax loss

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

Softmax

exp(score) :
from -10 to 10

Max함수와 유사함



지난시간 돌아보기 ...

- #lec_deeplearning Softmax loss



$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

강아지 점수

2.3

\exp

9.97

normalize

0.97

고양이 점수

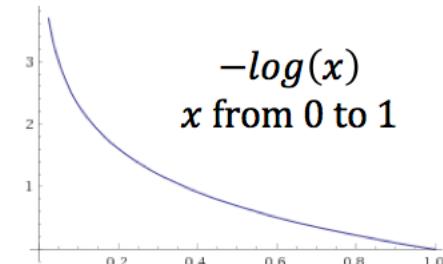
-1.2

0.3

0.03

전부다 양수가 됨

확률 $L_i = -\log(0.03) = 3.5$



지난시간 돌아보기 ...

- #lec_deeplearning

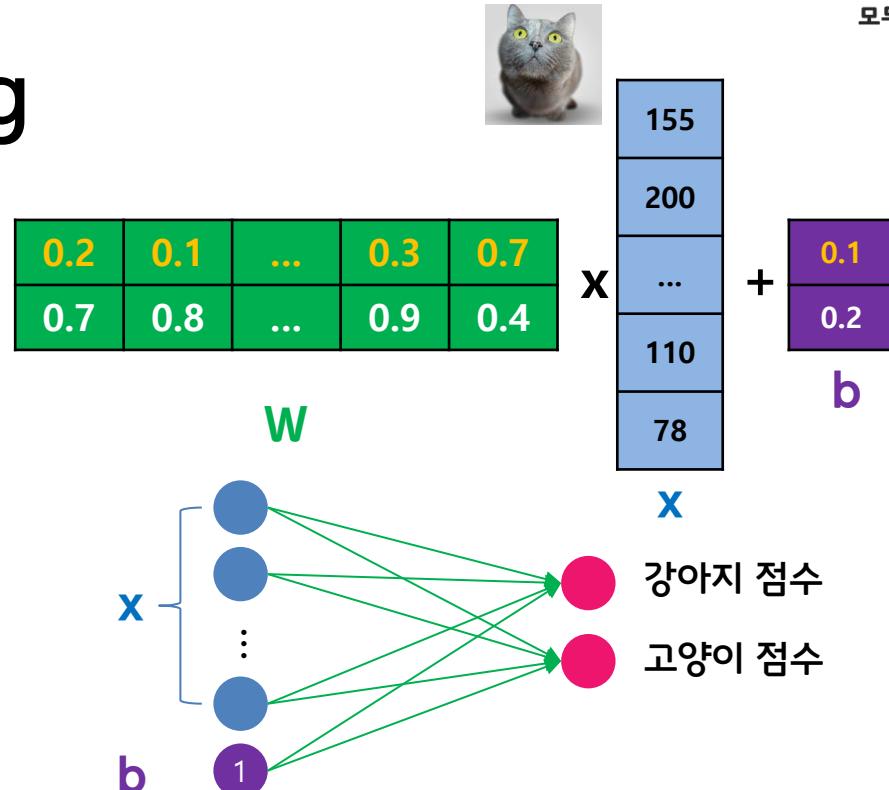
- 분류기의 구성

- Score function

- Loss function

- Optimization

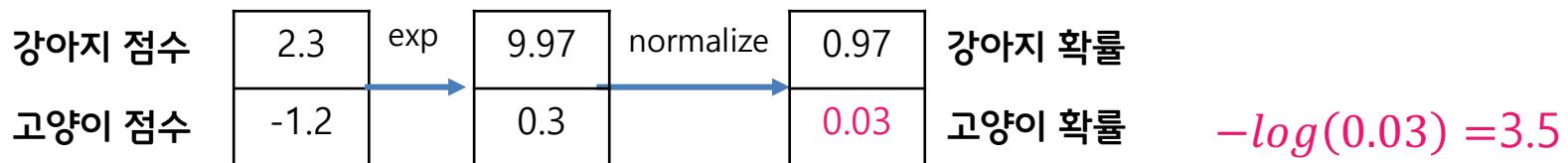
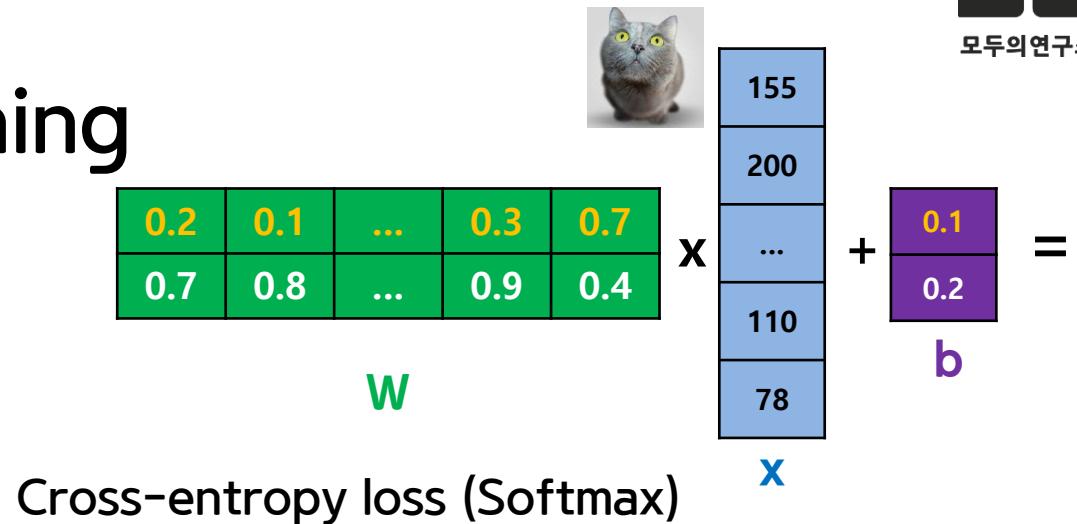
고양이가 입력이면 고양이
점수가 높아야 함



지난시간 돌아보기 ...

- #lec_deeplearning

- 분류기의 구성
 - Score function
 - Loss function
 - Optimization



현재의 분류기는 3.5만큼 안 좋음. 이 loss 값을 줄이는게 목표

지난시간 돌아보기 ...

- #lec_deeplearning
- 분류기의 구성
 - Score function
 - Loss function
 - Optimization



$$\text{cat image} \quad \begin{matrix} 155 \\ 200 \\ \dots \\ 110 \\ 78 \end{matrix} \times \begin{matrix} 0.2 & 0.1 & \dots & 0.3 & 0.7 \\ 0.7 & 0.8 & \dots & 0.9 & 0.4 \end{matrix} + \begin{matrix} 0.1 \\ 0.2 \end{matrix} =$$

오늘의 주제

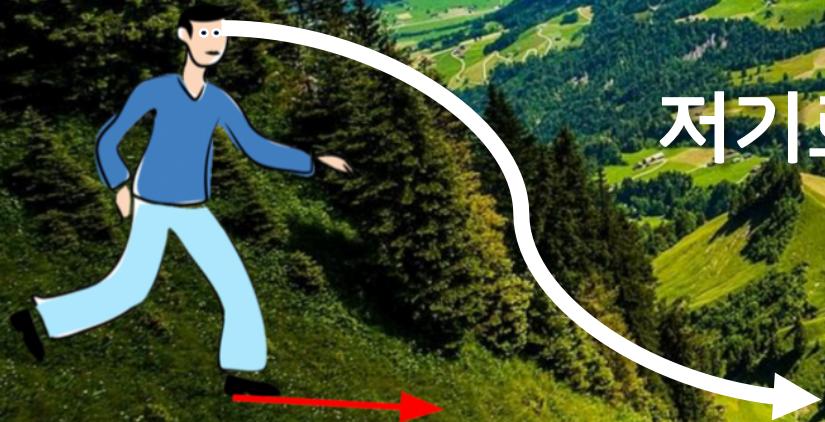
Loss를 최소화하는 w와 b
를 찾아라



최저점을 향해 가시오

저기 보이네~~~!!!!

저기로 가자~~!

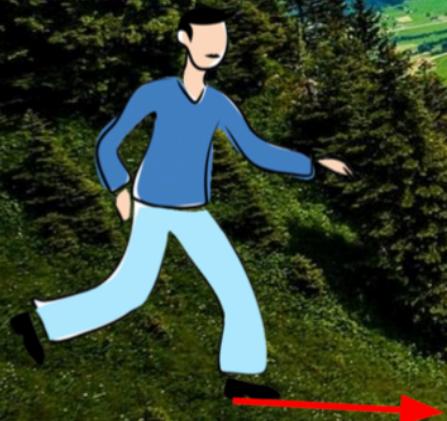




최저점을 향해 가시오

헉?!

눈이 없으면 ?



최저점을 향해 가시오

넘어질 것 같아..
어디까지 갈 수 있을까..



[대안]

발로 더듬더듬 해서 내리막이면 가자!

그냥 눈을 다시 그려줘 ...

내리막을 찾는 방법

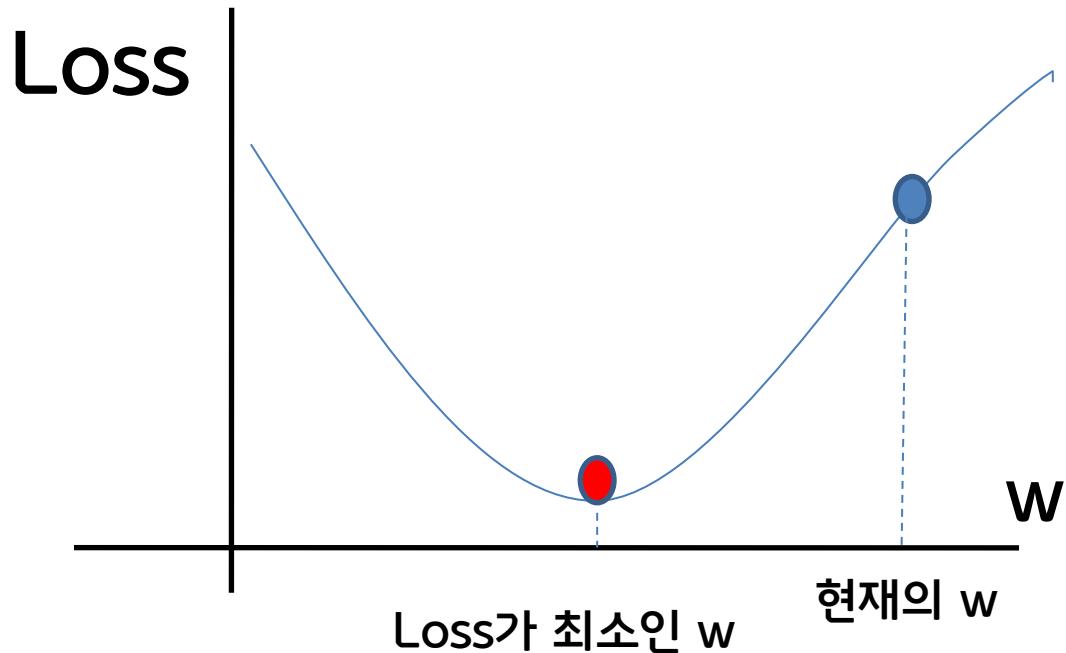


내리막 ?? == 기울기 ??

기울기를 따라 내려 가보자

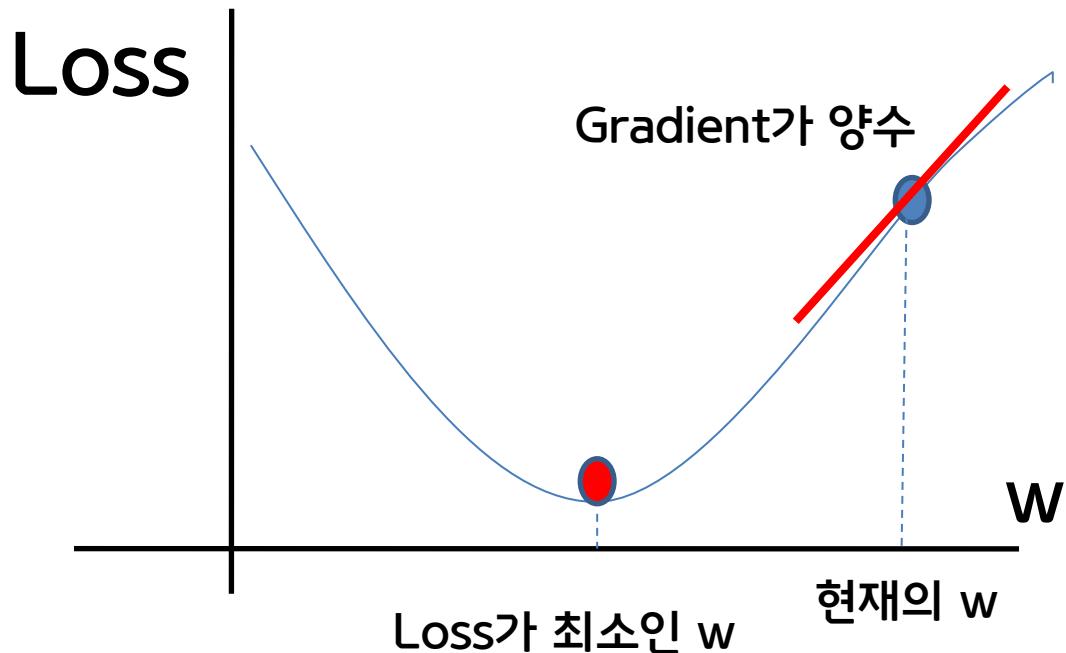
Gradient Descent

Gradient Descent



$$w = w - \eta \frac{\partial L}{\partial w}$$

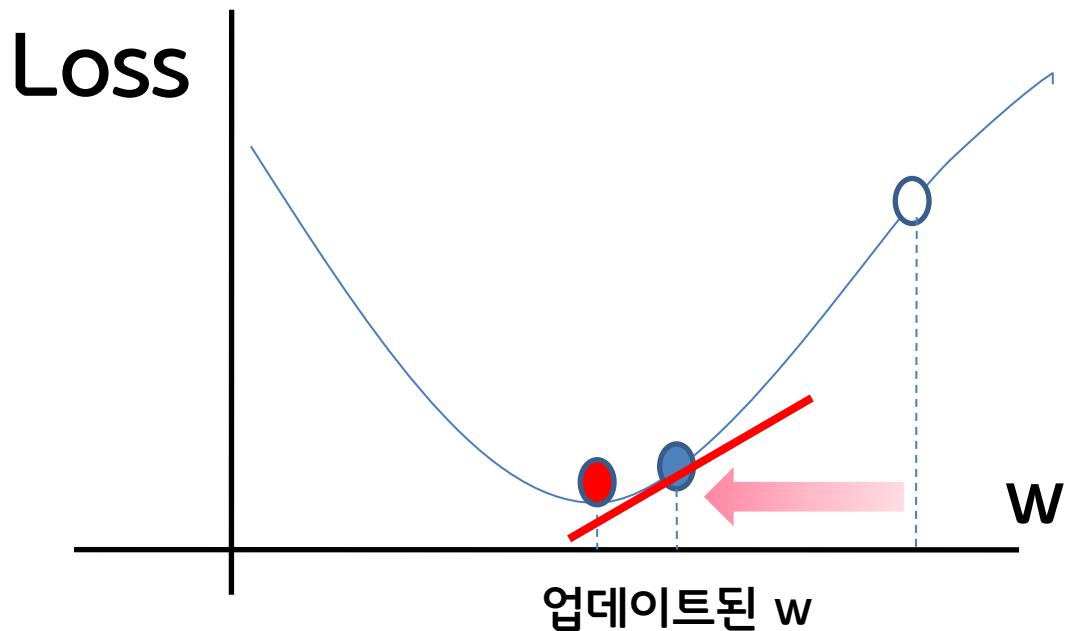
Gradient Descent



$$w = w - \eta \frac{\partial L}{\partial w}$$

η : learning rate

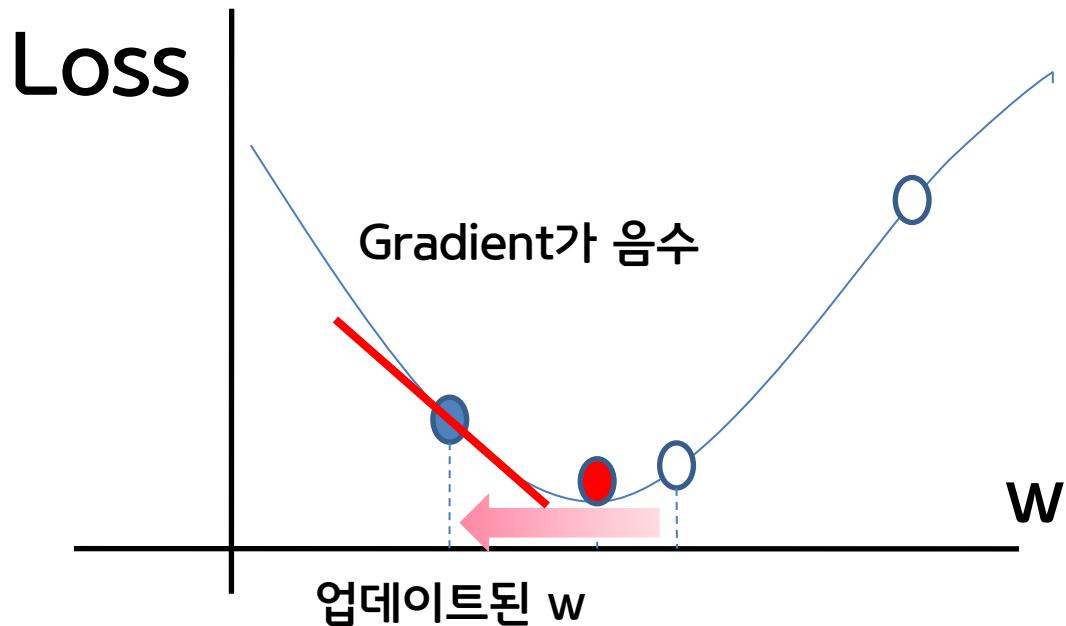
Gradient Descent



$$w = w - \eta \frac{\partial L}{\partial w}$$

η : learning rate

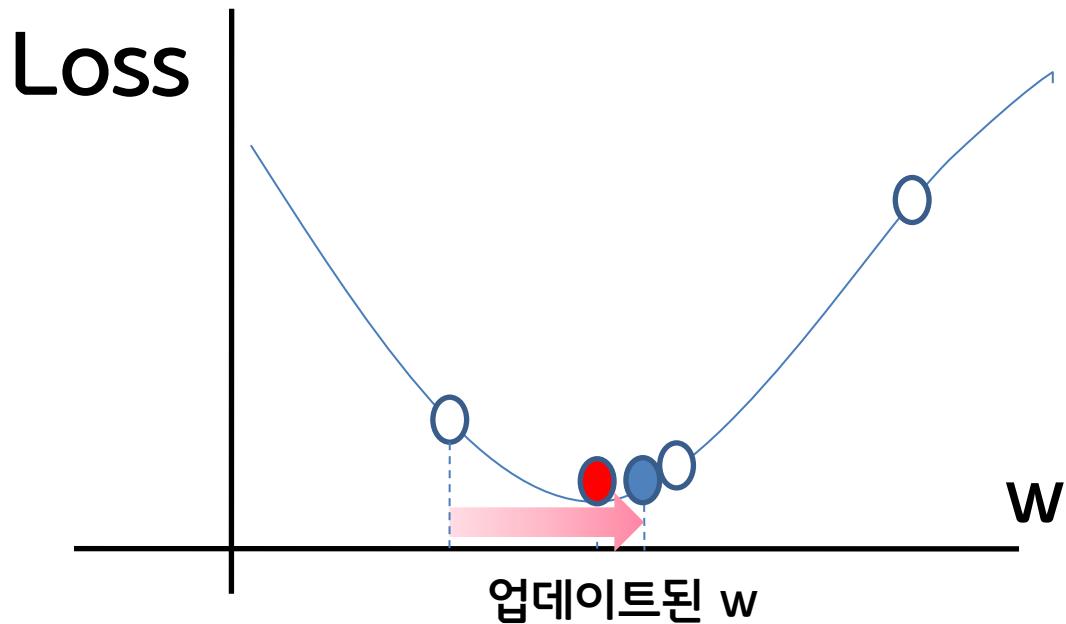
Gradient Descent



$$w = w - \eta \frac{\partial L}{\partial w}$$

η : learning rate

Gradient Descent



$$w = w - \eta \frac{\partial L}{\partial w}$$

Loss 함수에 대한 w 의 음의 Gradient 찾아서 연속적으로 업데이트해 주면 되는군요

그런데 어떻게 Gradient를 찾죠?

Gradient Descent

Strategy #2: Follow the slope

In 1-dimension, the derivative of a function:

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

In multiple dimensions, the **gradient** is the vector of (partial derivatives) along each dimension

The slope in any direction is the **dot product** of the direction with the gradient
The direction of steepest descent is the **negative gradient**

Gradient Descent

current W:

[0.34,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

loss 1.25347

gradient dW:

[?,
?,
?,
?,
?,
?,
?,
?,
?,
?,...]

Gradient Descent

current W:	$W + h$ (first dim):	gradient dW:
[0.34, -1.11, 0.78, 0.12, 0.55, 2.81, -3.1, -1.5, 0.33,...]	[0.34 + 0.0001 , -1.11, 0.78, 0.12, 0.55, 2.81, -3.1, -1.5, 0.33,...]	[?, ?, ?, ?, ?, ?, ?, ?, ?,...]

loss 1.25347 **loss 1.25322**

Gradient Descent

current W:

[0.34,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

loss 1.25347

W + h (first dim):

[0.34 + **0.0001**,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

loss 1.25322

gradient dW:

[-2.5,
?,
?,

$$\frac{(1.25322 - 1.25347)}{0.0001} = -2.5$$

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

?,
?,...]

Gradient Descent

current W:	$W + h$ (second dim):	gradient dW:
[0.34, -1.11, 0.78, 0.12, 0.55, 2.81, -3.1, -1.5, 0.33,...]	[0.34, -1.11 + 0.0001 , 0.78, 0.12, 0.55, 2.81, -3.1, -1.5, 0.33,...]	[-2.5, ?, ?, ?, ?, ?, ?, ?, ?,...]

loss 1.25347 loss 1.25353

Gradient Descent

current W:

[0.34,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

loss 1.25347

W + h (second dim):

[0.34,
-1.11 + 0.0001,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

loss 1.25353

gradient dW:

[-2.5,
0.6,
?,
?,
?]

$$(1.25353 - 1.25347)/0.0001
= 0.6$$

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

?,...]

Gradient Descent

current W:	W + h (third dim):	gradient dW:
[0.34, -1.11, 0.78, 0.12, 0.55, 2.81, -3.1, -1.5, 0.33,...]	[0.34, -1.11, 0.78 + 0.0001, 0.12, 0.55, 2.81, -3.1, -1.5, 0.33,...]	[-2.5, 0.6, ?, ?, ?, ?, ?, ?, ?, ?,...]

loss 1.25347 loss 1.25347

Gradient Descent

current W:

[0.34,
-1.11,
0.78,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

loss 1.25347

W + h (third dim):

[0.34,
-1.11,
0.78 + **0.0001**,
0.12,
0.55,
2.81,
-3.1,
-1.5,
0.33,...]

loss 1.25347

gradient dW:

[-2.5,
0.6,
0,
?,
?]

$$\frac{(1.25347 - 1.25347)}{0.0001} = 0$$

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

?,...]

Gradient Descent

This is silly. The loss is just a function of W:

$$L = \frac{1}{N} \sum_{i=1}^N L_i + \sum_k W_k^2$$

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$s = f(x; W) = Wx$$

want $\nabla_W L$

Gradient Descent

[Hammer image](#) is in the public domain

This is silly. The loss is just a function of W:

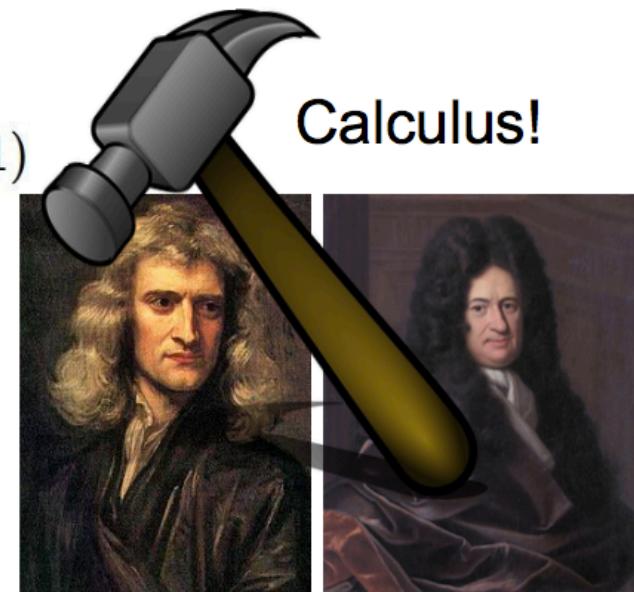
$$L = \frac{1}{N} \sum_{i=1}^N L_i + \sum_k W_k^2$$

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$s = f(x; W) = Wx$$

want $\nabla_W L$

Use calculus to compute an
analytic gradient



[This image](#) is in the public domain

[This image](#) is in the public domain

수치 미분은 구현은 쉽지만 시간이 오래걸립니다

또한 해석적 방법에 비해 부정확합니다



오차역전파법 (Backpropagation)

Computational Graph

- 연쇄법칙 (chain rule) 이란?
 - 합성함수의 미분은 합성 함수를 구성하는 각 함수의 미분의 곱으로 나타낼 수 있다

$$z = (x + y)^2 \rightarrow z = t^2$$

$$t = x + y$$

**z의 x 에 대한
미분의 연쇄법
칙 표현**

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t} \frac{\partial t}{\partial x} \rightarrow \frac{\partial z}{\partial x} = \cancel{\frac{\partial z}{\partial t}} \frac{\cancel{\partial t}}{\partial x}$$

Computational Graph

- 연쇄법칙이란?

$$z = (x + y)^2 \quad \Rightarrow \quad \begin{aligned} z &= t^2 \\ t &= x + y \end{aligned}$$

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t} \frac{\partial t}{\partial x} \quad \Rightarrow \quad \begin{aligned} \frac{\partial z}{\partial t} &= 2t \\ \frac{\partial t}{\partial x} &= 1 \end{aligned}$$

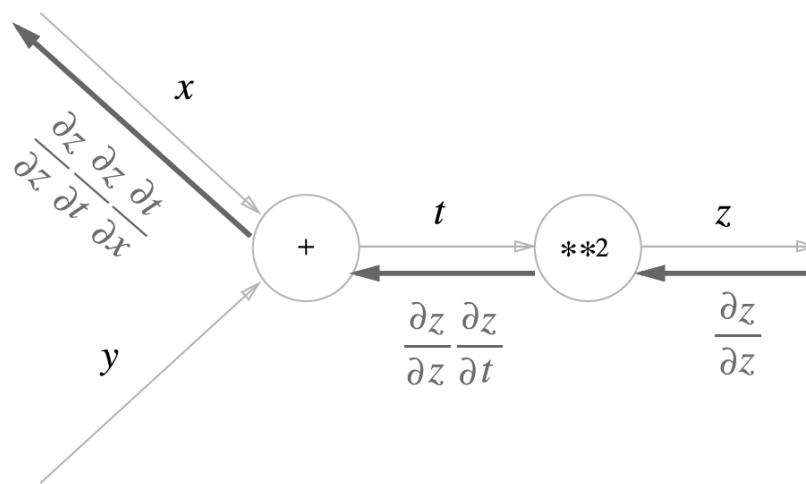
결과 $\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t} \frac{\partial t}{\partial x} = 2t \cdot 1 = 2(x + y)$

Computational Graph

- 연쇄법칙과 계산 그래프

역전파 과정

$$\begin{array}{c} z = (x + y)^2 \\ t = x + y \\ z = t^2 \end{array}$$

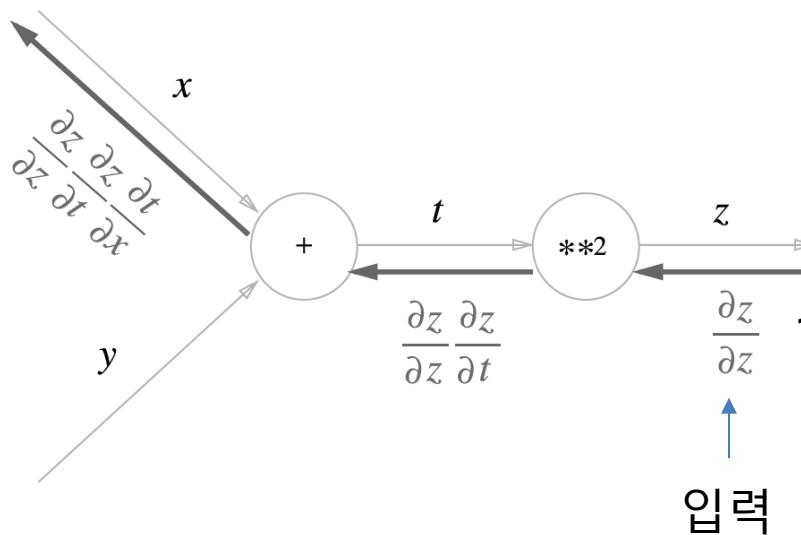


Computational Graph

- 연쇄법칙과 계산 그래프

역전파 과정

$$\begin{array}{ccc} z = (x + y)^2 & \Rightarrow & z = t^2 \\ & & t = x + y \end{array}$$



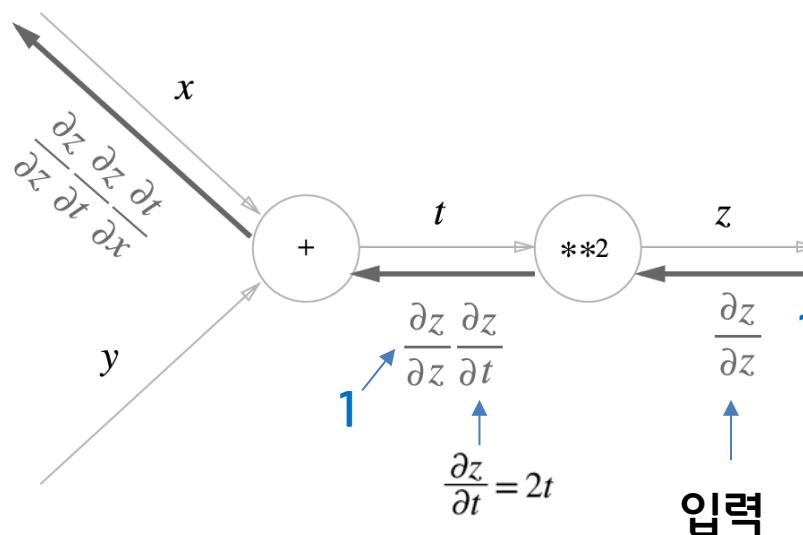
Computational Graph

- 연쇄법칙과 계산 그래프

역전파 과정

$$z = (x + y)^2 \quad \Rightarrow \quad z = t^2$$

$$t = x + y$$



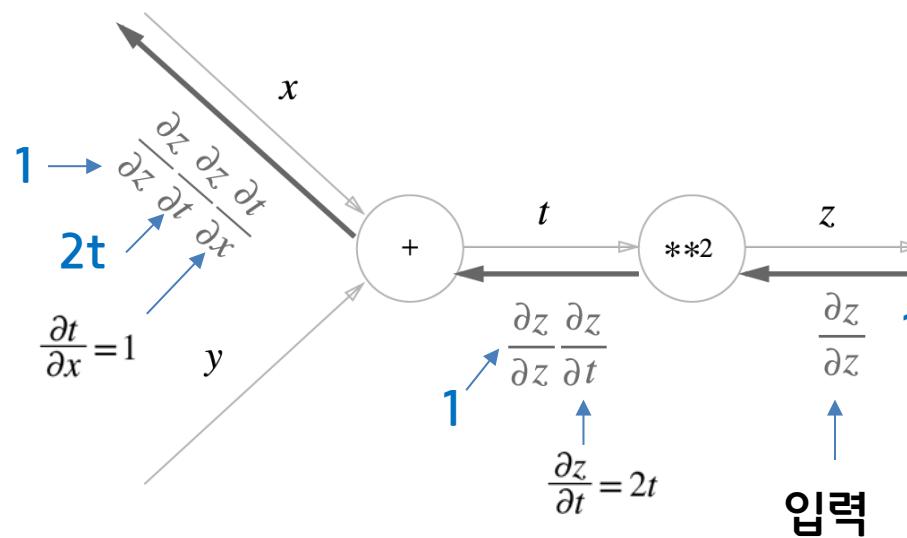
Computational Graph

- 연쇄법칙과 계산 그래프

역전파 과정

$$z = (x + y)^2 \quad \Rightarrow \quad z = t^2$$

$$t = x + y$$

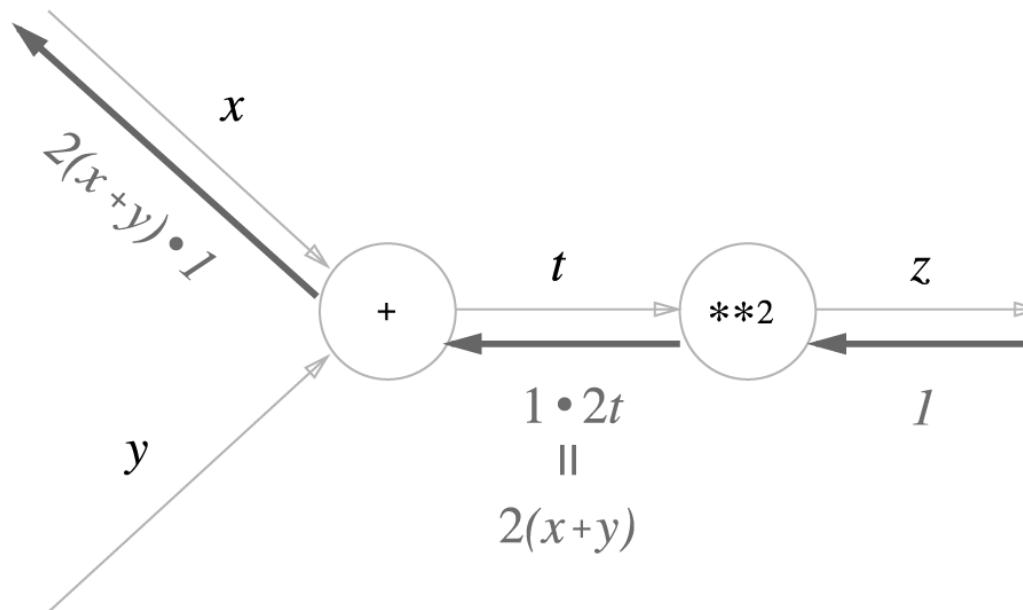


Computational Graph

- 연쇄법칙과 계산 그래프

역전파 과정

$$\begin{array}{ccc} z = (x + y)^2 & \xrightarrow{\hspace{1cm}} & z = t^2 \\ & & t = x + y \end{array}$$



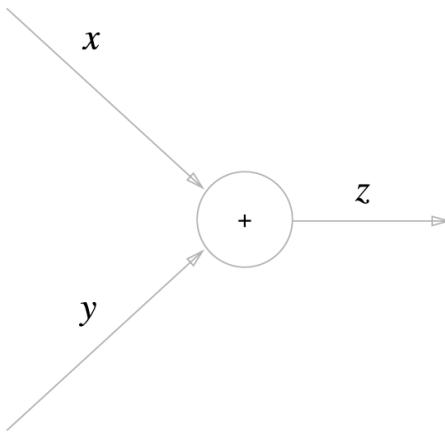
Computational Graph

- 덧셈 노드의 역전파

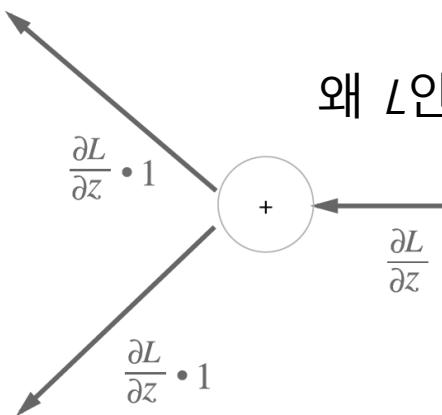
$$z = x + y$$

$$\frac{\partial z}{\partial x} = 1$$

$$\frac{\partial z}{\partial y} = 1$$



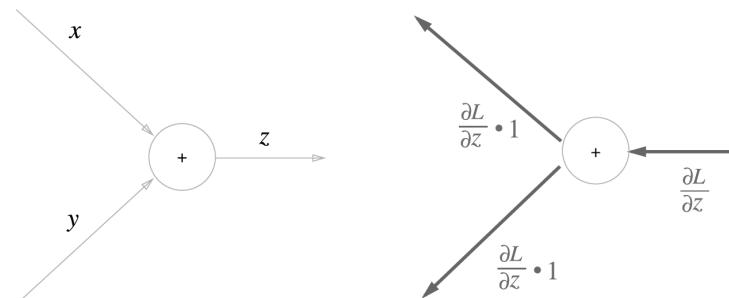
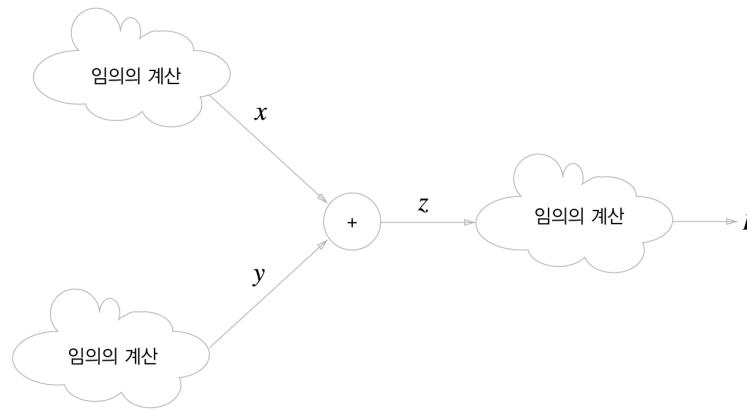
왜 L 인가요?



Computational Graph

- 덧셈 노드의 역전파

실제로 큰 계산을
가정하고 그중 한
노드가 덧셈노드임
을 가정한 겁니다



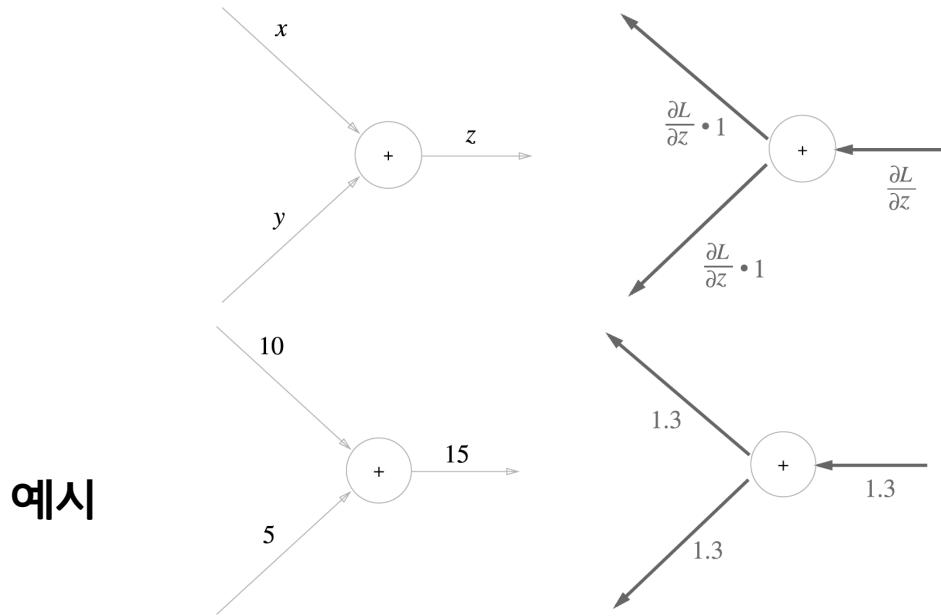
Computational Graph

- 덧셈 노드의 역전파

$$z = x + y \rightarrow$$

$$\frac{\partial z}{\partial x} = 1$$

$$\frac{\partial z}{\partial y} = 1$$

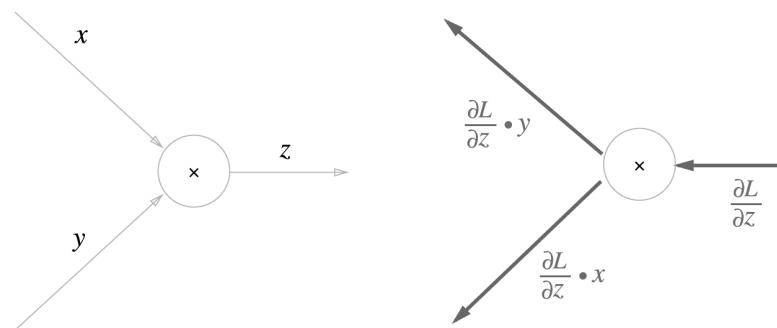


예시

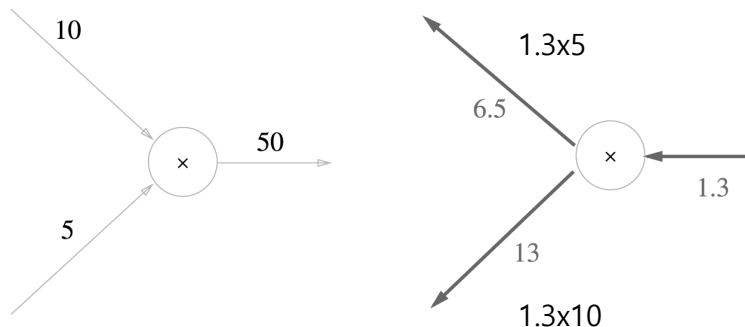
Computational Graph

- 곱셈 노드의 역전파

$$z = xy \quad \rightarrow \quad \begin{aligned} \frac{\partial z}{\partial x} &= y \\ \frac{\partial z}{\partial y} &= x \end{aligned}$$



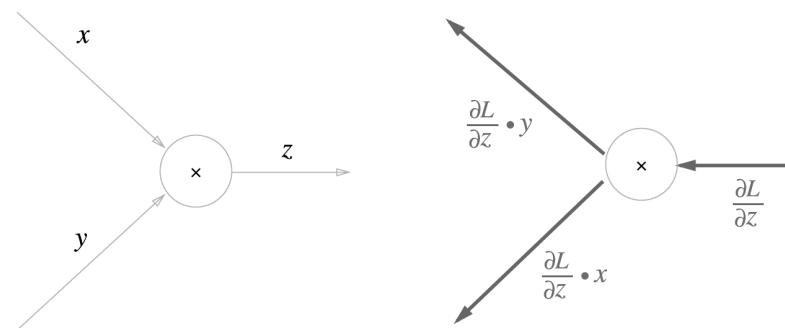
예시



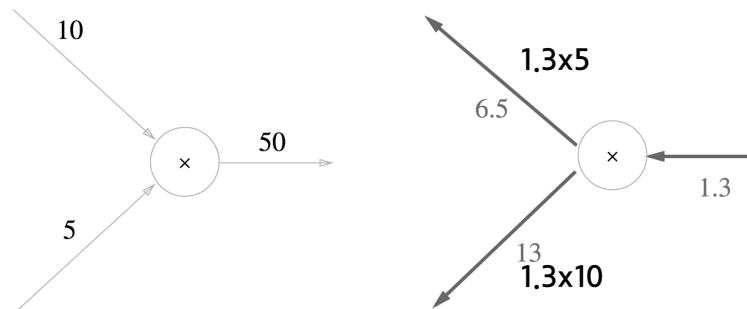
Computational Graph

- 곱셈 노드의 역전파

$$z = xy \rightarrow \begin{aligned} \frac{\partial z}{\partial x} &= y \\ \frac{\partial z}{\partial y} &= x \end{aligned}$$

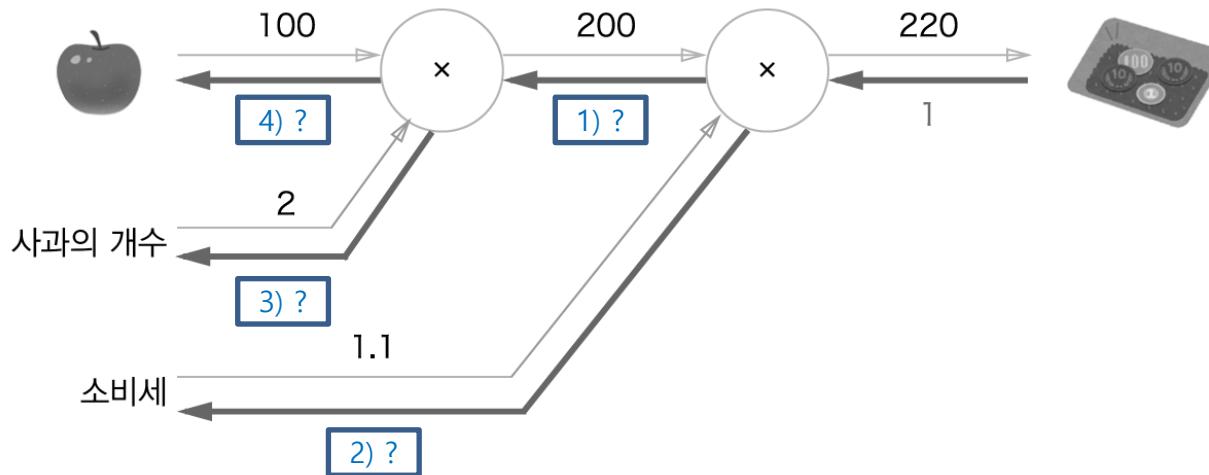


- 곱셈의 역전파는 순방향 입력신호의 값이 필요합니다
- 곱셈노드 구현 시 순전파 입력신호를 변수에 저장해 둡니다



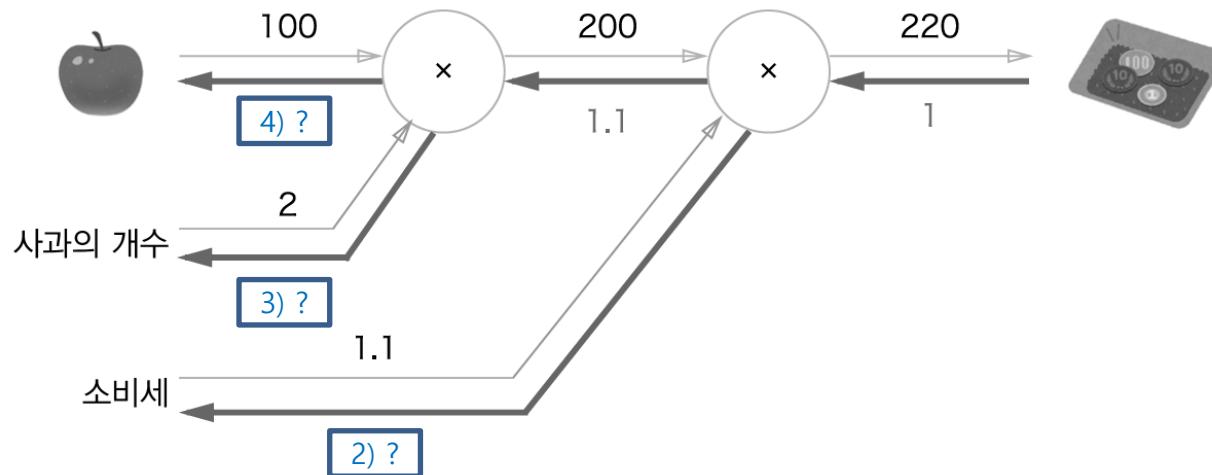
Computational Graph

- 사과쇼핑의 예



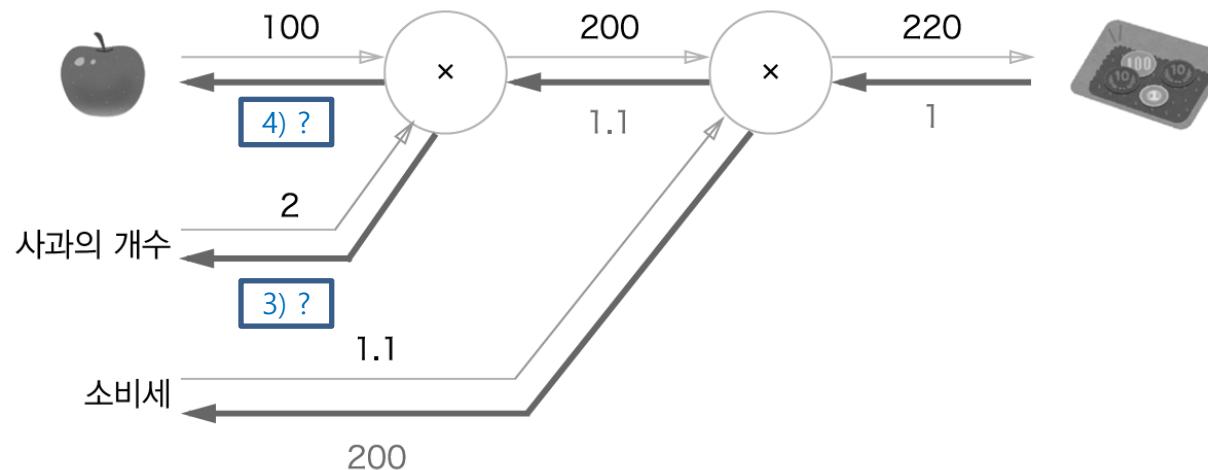
Computational Graph

- 사과쇼핑의 예



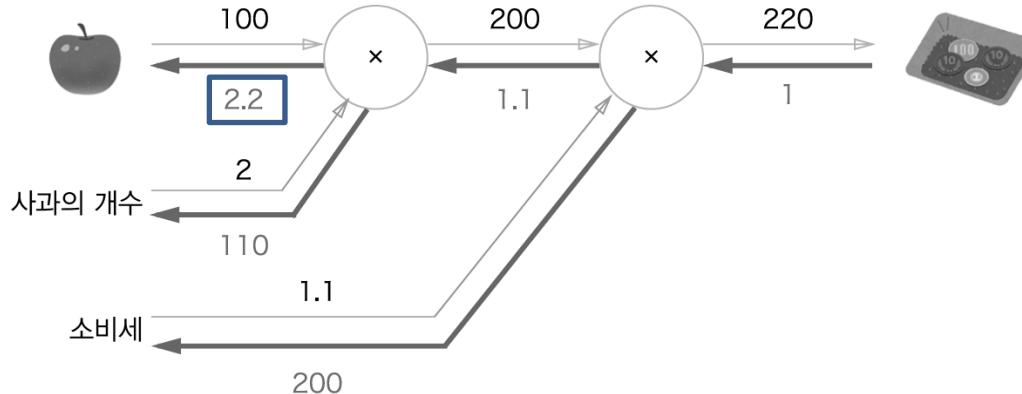
Computational Graph

- 사과쇼핑의 예



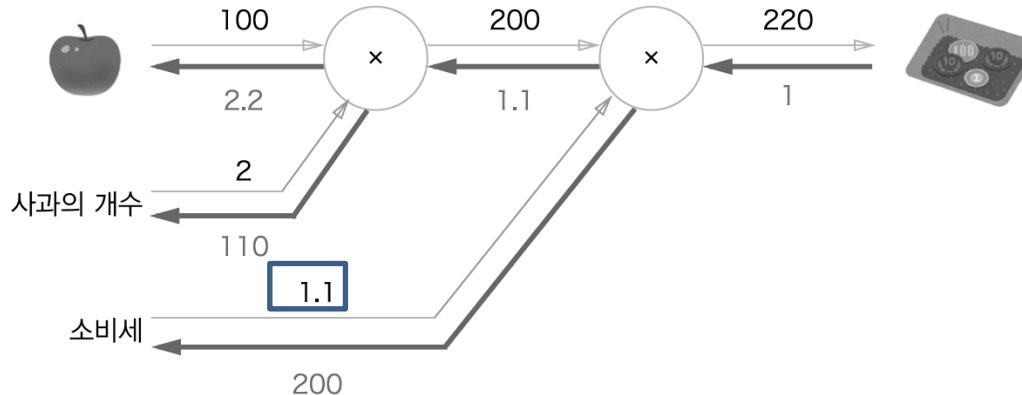
Computational Graph

- 사과쇼핑의 예



- 사과가격이 1원 오르면, 최종 가격 2.2 증가
 - $101(\text{사과 가격}) \times 2 \times 1.1 = 222.2 \text{ 원}$

Computational Graph



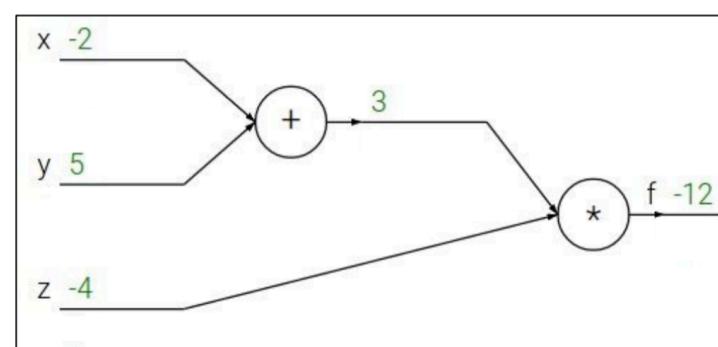
- 사과가격이 1원 오르면, 최종 가격 2.2 증가
 - $101(\text{사과 가격}) \times 2 \times 1.1 = 222.2$ 원
- 소비세가 100%($1.1+1 = 2.1$)오르면 가격 200증가
 - $100 \times 2 \times 2.1(\text{소비세}) = 420$ 원

Computational Graph

Backpropagation: a simple example

$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$



Computational Graph

Backpropagation: a simple example

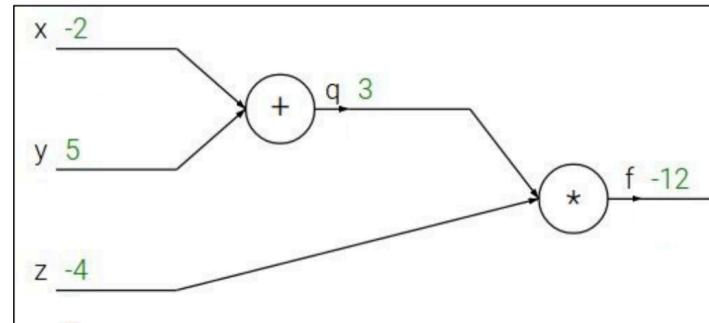
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Computational Graph

Backpropagation: a simple example

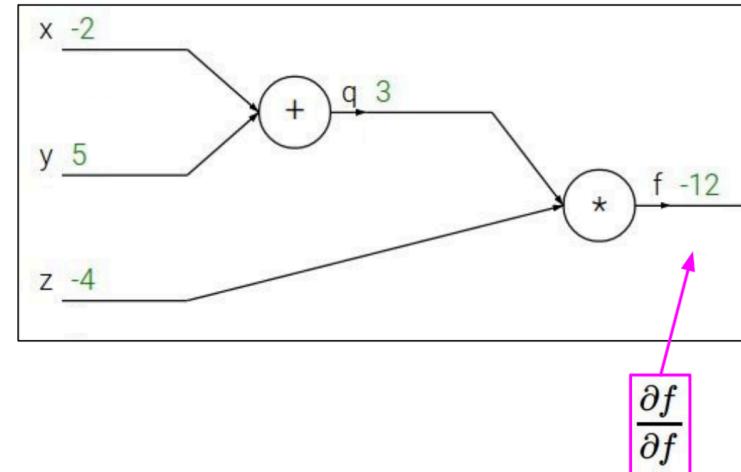
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Computational Graph

Backpropagation: a simple example

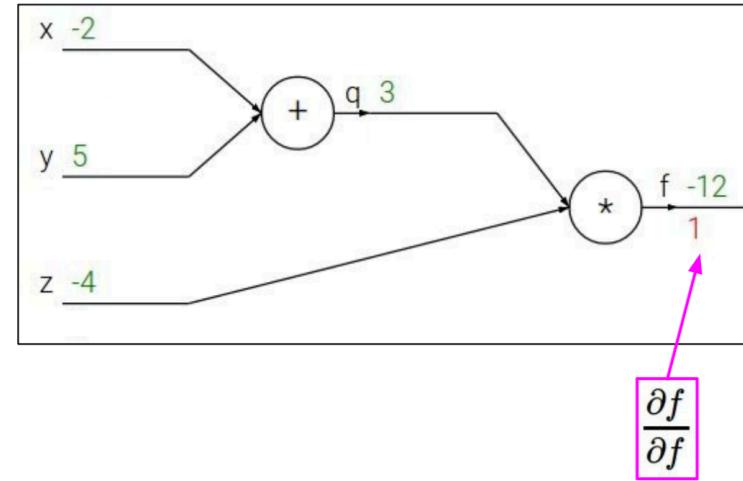
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2, y = 5, z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Computational Graph

Backpropagation: a simple example

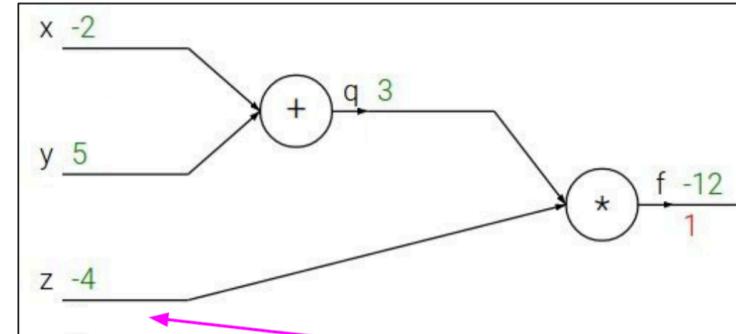
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial z}$$

Computational Graph

Backpropagation: a simple example

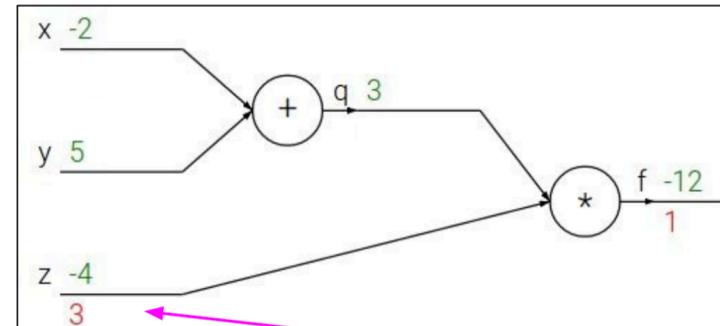
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial z}$$

Computational Graph

Backpropagation: a simple example

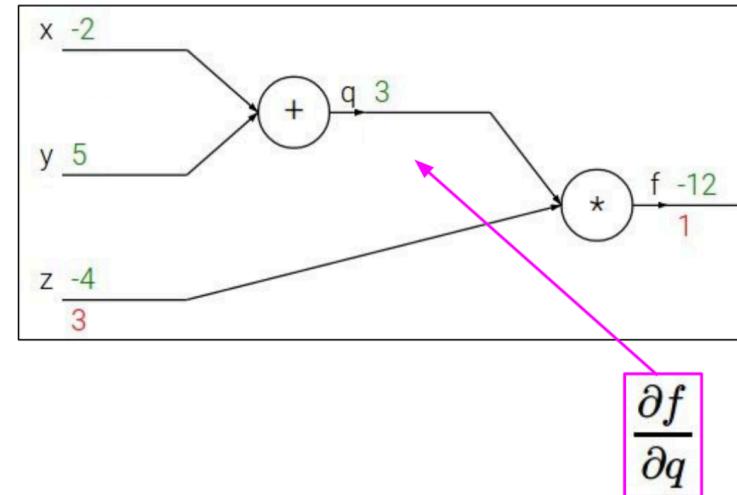
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Computational Graph

Backpropagation: a simple example

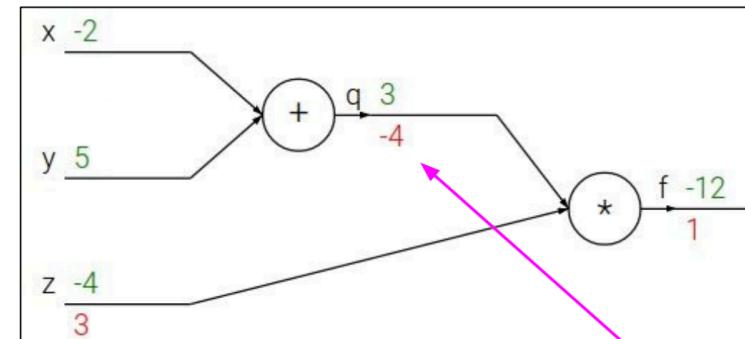
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial q}$$

Computational Graph

Backpropagation: a simple example

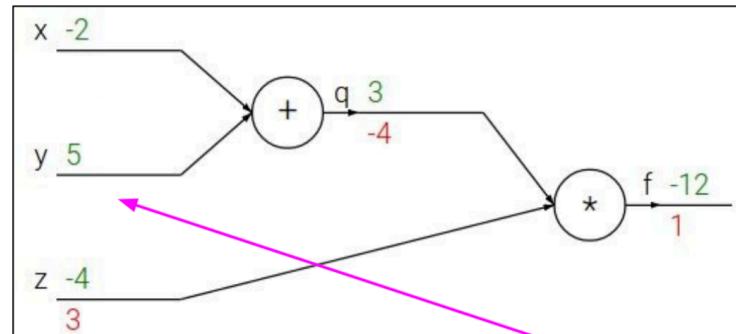
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial y}$$

Computational Graph

Backpropagation: a simple example

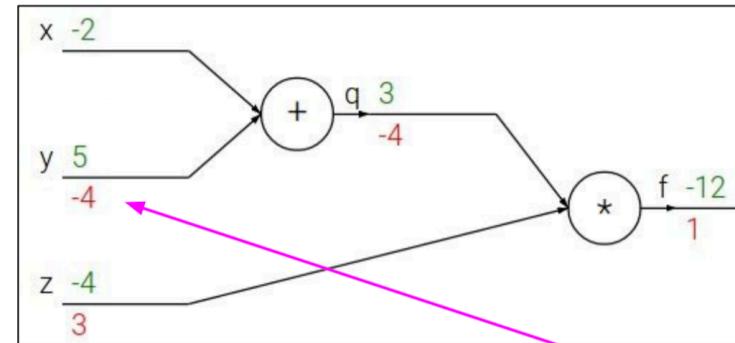
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Chain rule:

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial y}$$

$$\frac{\partial f}{\partial y}$$

Computational Graph

Backpropagation: a simple example

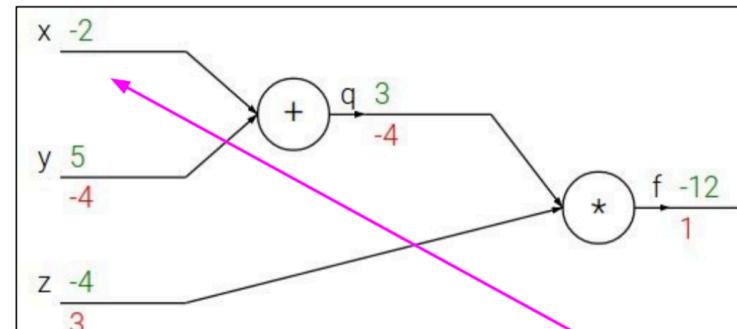
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



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

Computational Graph

Backpropagation: a simple example

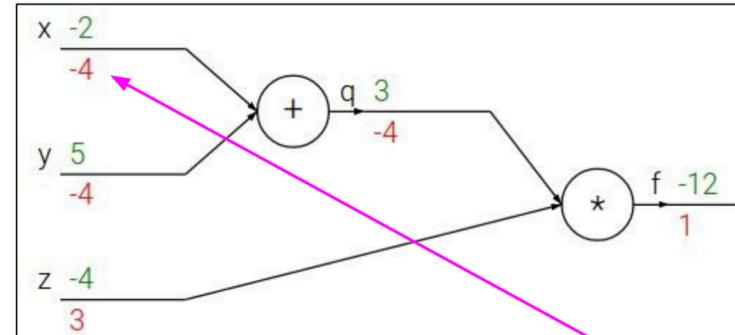
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

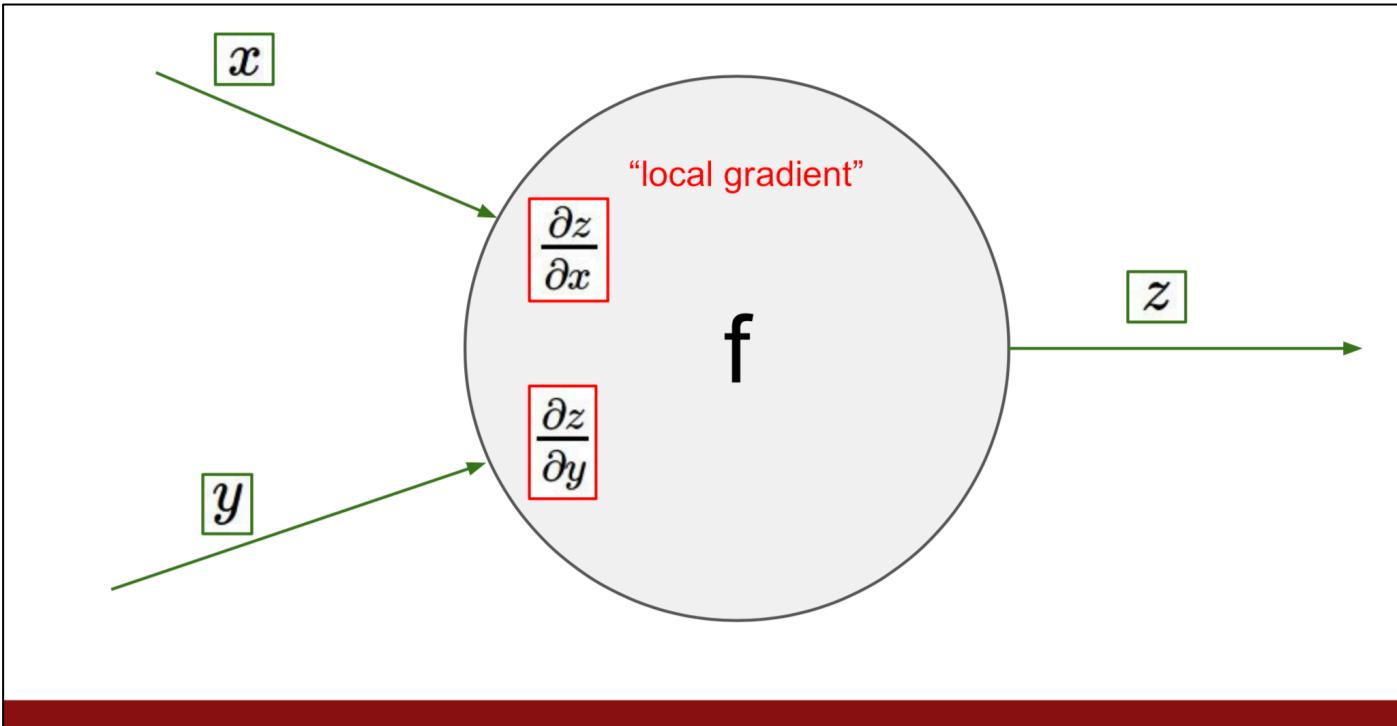


Chain rule:

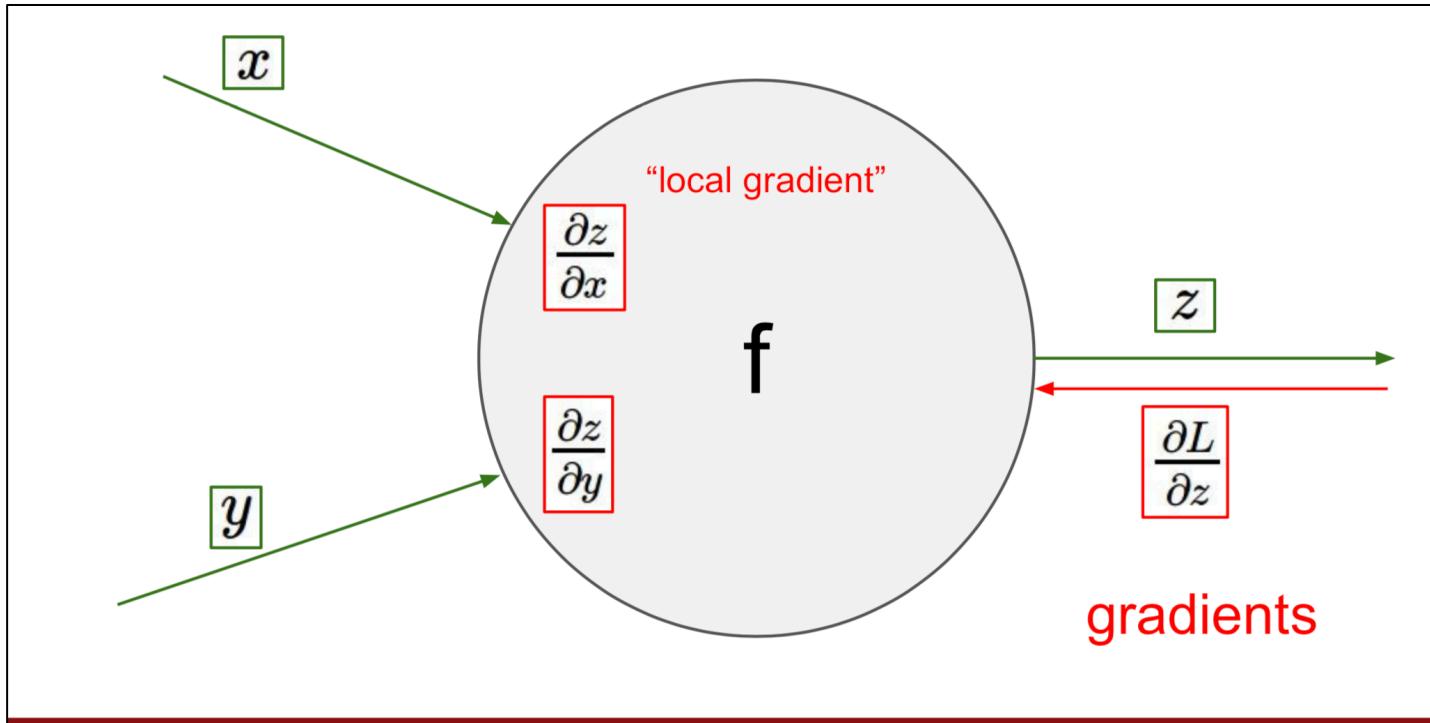
$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}$$

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

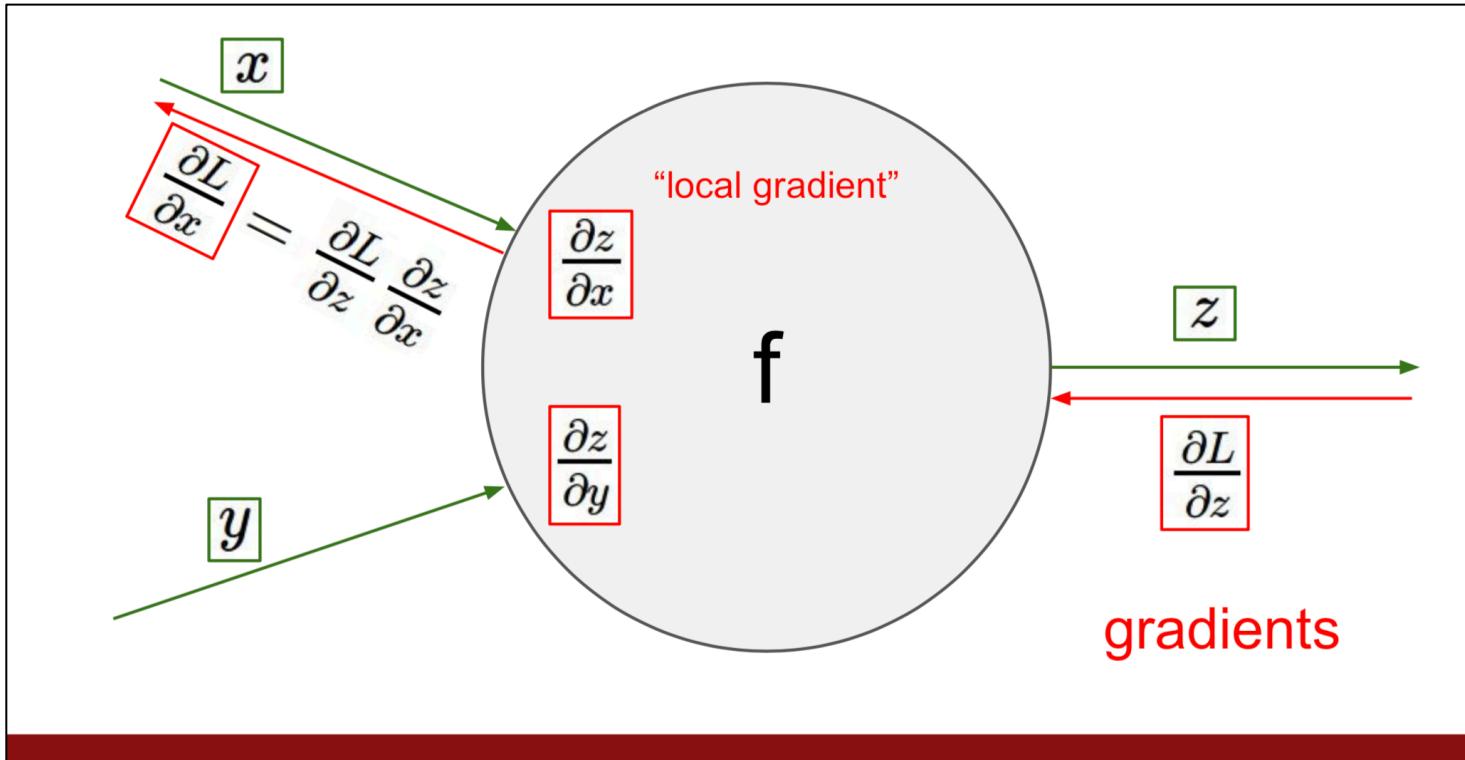
Computational Graph



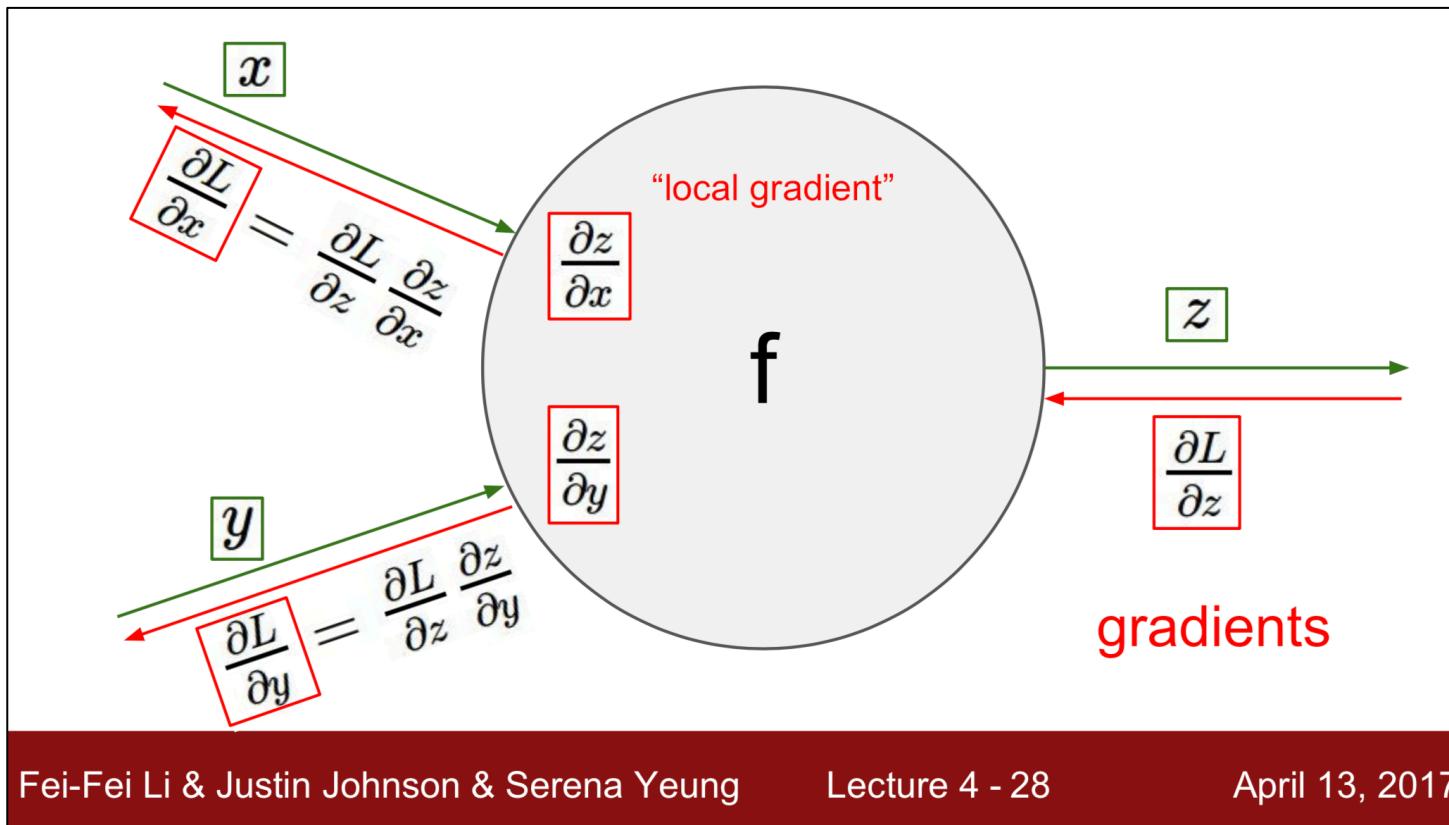
Computational Graph



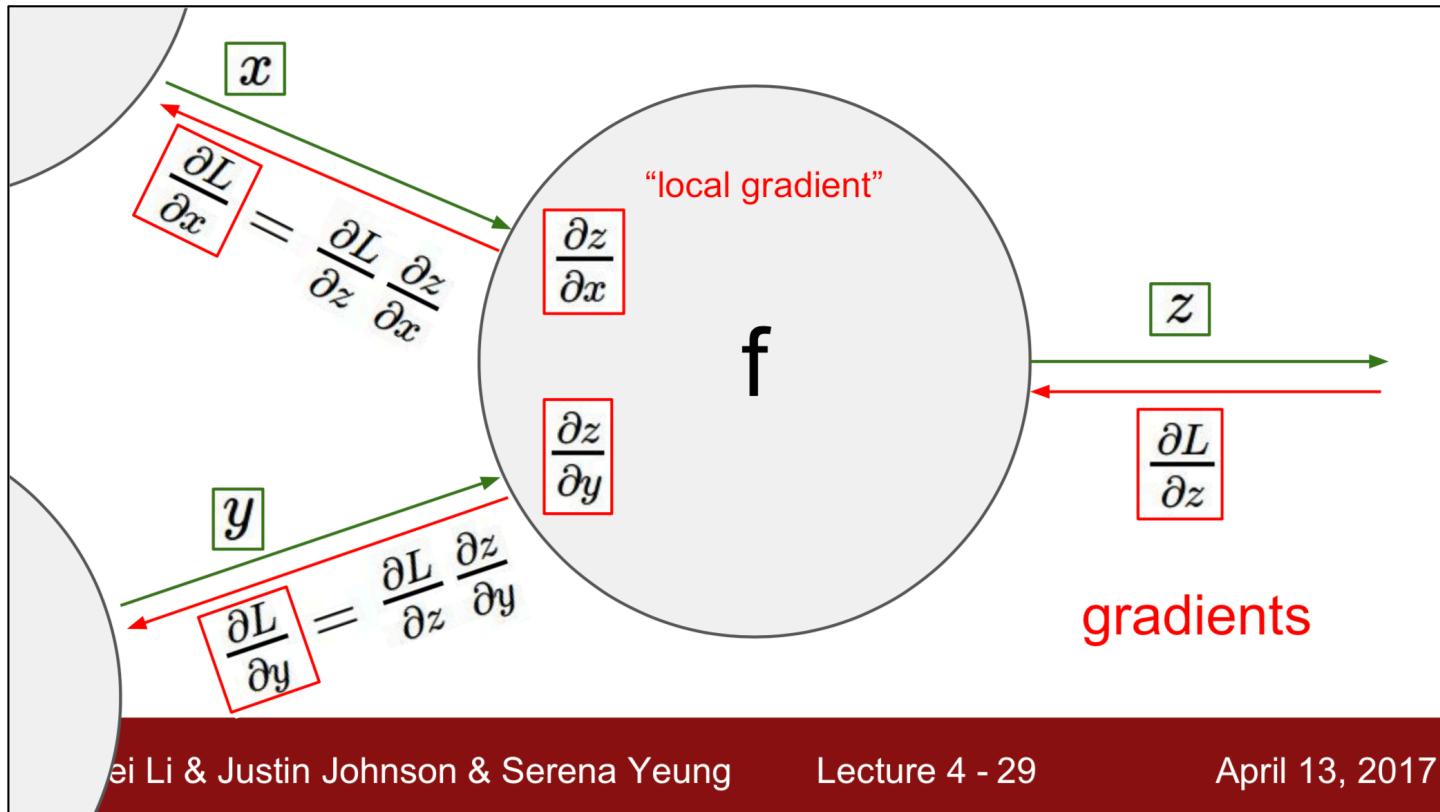
Computational Graph



Computational Graph

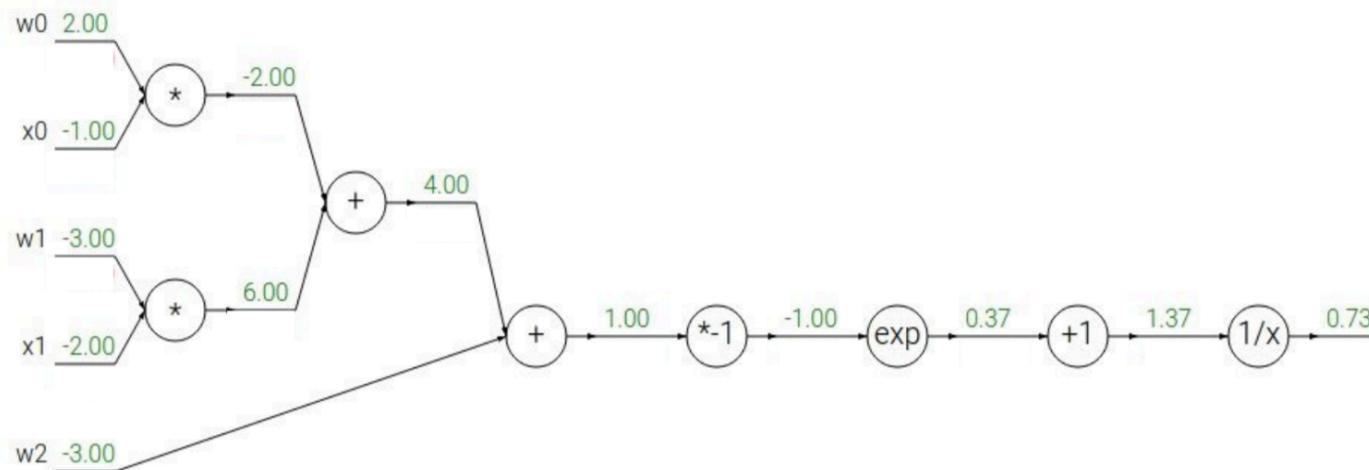


Computational Graph



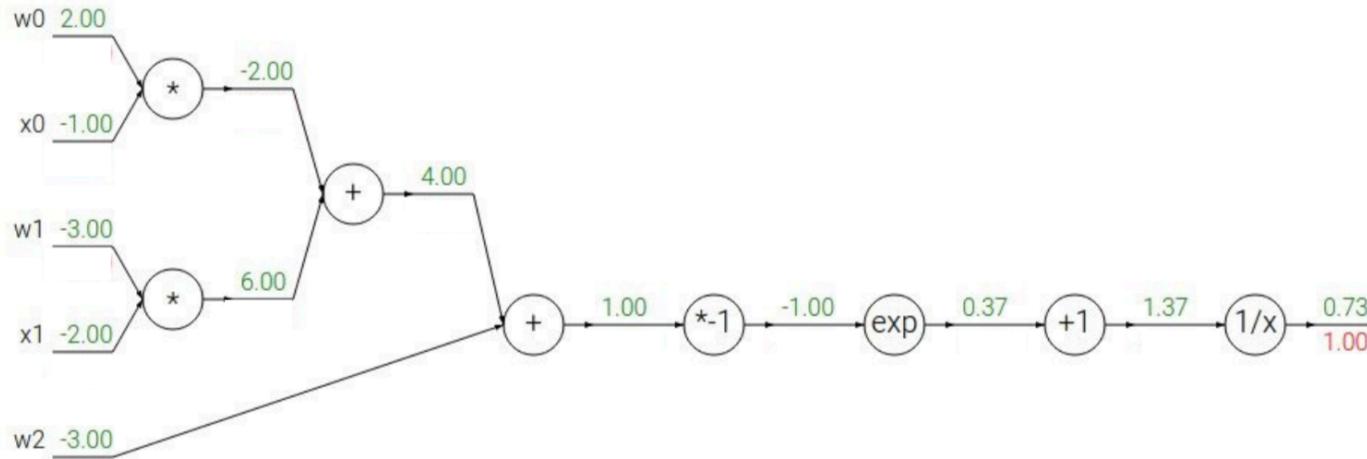
Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

→

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

$$f_a(x) = ax$$

→

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

$$f_c(x) = c + x$$

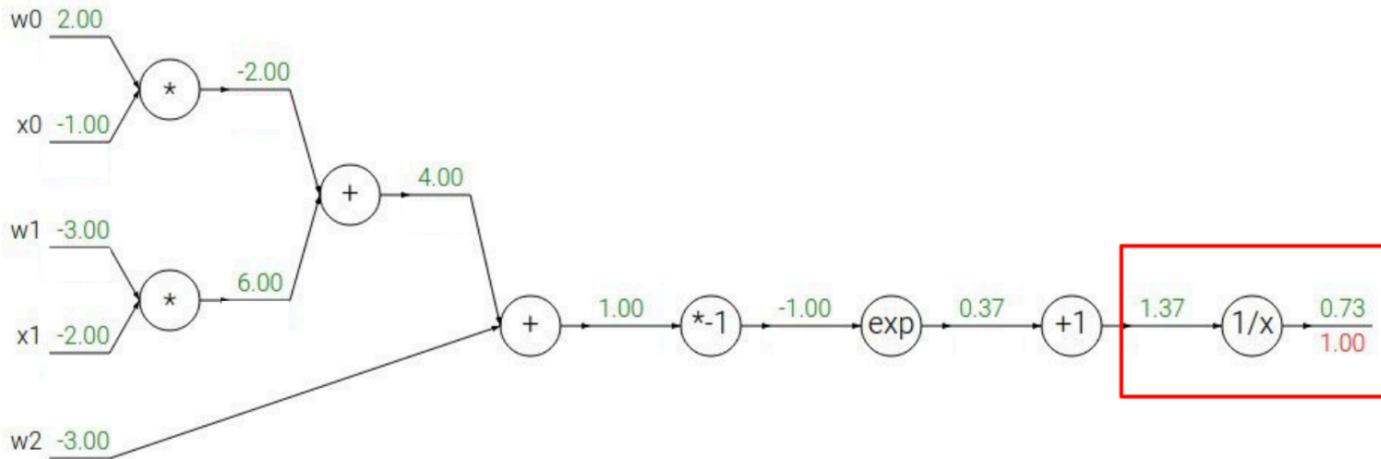
→

$$\frac{df}{dx} = -1/x^2$$

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

\rightarrow

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

$$f_a(x) = ax$$

\rightarrow

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

$$f_c(x) = c + x$$

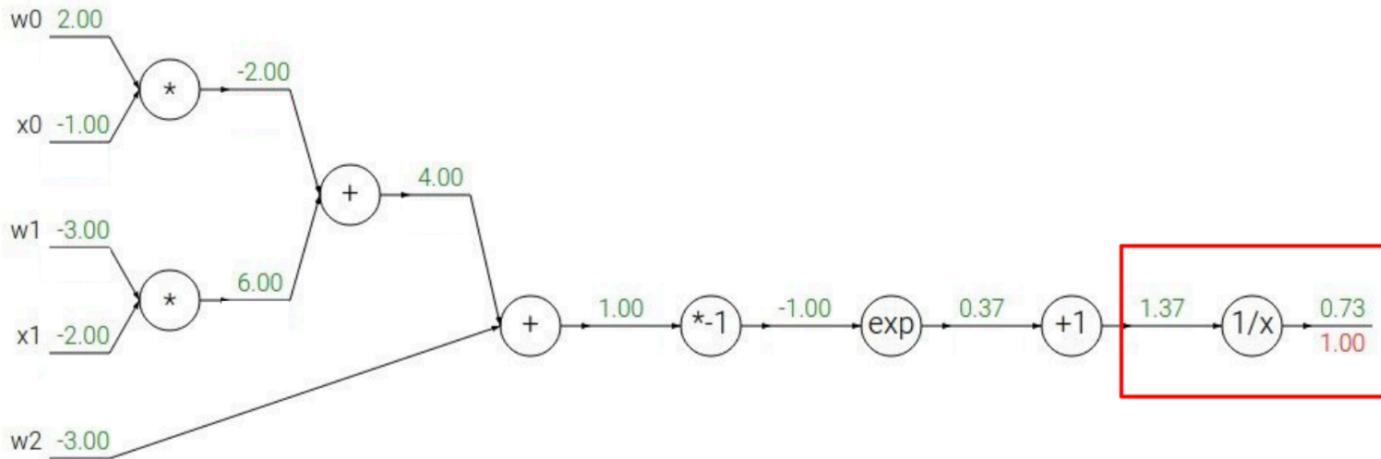
\rightarrow

$$\frac{df}{dx} = -1/x^2$$

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

\rightarrow

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

$$f_a(x) = ax$$

\rightarrow

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

$$f_c(x) = c + x$$

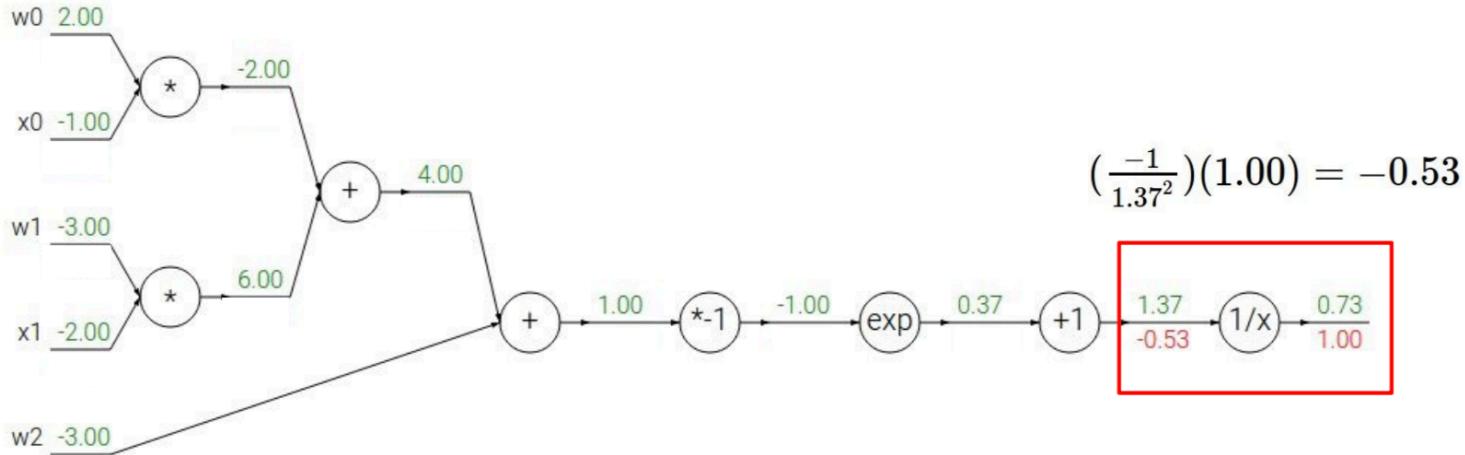
\rightarrow

$$\frac{df}{dx} = -1/x^2$$

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

\rightarrow

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

$$f_a(x) = ax$$

\rightarrow

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

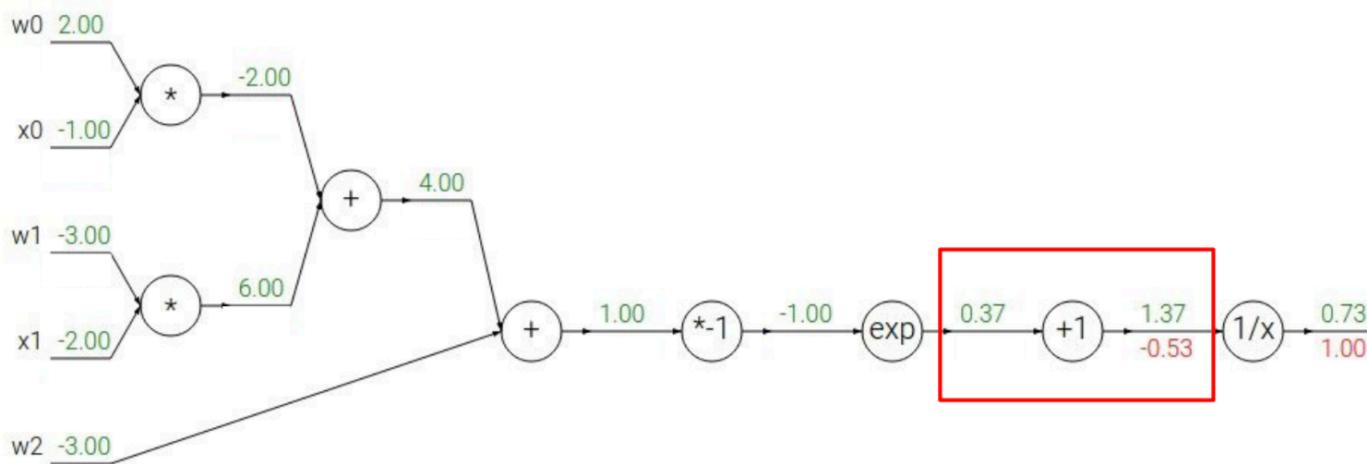
$$f_c(x) = c + x$$

$$\frac{df}{dx} = -1/x^2$$

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

\rightarrow

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

$$f_a(x) = ax$$

\rightarrow

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

\rightarrow

$$\frac{df}{dx} = -1/x^2$$

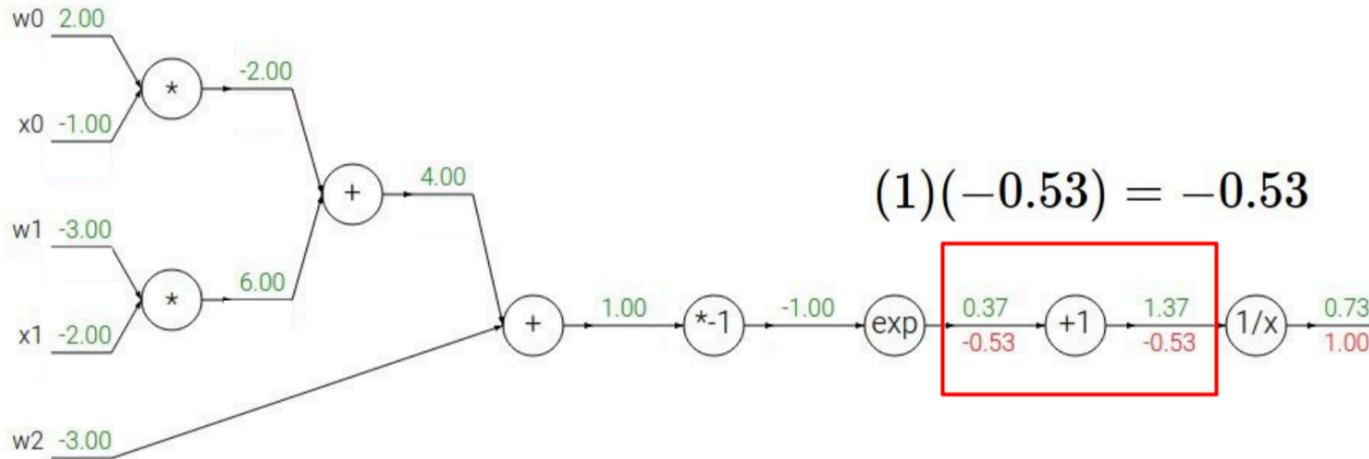
$$f_c(x) = c + x$$

\rightarrow

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

\rightarrow

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

$$f_a(x) = ax$$

\rightarrow

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

\rightarrow

$$\frac{df}{dx} = -1/x^2$$

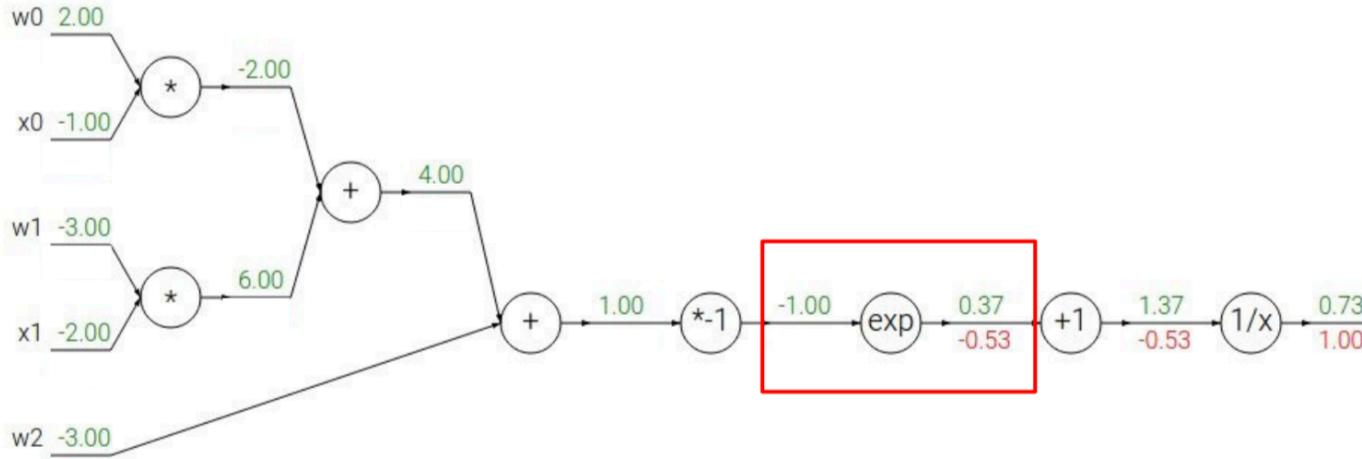
$$f_c(x) = c + x$$

\rightarrow

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

\rightarrow

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

$$f_a(x) = ax$$

\rightarrow

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

\rightarrow

$$\frac{df}{dx} = -1/x^2$$

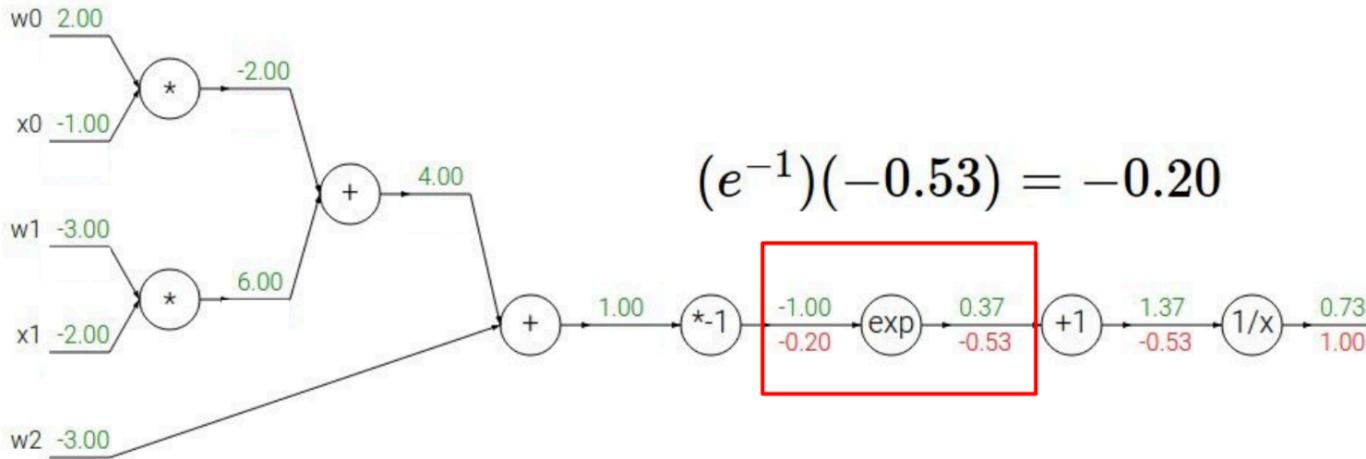
$$f_c(x) = c + x$$

\rightarrow

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

\rightarrow

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

$$f_a(x) = ax$$

\rightarrow

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

$$f_c(x) = c + x$$

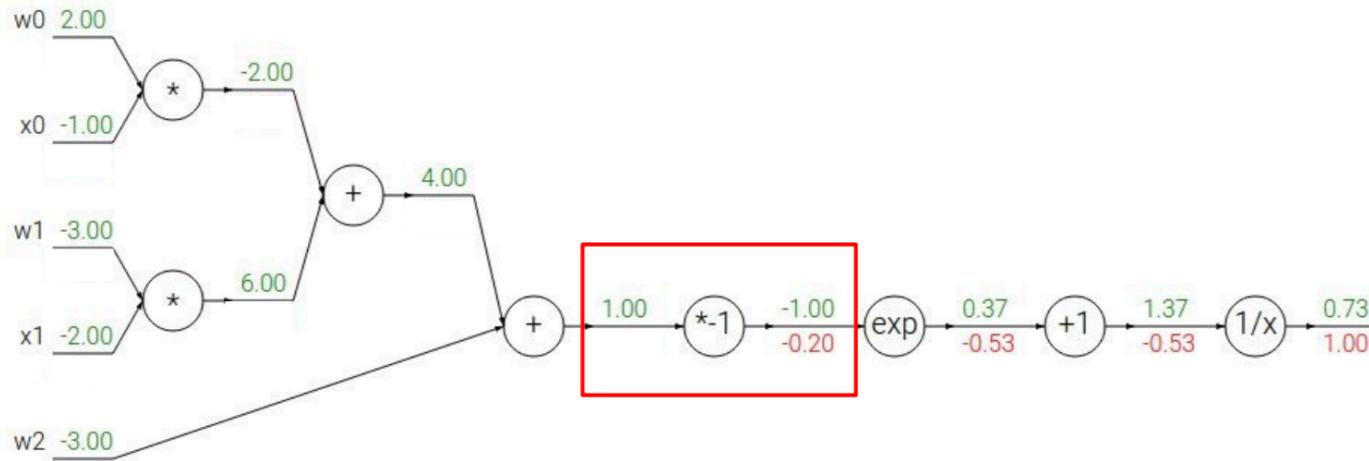
\rightarrow

$$\frac{df}{dx} = -1/x^2$$

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

\rightarrow

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

$$f_a(x) = ax$$

\rightarrow

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

\rightarrow

$$\frac{df}{dx} = -1/x^2$$

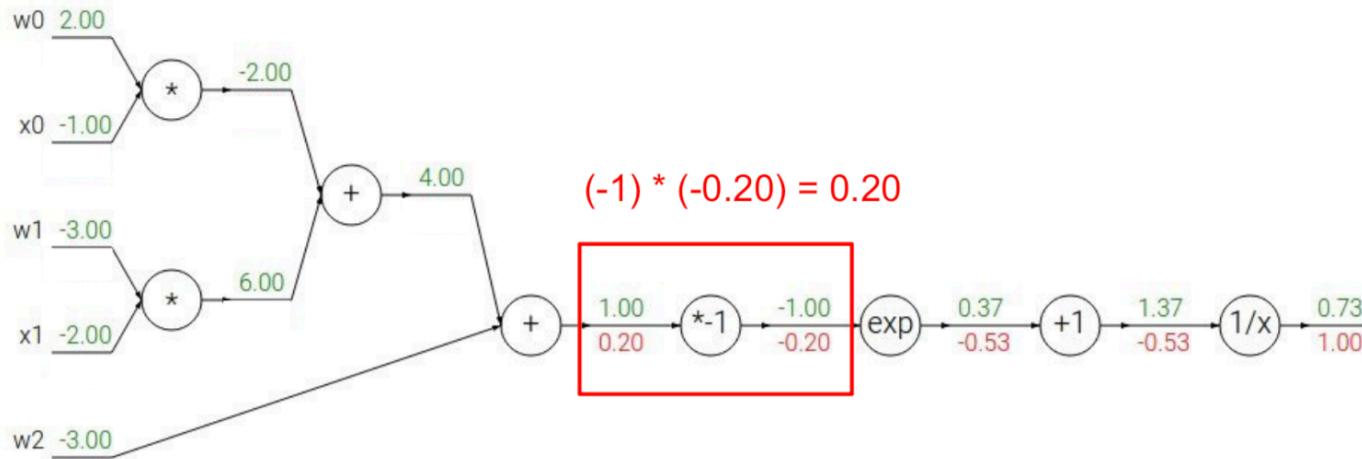
$$f_c(x) = c + x$$

\rightarrow

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

\rightarrow

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

$$f_a(x) = ax$$

\rightarrow

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

\rightarrow

$$\frac{df}{dx} = -1/x^2$$

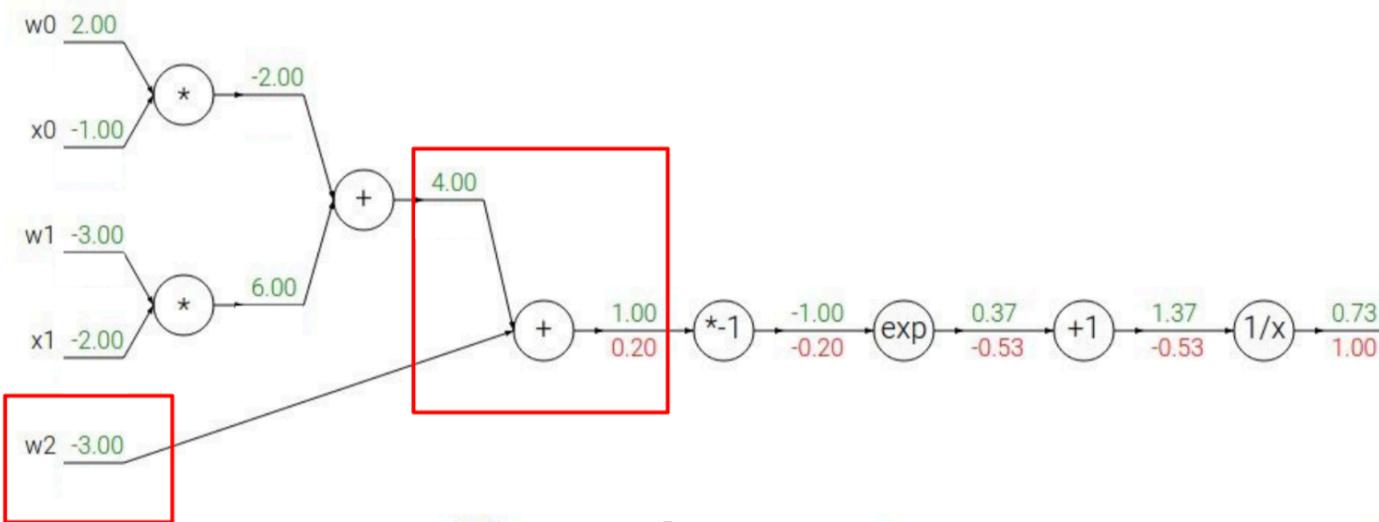
$$f_c(x) = c + x$$

\rightarrow

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

→

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

$$f_a(x) = ax$$

→

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

→

$$\frac{df}{dx} = -1/x^2$$

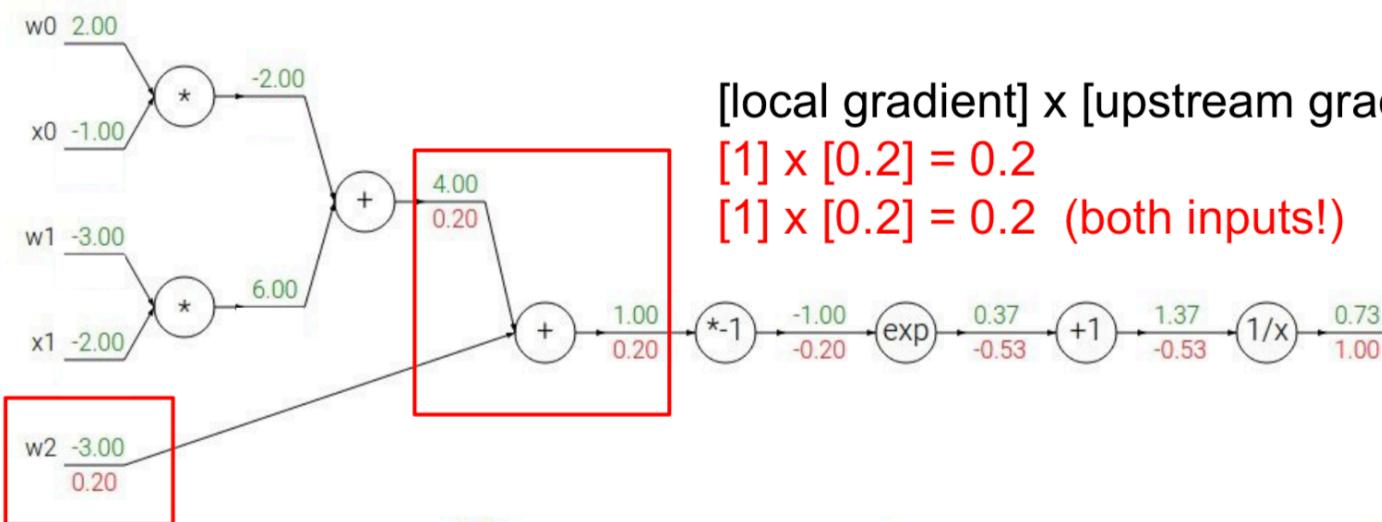
$$f_c(x) = c + x$$

→

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

\rightarrow

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

$$f_a(x) = ax$$

\rightarrow

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

$$f_c(x) = c + x$$

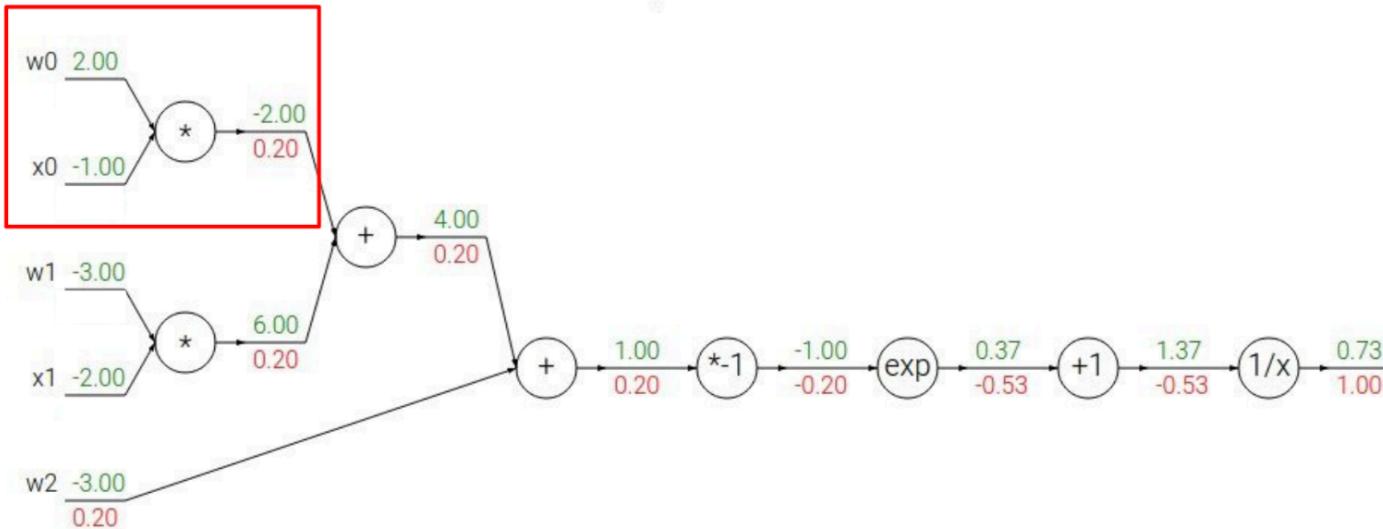
\rightarrow

$$\frac{df}{dx} = -1/x^2$$

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

\rightarrow

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

$$f_a(x) = ax$$

\rightarrow

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

\rightarrow

$$\frac{df}{dx} = -1/x^2$$

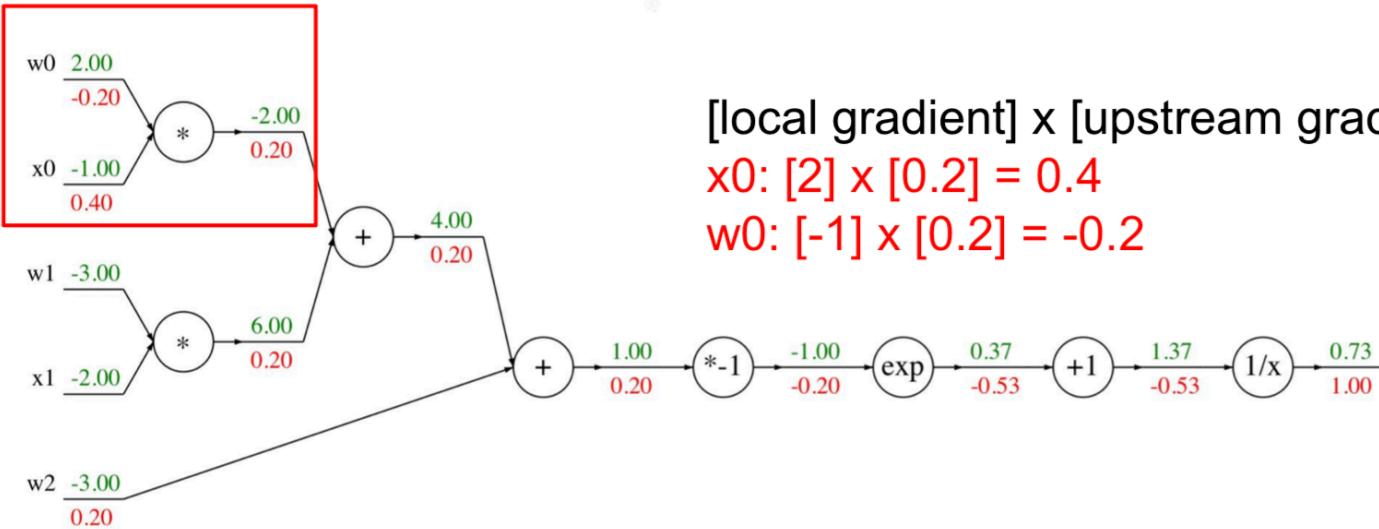
$$f_c(x) = c + x$$

\rightarrow

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



[local gradient] x [upstream gradient]
 $x_0: [2] \times [0.2] = 0.4$
 $w_0: [-1] \times [0.2] = -0.2$

$$f(x) = e^x$$

→

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

$$f_a(x) = ax$$

→

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

→

$$\frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x$$

→

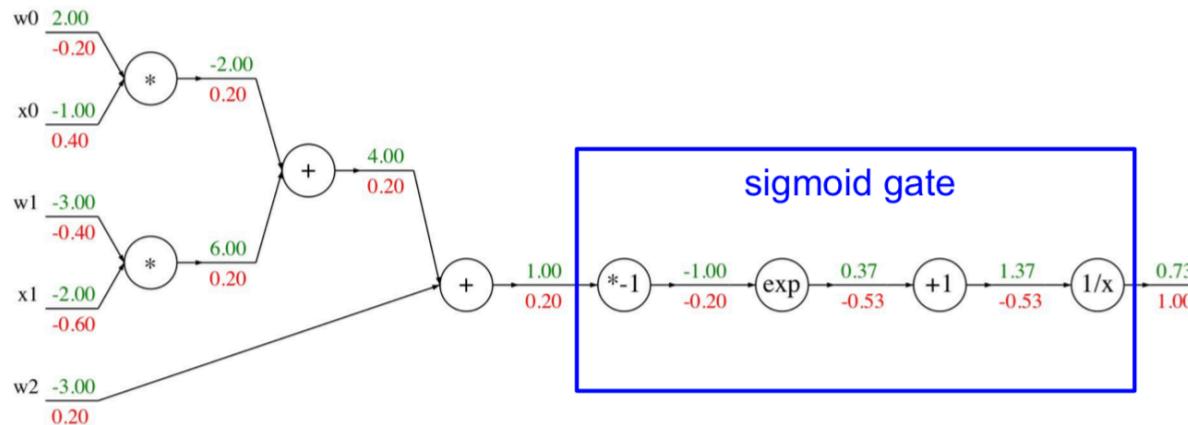
$$\frac{df}{dx} = 1$$

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$

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

sigmoid function

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1 + e^{-x})^2} = \left(\frac{1 + e^{-x} - 1}{1 + e^{-x}} \right) \left(\frac{1}{1 + e^{-x}} \right) = (1 - \sigma(x)) \sigma(x)$$

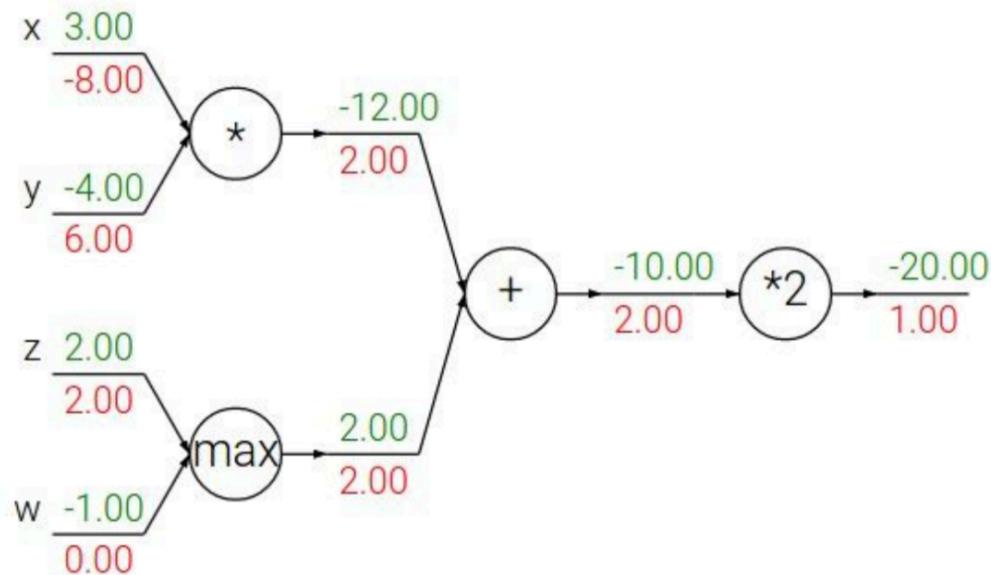


Patterns in backward flow

add gate: gradient distributor

max gate: gradient router

mul gate: gradient switcher

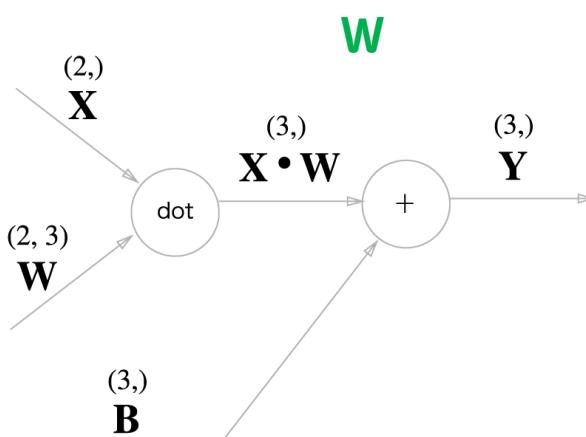


Computational Graph

강아지 파라미터

0.2	0.1	...	0.3	0.7
0.7	0.8	...	0.9	0.4

고양이 파라미터



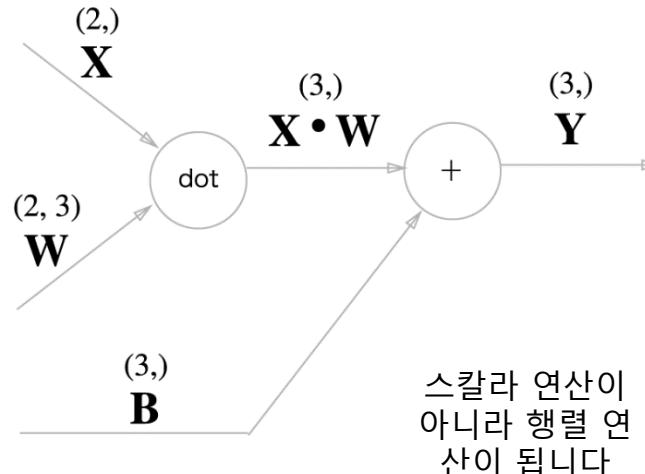
$$\begin{matrix} 155 \\ 200 \\ \dots \\ 110 \\ 78 \end{matrix} \textbf{x} + \begin{matrix} 0.1 \\ 0.2 \end{matrix} \textbf{b}$$

- Affine 계층

Affine / Softmax 계층 구현 하기

- Affine 계층

```
In [18]: X = np.random.rand(2)
In [19]: W = np.random.rand(2,3)
In [20]: B = np.random.rand(3)
In [21]: X.shape
Out[21]: (2,)
In [22]: W.shape
Out[22]: (2, 3)
In [23]: B.shape
Out[23]: (3,)
In [24]: Y = np.dot(X, W) + B
```

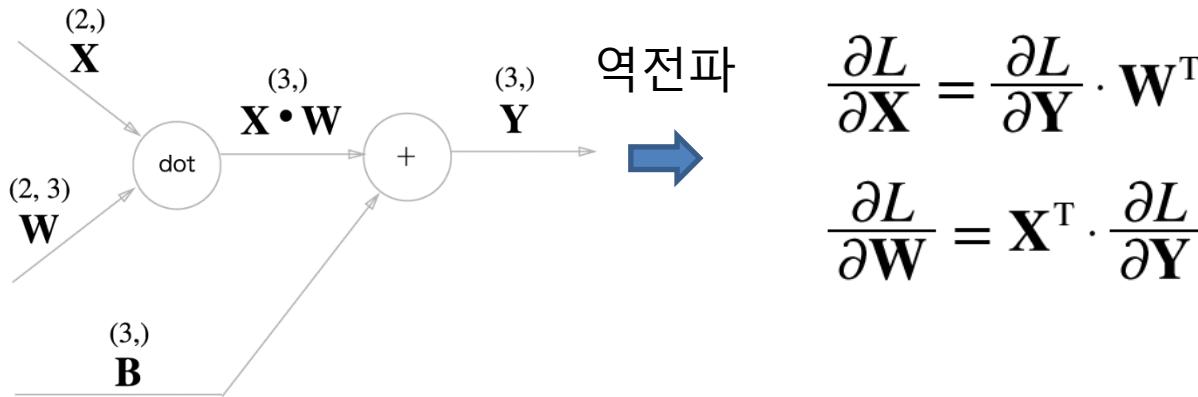


- 신경망의 순전파 때 수행하는 행렬의 내적은 기하학에서 어파인 변환(affine transformation)이라고 합니다.

Affine / Softmax 계층 구현

하기

- Affine 계층



- \mathbf{W}^T 의 T는 전치(transpose)행렬을 뜻합니다. 전치행렬은 \mathbf{W} 의 (i, j) 위치의 원소를 (j, i) 위치로 바꾼 것을 말합니다.

Affine / Softmax 계층 구현 하기



- Affine 계층

- \mathbf{W}^T 의 T는 전치(transpose)행렬을 뜻합니다. 전치행렬은 \mathbf{W} 의 (i, j) 위치의 원소를 (j, i) 위치로 바꾼 것을 말합니다.

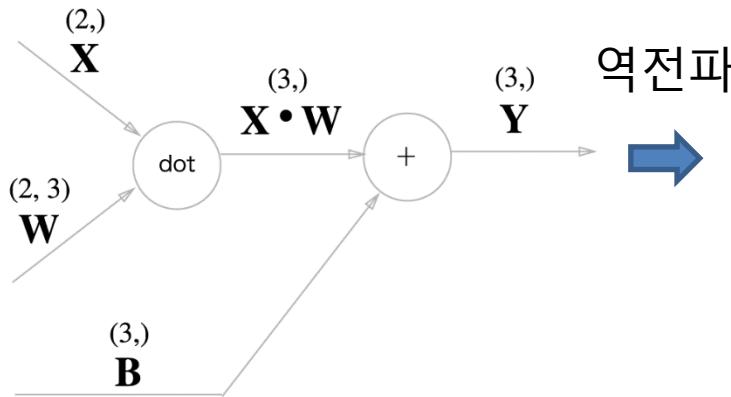
$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \end{pmatrix}$$

$$\mathbf{W}^T = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{pmatrix}$$

Affine / Softmax 계층 구현

하기

- Affine 계층



$$\frac{\partial L}{\partial \mathbf{X}} = \frac{\partial L}{\partial \mathbf{Y}} \cdot \mathbf{W}^T$$

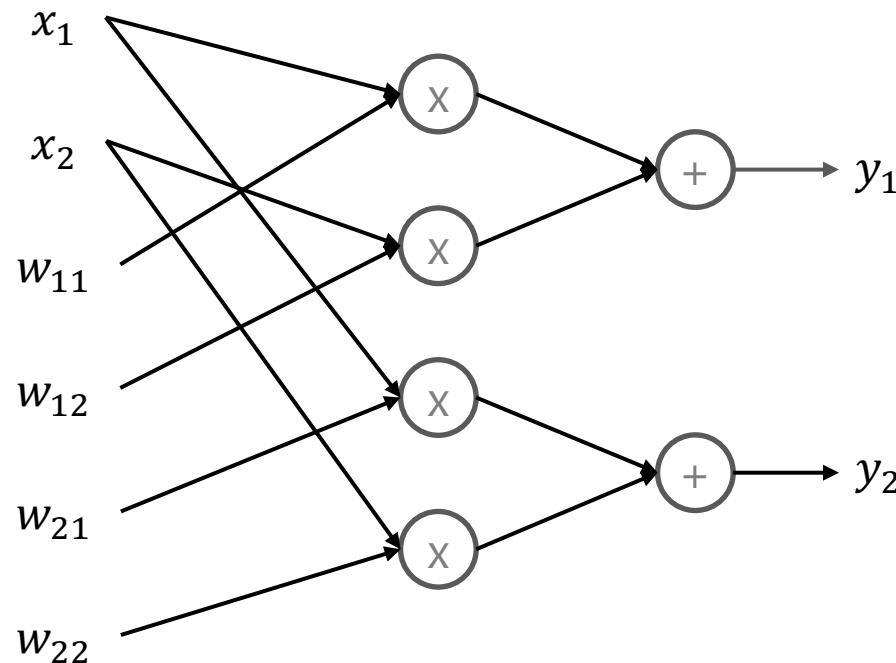
$$\frac{\partial L}{\partial \mathbf{W}} = \mathbf{X}^T \cdot \frac{\partial L}{\partial \mathbf{Y}}$$

이건 왜 이렇게 된건가요?

Computational Graph

- Affine 계층

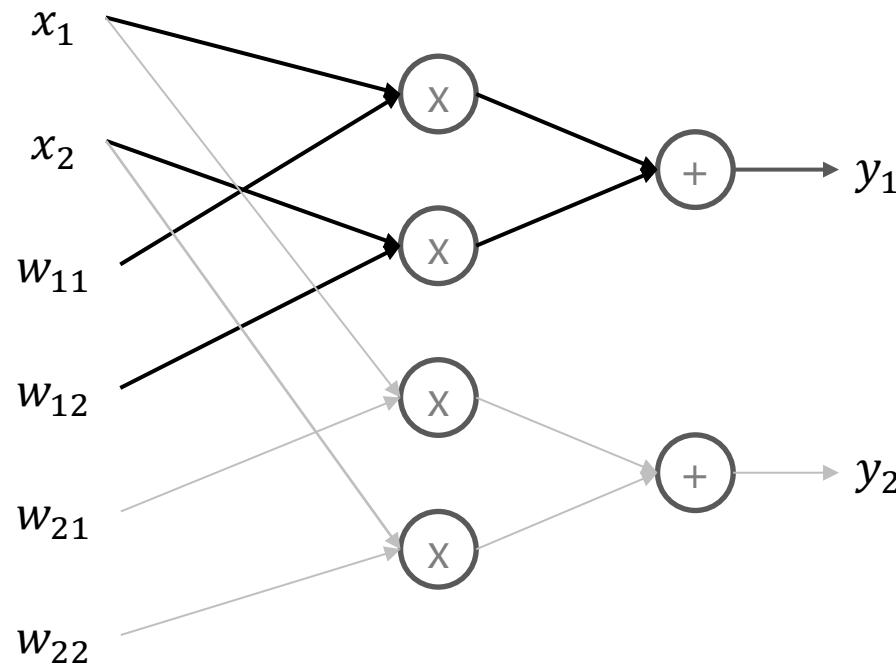
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

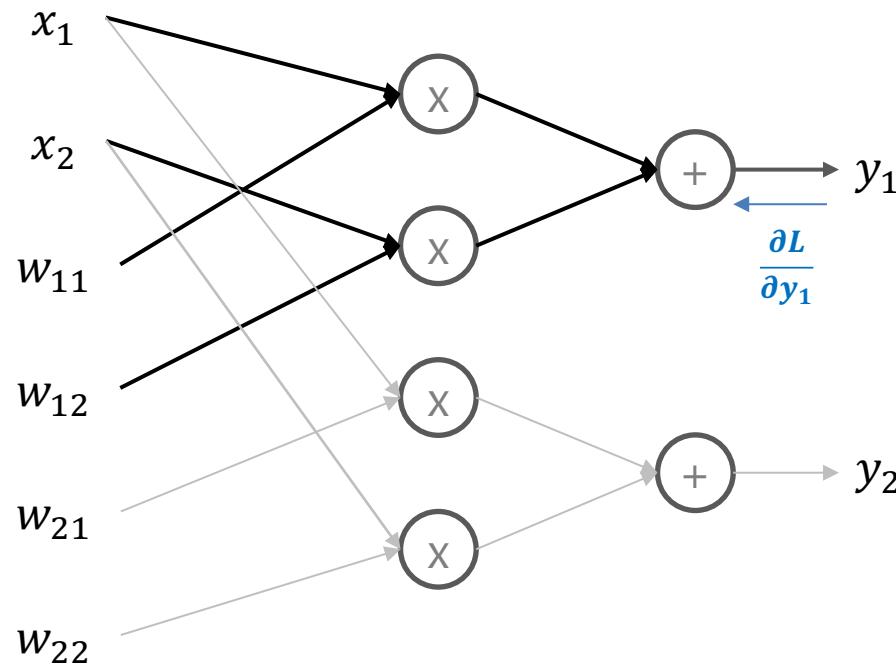
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} \\ w_{12} \end{bmatrix} + \begin{bmatrix} w_{21} \\ w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

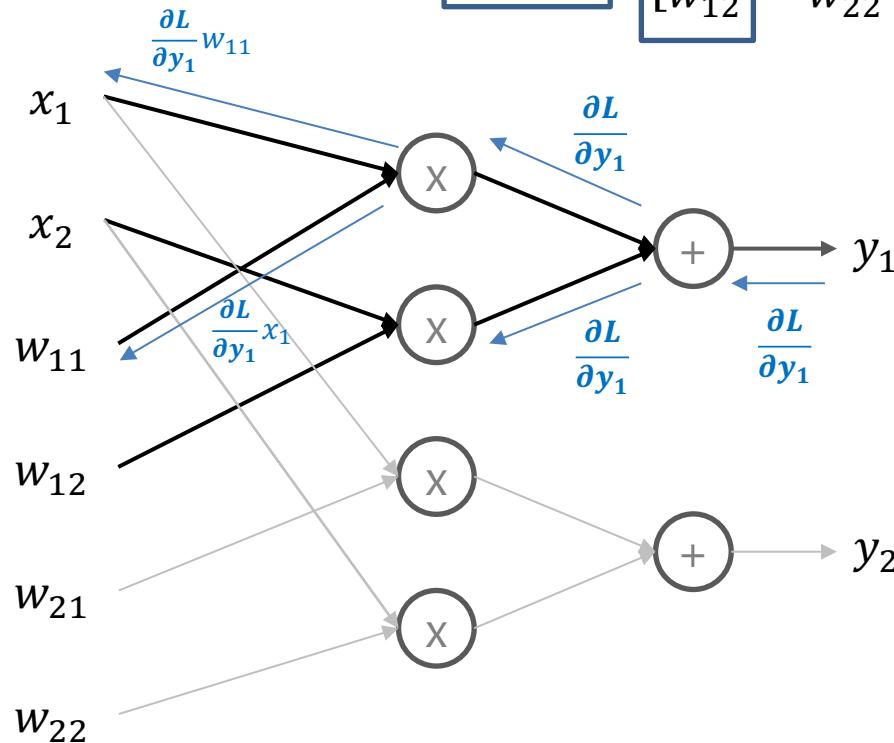
$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \times \begin{bmatrix} w_{11} \\ w_{12} \end{bmatrix} + \begin{bmatrix} w_{21} \\ w_{22} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$



Computational Graph

- Affine 계층

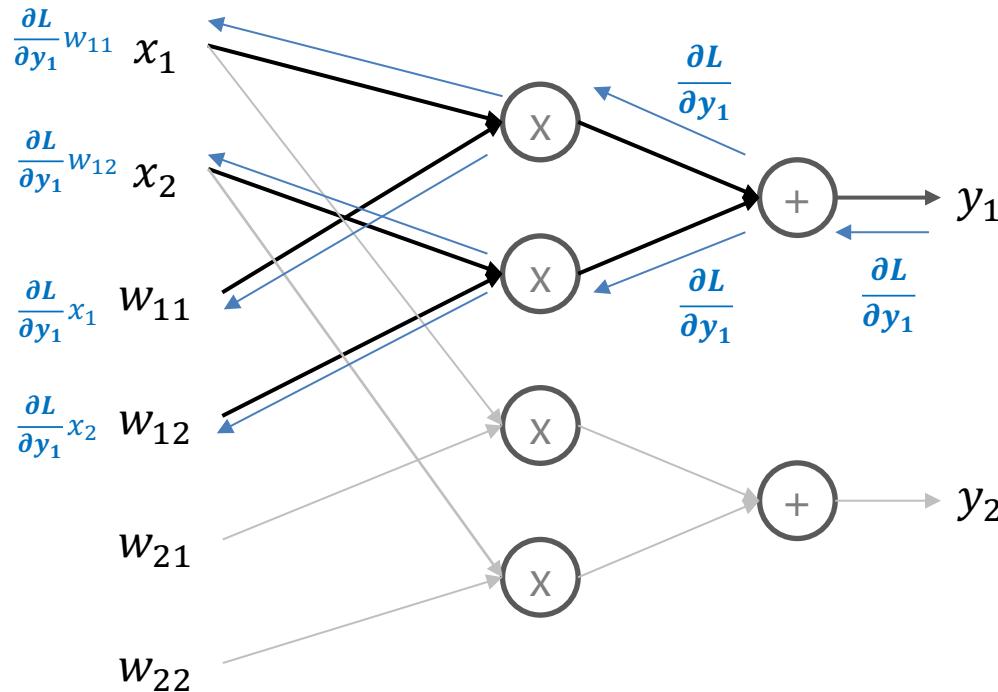
$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \times \begin{bmatrix} w_{11} \\ w_{12} \end{bmatrix} + \begin{bmatrix} w_{21} \\ w_{22} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$



Computational Graph

- Affine 계층

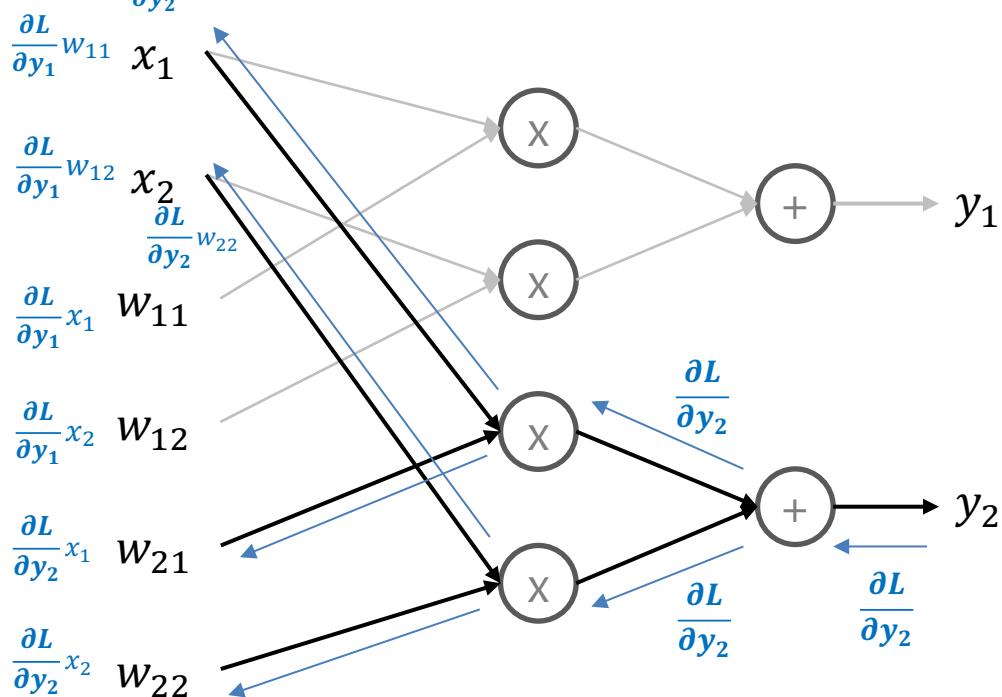
$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \times \begin{bmatrix} w_{11} \\ w_{12} \end{bmatrix} + \begin{bmatrix} w_{21} \\ w_{22} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$



Computational Graph

- Affine 계층

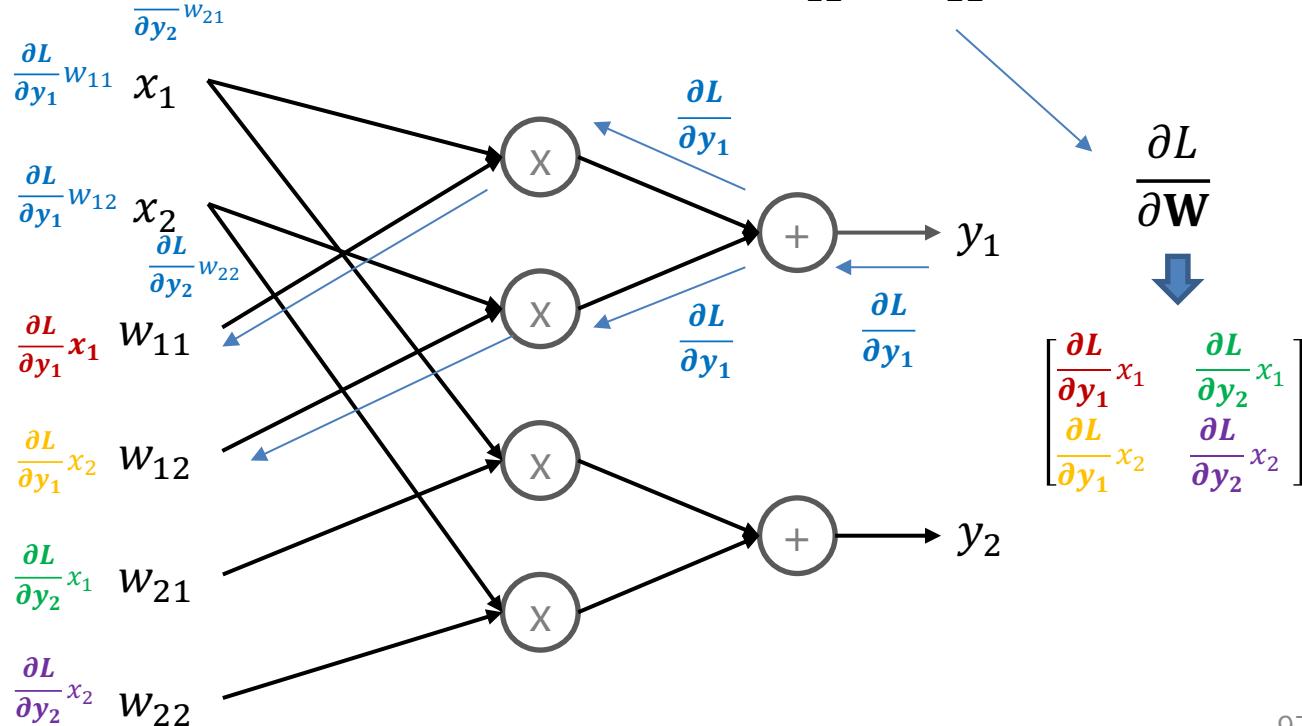
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} \\ w_{12} \\ w_{21} \\ w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

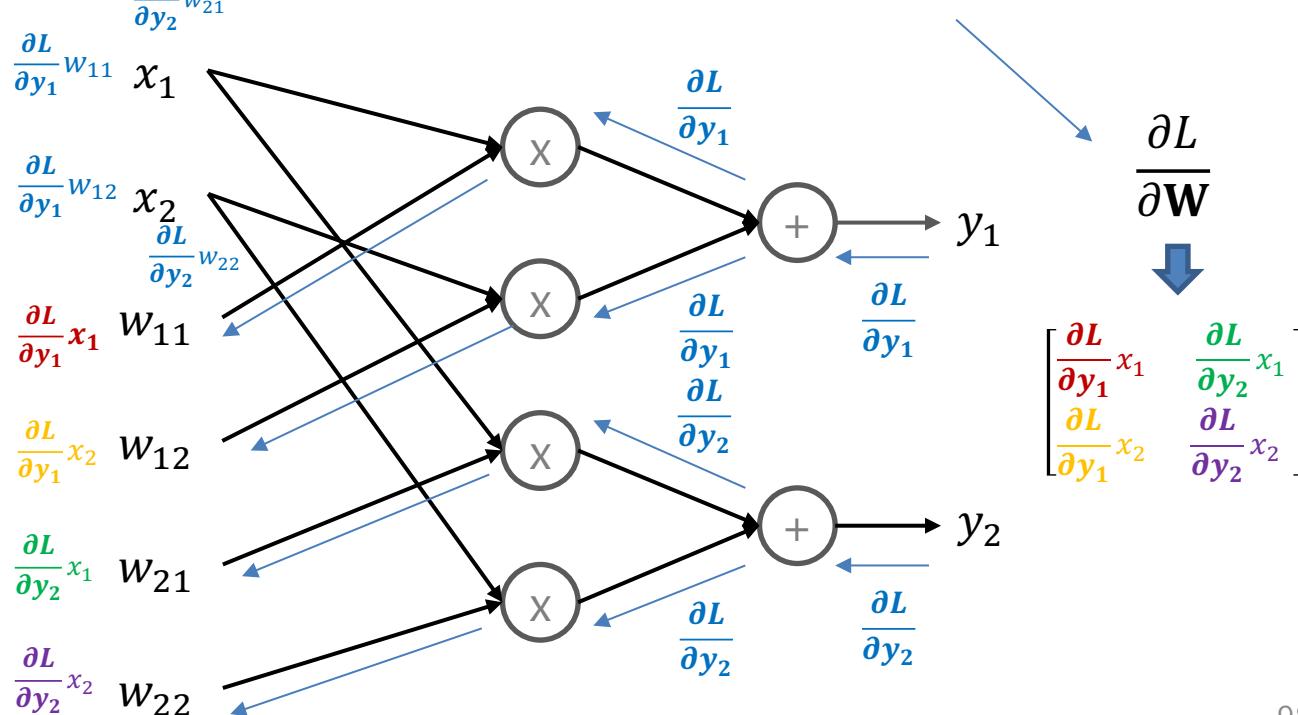
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

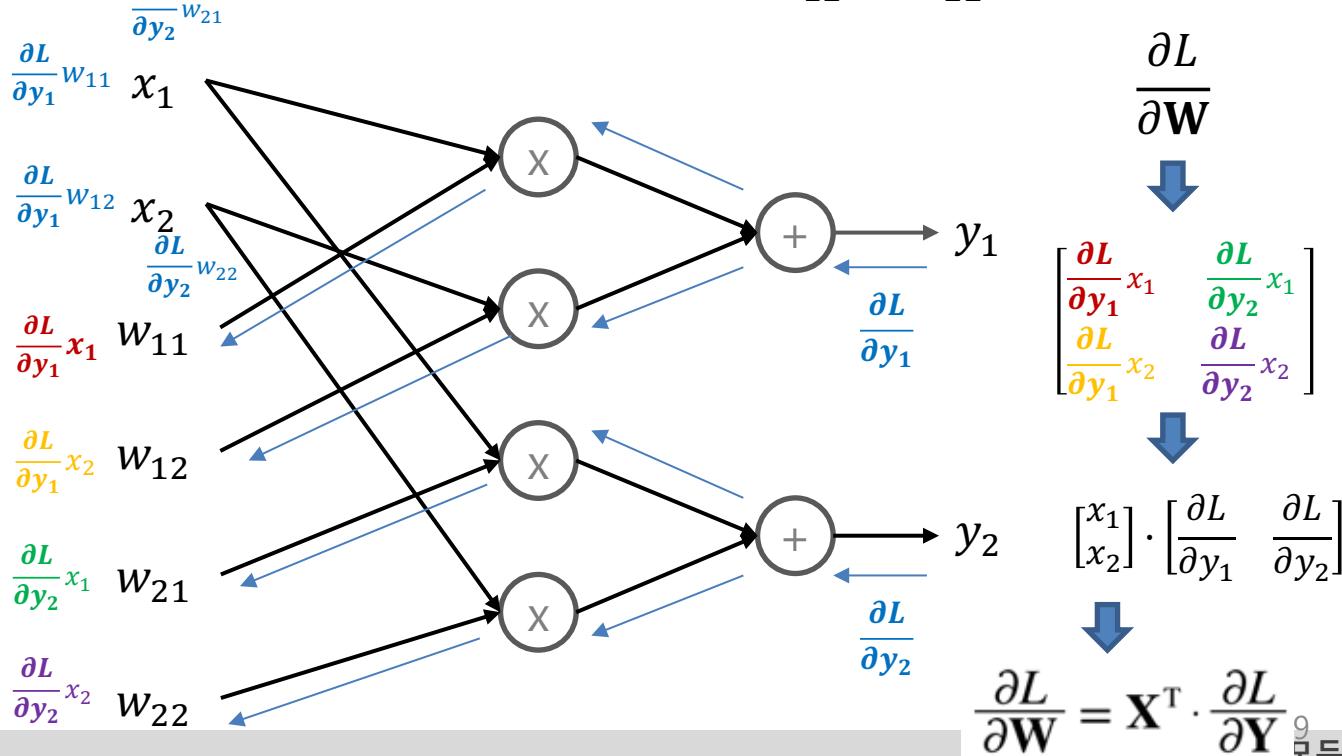
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$

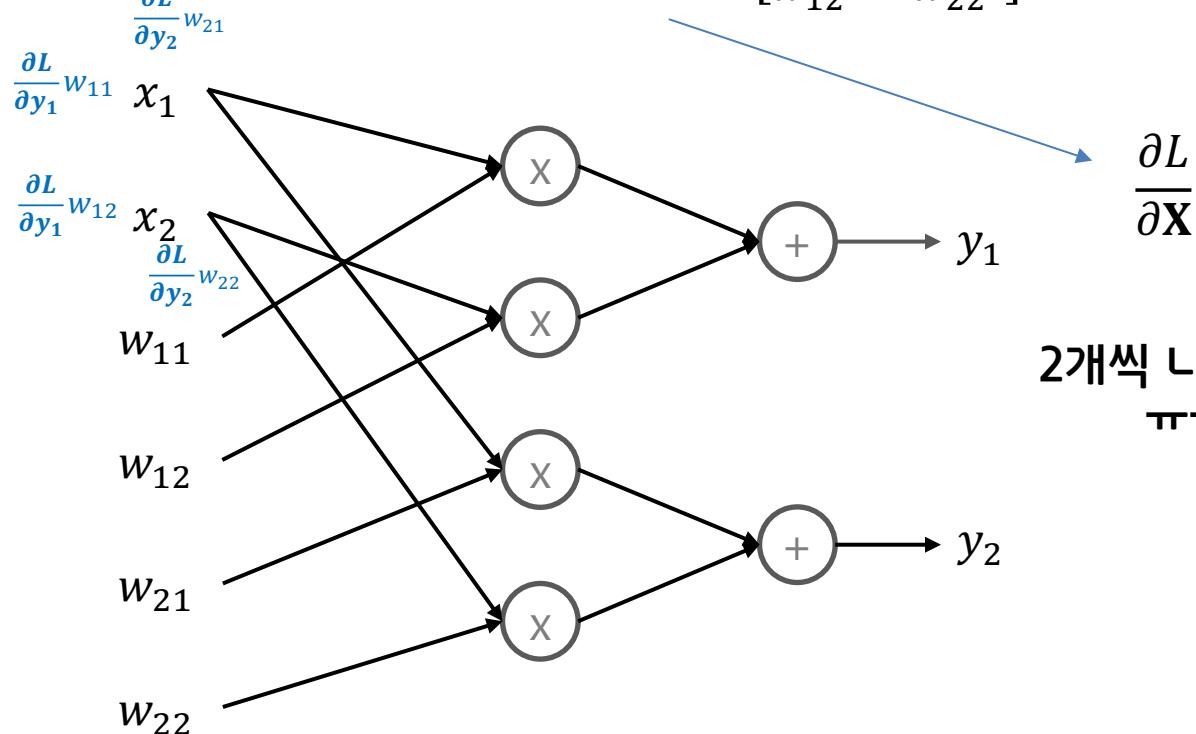


$$\frac{\partial L}{\partial \mathbf{W}} = \mathbf{X}^T \cdot \frac{\partial L}{\partial \mathbf{Y}}$$

Computational Graph

- Affine 계층

$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$

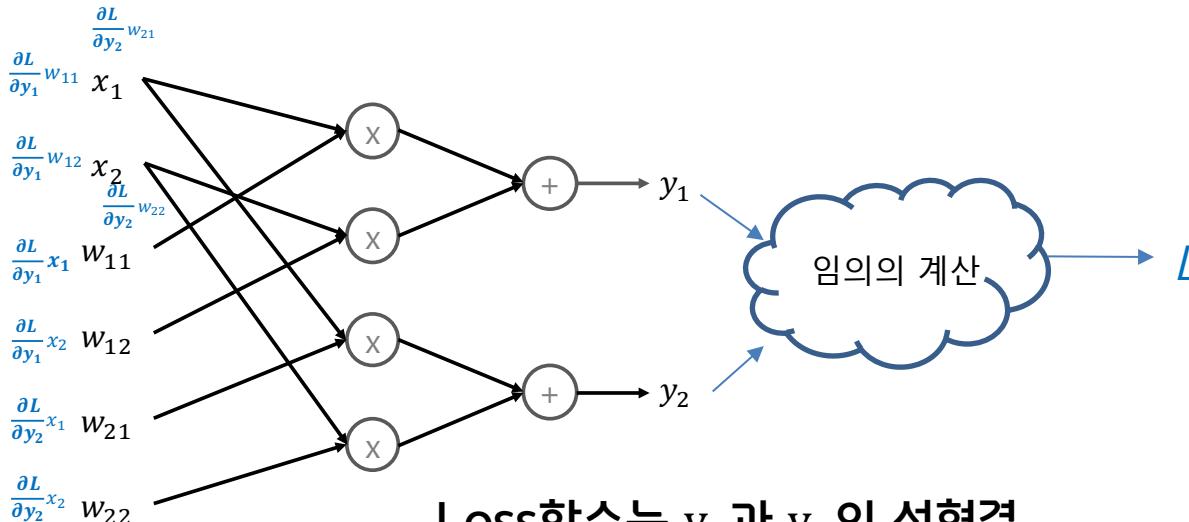


2개씩 나왔어요
ㅠㅠ

Computational Graph

- Affine 계층

$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



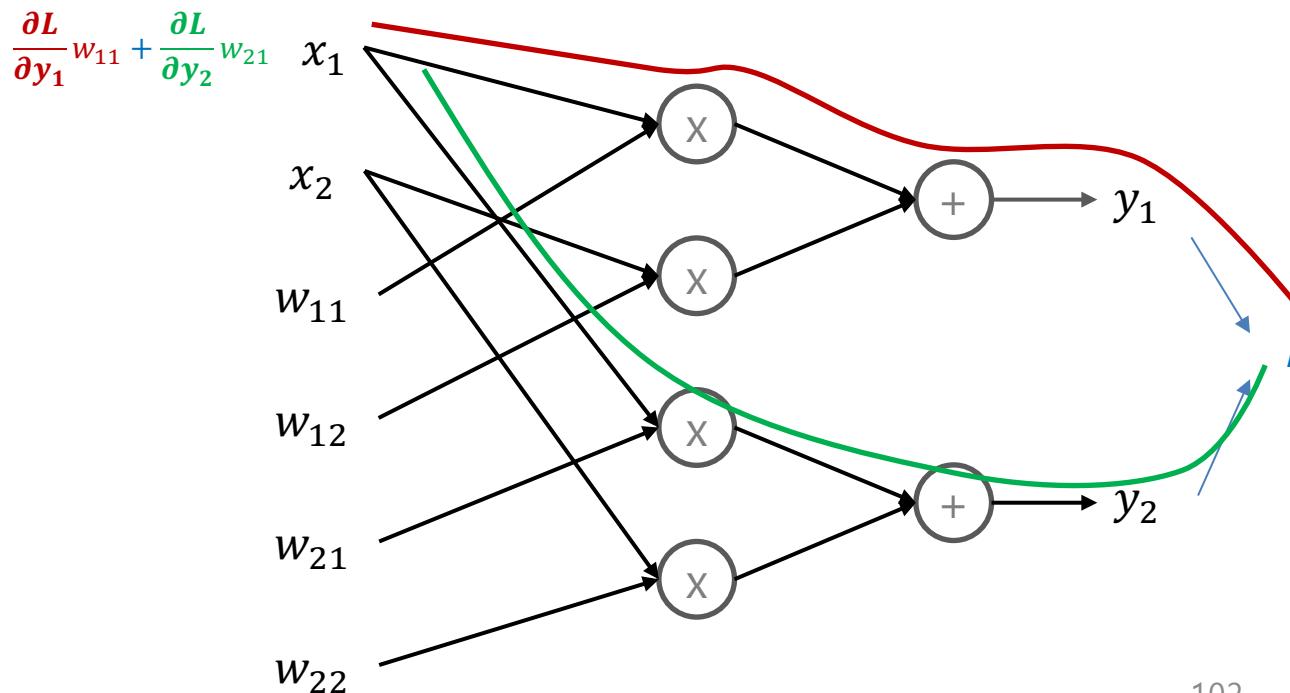
Loss함수는 y_1 과 y_2 의 선형결합일 겁니다

예) $L = ay_1 + by_2$

Computational Graph

- Affine 계층

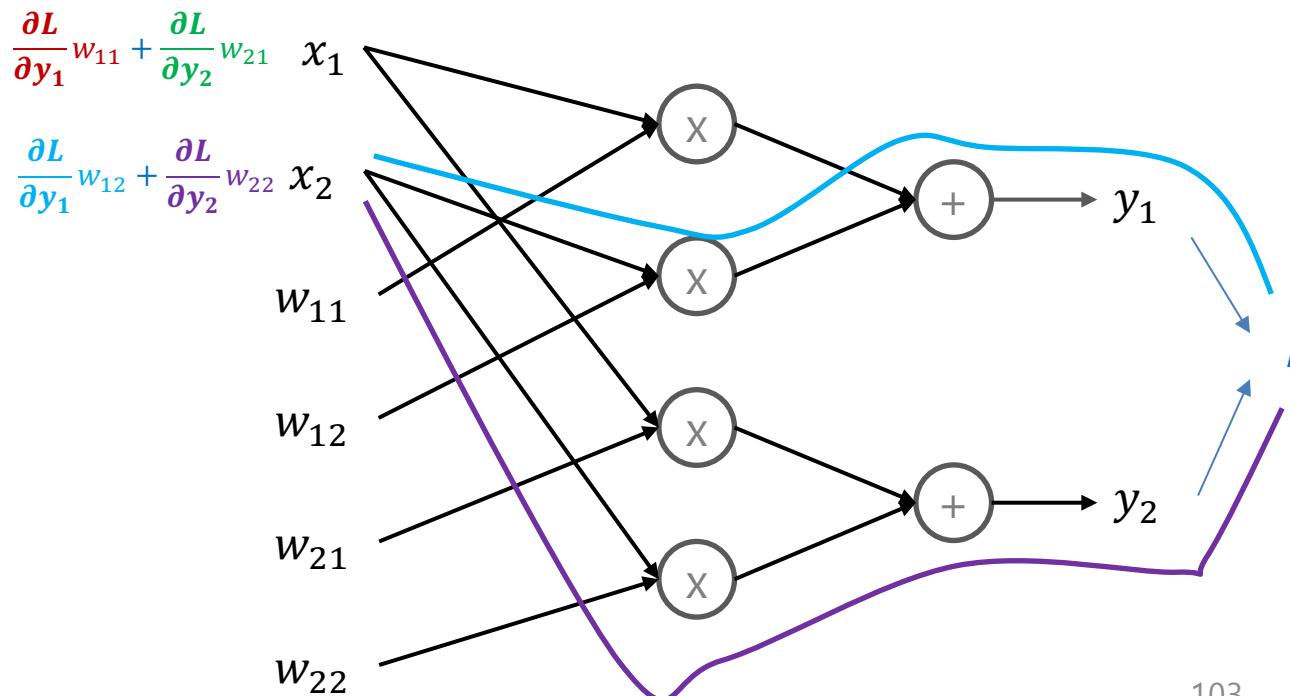
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

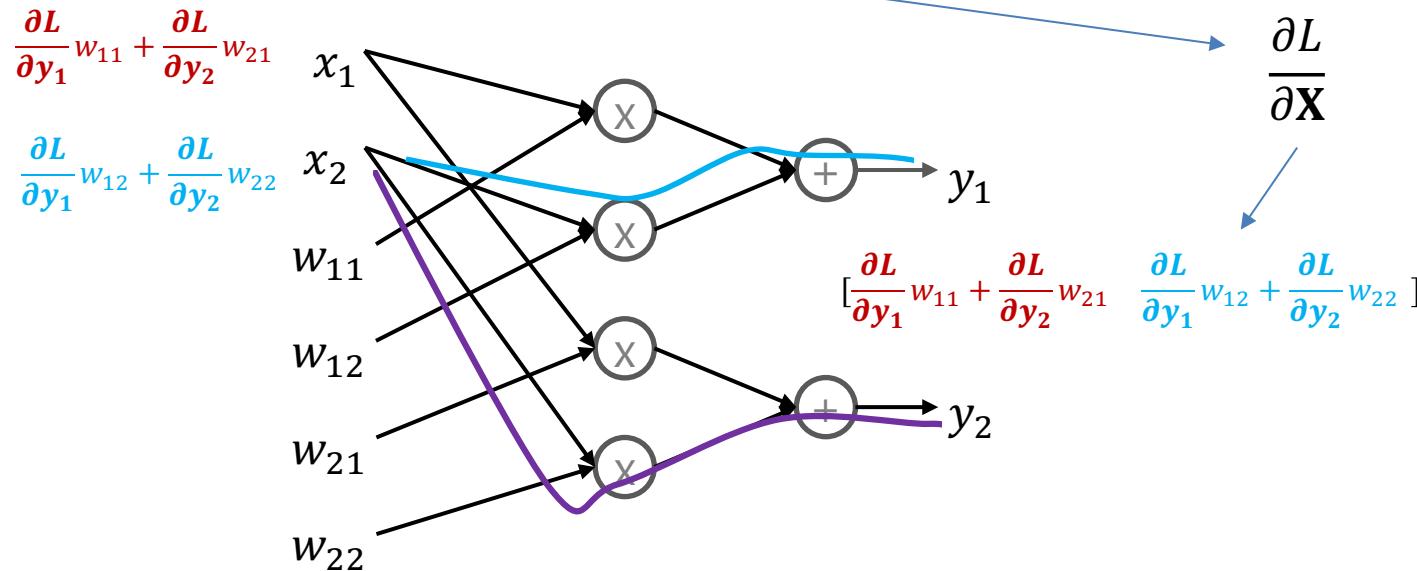
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$

$$\frac{\partial L}{\partial \mathbf{X}} = \left[\frac{\partial L}{\partial y_1} w_{11} + \frac{\partial L}{\partial y_2} w_{21} \quad \frac{\partial L}{\partial y_1} w_{12} + \frac{\partial L}{\partial y_2} w_{22} \right]$$



$$\left[\frac{\partial L}{\partial y_1} \quad \frac{\partial L}{\partial y_2} \right] \times \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

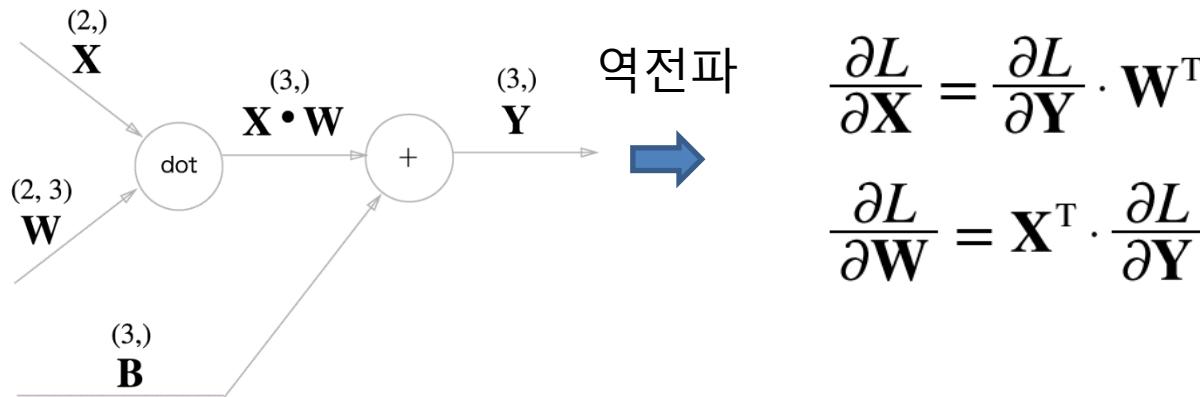


이제 다시 예제
로 돌아와 봅시
다

$$\frac{\partial L}{\partial \mathbf{X}} = \frac{\partial L}{\partial \mathbf{Y}} \cdot \mathbf{W}^T$$

Computational Graph

- Affine 계층



Computational Graph

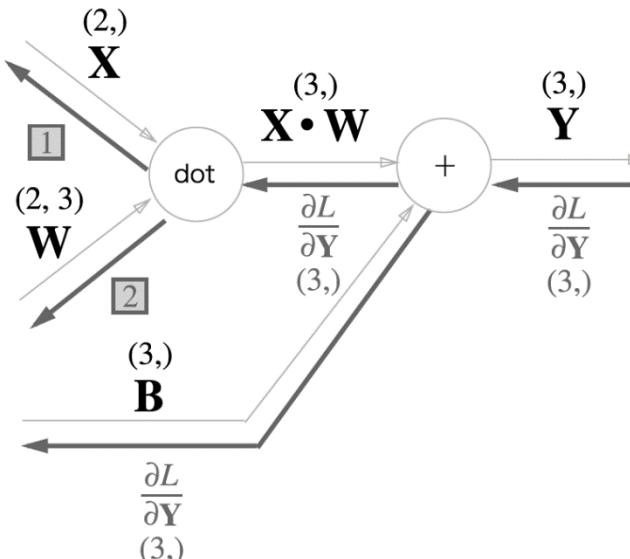
- Affine 계층

$$1 \quad \frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y} \quad W^T$$

(2,) (3,) (3, 2)

$$2 \quad \frac{\partial L}{\partial W} = X^T \quad \frac{\partial L}{\partial Y}$$

(2, 3) (2, 1) (1, 3)



매트릭스의 형상(shape)을 잘 생각해 보면 쉽습니다

Computational Graph

- Affine 계층

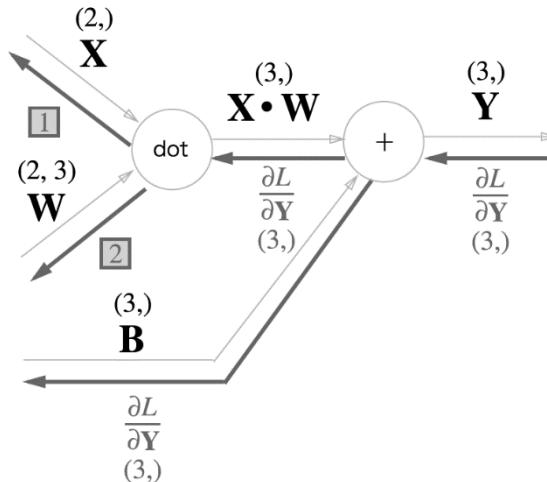
$$1 \quad \frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y} W^T$$

(2,) (3,) (3, 2)

$$2 \quad \frac{\partial L}{\partial W} = X^T \frac{\partial L}{\partial Y}$$

(2, 3) (2, 1) (1, 3)

X 와 $\frac{\partial L}{X}$ 와 형상이 같아야 하고 W 와 $\frac{\partial L}{W}$ 는 형상이 같아야 합니다



$$\mathbf{X} = (x_0, x_1, \dots, x_n)$$

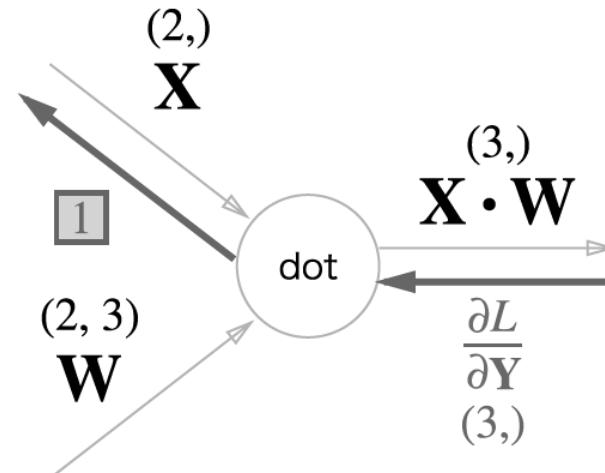
$$\frac{\partial L}{\partial \mathbf{X}} = \left(\frac{\partial L}{\partial x_0}, \frac{\partial L}{\partial x_1}, \dots, \frac{\partial L}{\partial x_n} \right)$$

Computational Graph

- Affine 계층

$$1 \quad \frac{\partial L}{\partial Y} \cdot W^T = \frac{\partial L}{\partial X}$$

(3,)	(3, 2)	(2,)
T	T T	↑



그냥 곱셈계층으로 이해하고 형상을 맞춰주면 됩니다

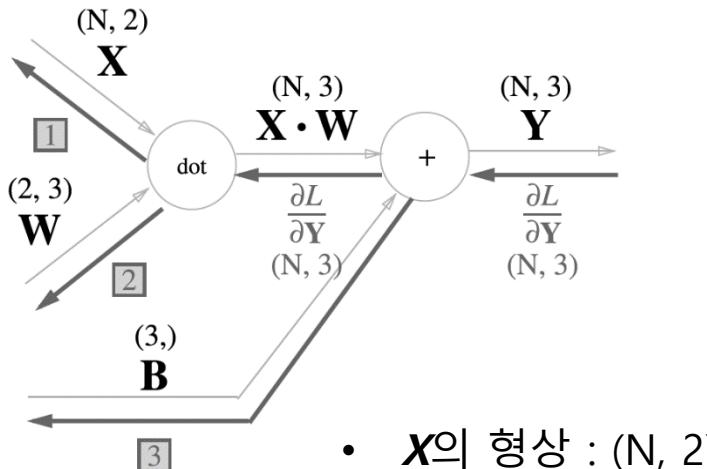
Computational Graph

- 배치용 Affine 계층

1 $\frac{\partial L}{\partial \mathbf{X}} = \frac{\partial L}{\partial \mathbf{Y}} \cdot \mathbf{W}^T$
 $(N, 2) \quad (N, 3) \quad (3, 2)$

2 $\frac{\partial L}{\partial \mathbf{W}} = \mathbf{X}^T \cdot \frac{\partial L}{\partial \mathbf{Y}}$
 $(2, 3) \quad (2, N) \quad (N, 3)$

3 $\frac{\partial L}{\partial \mathbf{B}} = \frac{\partial L}{\partial \mathbf{Y}}$ 의 첫 번째 축(0축, 열방향)의 합
 $(3) \quad (N, 3)$



- 편향에 주의하세요

Computational Graph

- 배치용 Affine 계층
 - 데이터가 2개 일 경우($N=2$)의 편향은 계산된 각각의 결과에 더해집니다

순전파의 경우

```
In [19]: X_dot_W = np.array([[0, 0, 0],[10, 10, 10]])
```

```
In [20]: B = np.array([1, 2, 3])
```

```
In [21]: X_dot_W
```

```
Out[21]:
```

```
array([[ 0,  0,  0], ← 데이터 1  
      [10, 10, 10]])← 데이터 2
```

```
In [22]: X_dot_W + B
```

```
Out[22]:
```

```
array([[ 1,  2,  3],  
      [11, 12, 13]])
```

Computational Graph

- 배치용 Affine 계층
 - 데이터가 2개 일 경우($N=2$)의 편향은 계산된 각각의 결과에 더해집니다

역전파의 경우

```
In [23]: dY = np.array([[1,2,3],[4,5,6]])
```

```
In [24]: dY
```

```
Out[24]:
```

```
array([[1, 2, 3],  
       [4, 5, 6]])
```

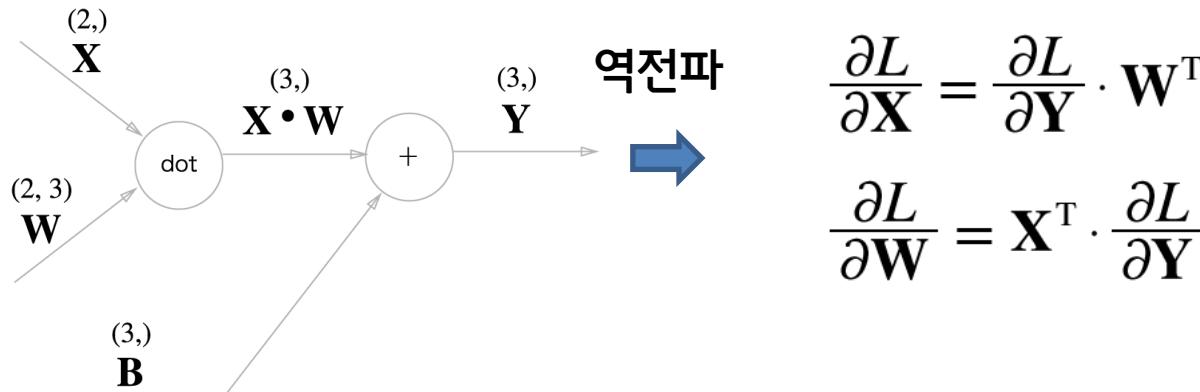
```
In [25]: dB = np.sum(dY, axis=0)
```

```
In [26]: dB
```

```
Out[26]: array([5, 7, 9])
```

Computational Graph

- Affine 계층



w 매트릭스가 업데이트 되겠군요

$$w = w - \eta \frac{\partial L}{\partial w}$$

신경망 관련 계층구현

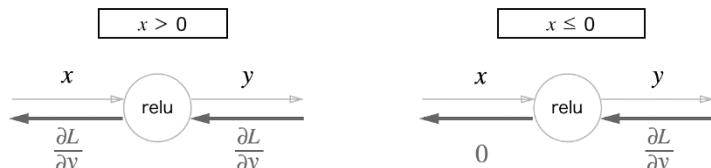
활성화 함수계층 구현하기

- ReLU 계층

$$y = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases}$$



$$\frac{\partial y}{\partial x} = \begin{cases} 1 & (x > 0) \\ 0 & (x \leq 0) \end{cases}$$



```

class Relu:
    def __init__(self):
        백워드 용 self.mask = None

    def forward(self, x):
        백워드 용 self.mask = (x <= 0)
        저장      out = x.copy()
                  out[self.mask] = 0

        return out

    def backward(self, dout):
        저장된 마
        스크 이용 dout[self.mask] = 0
                  dx = dout

        return dx

```

활성화 함수계층 구현하기

- ReLU 계층

```
In [12]: x = np.array([[1.0, -0.5],[-2.0, 3.0]])  
  
In [13]: print(x)  
[[ 1. -0.5]  
 [-2.  3.]]  
  
In [14]: mask = (x<=0)  
  
In [15]: print(mask)  
[[False True]  
 [ True False]]  
  
In [16]: x[mask] = 0  
  
In [17]: print(x)  
[[ 1.  0.]  
 [ 0.  3.]]
```

```
class Relu:  
    def __init__(self):  
        백워드 용 self.mask = None  
  
    def forward(self, x):  
        백워드 용 self.mask = (x <= 0)  
        저장 out = x.copy()  
        out[self.mask] = 0  
  
        return out  
  
    def backward(self, dout):  
        저장된 마  
        스크 이용 dout[self.mask] = 0  
        dx = dout  
  
        return dx
```

활성화 함수계층 구현하기

- Sigmoid 계층 : $y = \frac{1}{1 + \exp(-x)}$
- /codes/common/layers.py

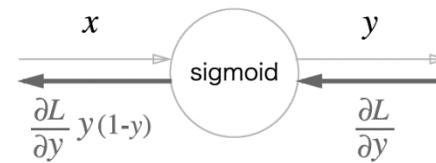
```

class Sigmoid:
    def __init__(self):
        self.out = None

    def forward(self, x):
        out = sigmoid(x)
        self.out = out    출력 y를 저장
        return out

    def backward(self, dout):
        dx = dout * (1.0 - self.out) * self.out
        계산
        return dx

```



실습 1 : 2 레이어 네트워크 실습

- 모듈화 하지 않은 2레이어 역전파 구현
- 1) 실행 파일 (손글씨 인식) : 2day/prac1_train_neuralnet.py

```
24 # 1에폭당 반복 수
25 iter_per_epoch = max(train_size / batch_size, 1)
26
27 for i in range(iters_num):
28     # 미니배치 획득
29     batch_mask = np.random.choice(train_size, batch_size)
30     x_batch = x_train[batch_mask]
31     t_batch = t_train[batch_mask]
32
33     # 기울기 계산
34     #grad = network.numerical_gradient(x_batch, t_batch)
35     grad = network.gradient(x_batch, t_batch)
36
37     # 매개변수 갱신
38     for key in ('W1', 'b1', 'W2', 'b2'):
39         network.params[key] -= learning_rate * grad[key]
40
41     # 학습 결과 기록
42     loss = network.loss(x_batch, t_batch)
43     train_loss_list.append(loss)
44
45     # 1에폭당 정확도 계산
46     if i % iter_per_epoch == 0:
47         train_acc = network.accuracy(x_train, t_train)
48         test_acc = network.accuracy(x_test, t_test)
49         train_acc_list.append(train_acc)
50         test_acc_list.append(test_acc)
51         print("train acc, test acc | " + str(train_acc) + ", " + str(test_acc))
```

구현할 함수

실습 1: 2 레이어 네트워크 실습

- 모듈화 하지 않은 2레이어 역전파 구현
- 2) 구현할 함수 : 2day/prac1_simple_two_Layer_net.py

```
55     def gradient(self, x, t):  
56         W1, W2 = self.params['W1'], self.params['W2']  
57         b1, b2 = self.params['b1'], self.params['b2']  
58         grads = {}  
59  
60         batch_num = x.shape[0]  
61  
62         # forward  
63         a1 = np.dot(x, W1) + b1  
64         z1 = sigmoid(a1)  
65         a2 = np.dot(z1, W2) + b2  
66         y = softmax(a2)  
67  
68         # backward  
69         dy = (y - t) / batch_num  
70         ##### Write your codes #####  
71         ##### Write your codes #####  
72         grads['W2'] = None  
73         grads['b2'] = None  
74  
75         da1 = None  
76         dz1 = None  
77         grads['W1'] = None  
78         grads['b1'] = None  
79  
80     return grads  
81
```

실습. 1 : 2 레이어 네트워크 실습

모듈화 하지 않은 2레이어 역전파 구현



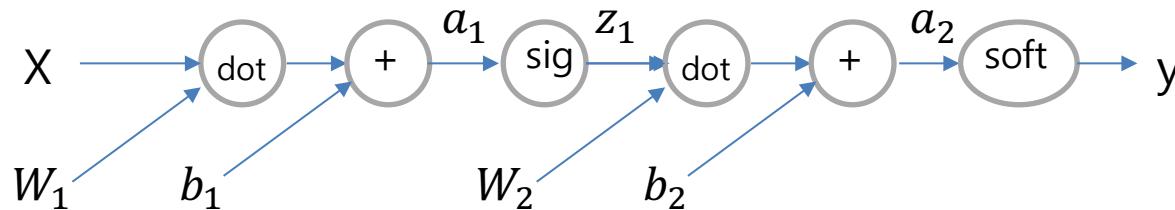
- 2) 구현할 함수 : 2day/prac1_simple_two_Layer_net.py
 - 순전파를 계산 그래프로 표현해 봅시다

```
55     def gradient(self, x, t):  
56         W1, W2 = self.params['W1'], self.params['W2']  
57         b1, b2 = self.params['b1'], self.params['b2']  
58         grads = {}  
59  
60         batch_num = x.shape[0]  
61  
62         # forward  
63         a1 = np.dot(x, W1) + b1  
64         z1 = sigmoid(a1)  
65         a2 = np.dot(z1, W2) + b2  
66         y = softmax(a2)
```

실습 1 : 2 레이어 네트워크 실습

- 2) 구현할 함수 : 2day/prac1_simple_two_Layer_net.py

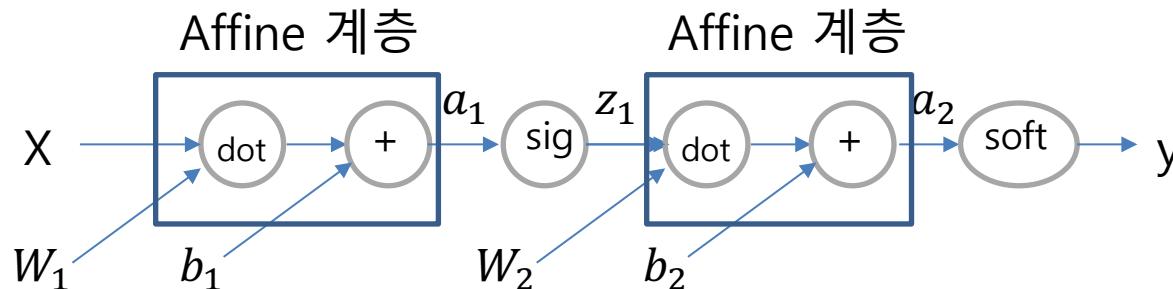
```
62          # forward  
63      a1 = np.dot(x, W1) + b1  
64      z1 = sigmoid(a1)  
65      a2 = np.dot(z1, W2) + b2  
66      y = softmax(a2)
```



실습 1 : 2 레이어 네트워크 실습

- 2) 구현할 함수 : 2day/prac1_simple_two_Layer_net.py

```
62      # forward
63      a1 = np.dot(x, W1) + b1
64      z1 = sigmoid(a1)
65      a2 = np.dot(z1, W2) + b2
66      y = softmax(a2)
```

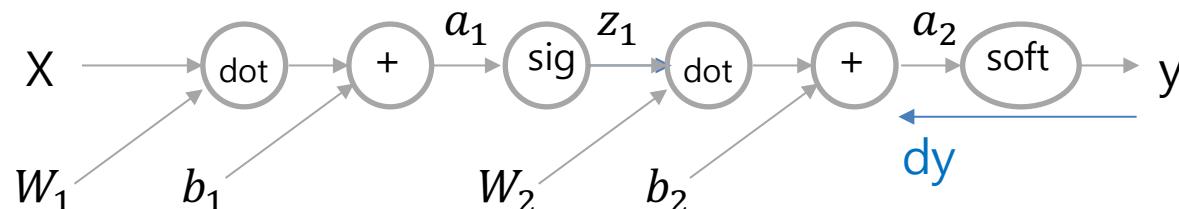
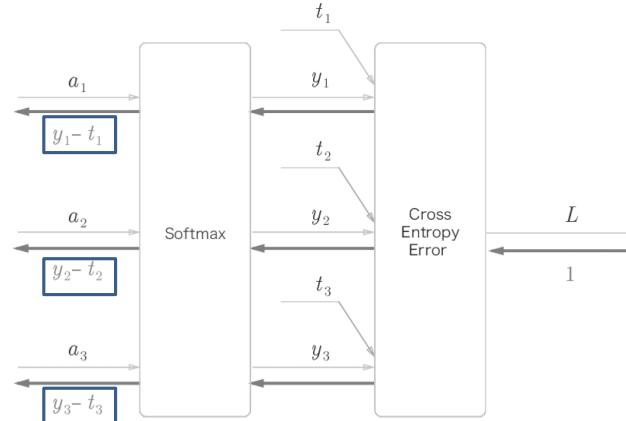


실습 1 : 2 레이어 네트워크 실습

- 2) 구현할 함수 : 2day/prac1_simple_two_Layer_net.py

Softmax-with-Loss의 역전파를 기억해 봅시다

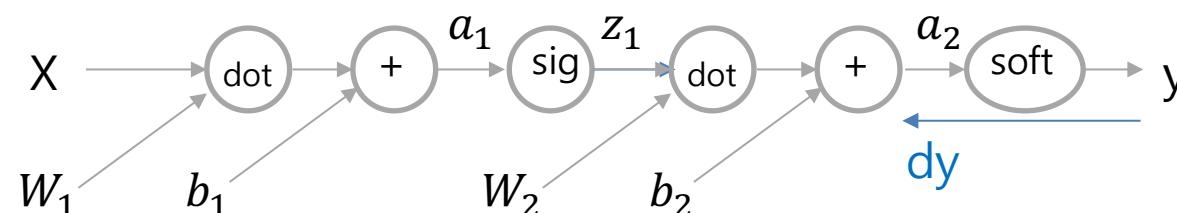
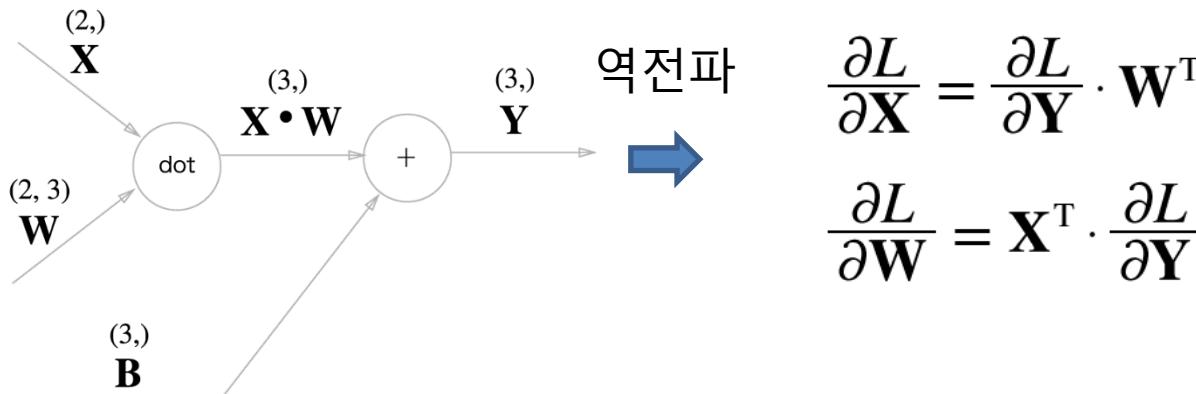
```
68      # backward
69      dy = (y - t) / batch_num
70      ##### Write your codes #####
71      #####
72      grads['W2'] = None
73      grads['b2'] = None
74
75      da1 = None
76      dz1 = None
77      grads['W1'] = None
78      grads['b1'] = None
79
80
81      return grads
```



실습 1 : 2 레이어 네트워크 실습

- 2) 구현할 함수 : 2day/prac1_simple_two_Layer_net.py

Affine 계층의 역전파를 기억해 봅시다



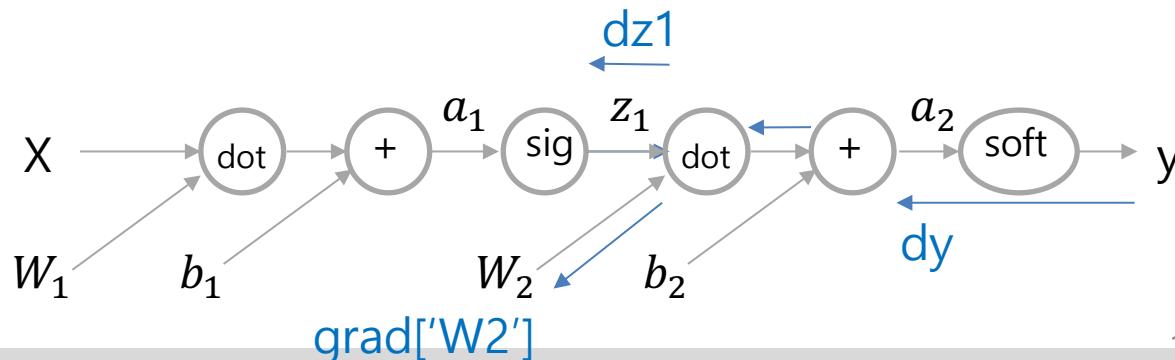
실습 1 : 2 레이어 네트워크 실습

- 2) 구현할 함수 : 2day/prac1_simple_two_Layer_net.py
Affine 계층의 역전파를 기억해 봅시다

```
68     # backward
69     dy = (y - t) / batch_num
70     ##### Write your codes #####
71     #####
72     #####
73     grads['W2'] = None
74     grads['b2'] = None
75
76     dz1 = None
77     da1 = None
78     grads['W1'] = None
79     grads['b1'] = None
```

$$\frac{\partial L}{\partial \mathbf{X}} = \frac{\partial L}{\partial \mathbf{Y}} \cdot \mathbf{W}^T$$

$$\frac{\partial L}{\partial \mathbf{W}} = \mathbf{X}^T \cdot \frac{\partial L}{\partial \mathbf{Y}}$$



실습 1 : 2 레이어 네트워크 실습

- 2) 구현할 함수 : 2day/prac1_simple_two_Layer_net.py

Affine 계층의 배치데이터 편향을 기억해 봅시다

```
68     # backward
69     dy = (y - t) / batch_num
70     ##### Write your codes #####
71
72     #####
73     grads['W2'] = None
74     grads['b2'] = None
75
76     dz1 = None
77     da1 = None
78     grads['W1'] = None
79     grads['b1'] = None
```

Affine / Softmax 계층 구현하기

- 배치용 Affine 계층
- 데이터가 2개 일 경우($N=2$)의 편향은 계산된 각각의 결과에 더해집니다

```
In [23]: dY = np.array([[1,2,3],[4,5,6]])
```

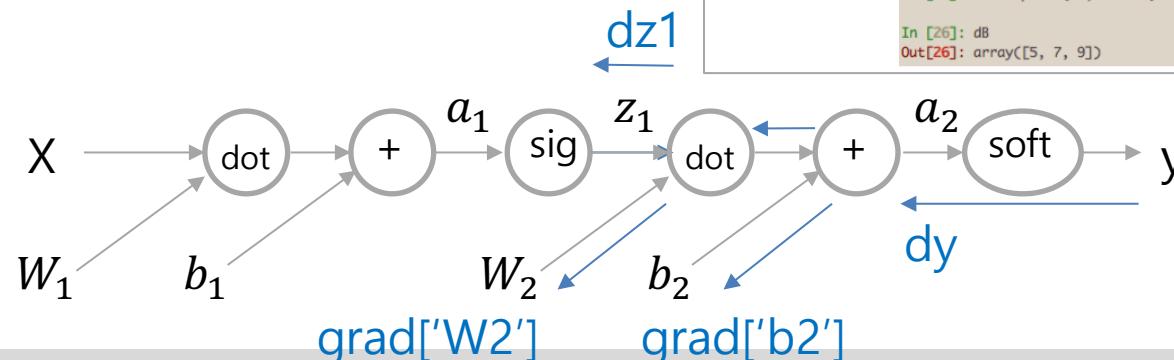
```
In [24]: dy
```

```
Out[24]: array([1, 2, 3],  
[4, 5, 6])
```

```
In [25]: dB = np.sum(dY, axis=0)
```

```
In [26]: dB  
Out[26]: array([5, 7, 9])
```

역전파의 경우

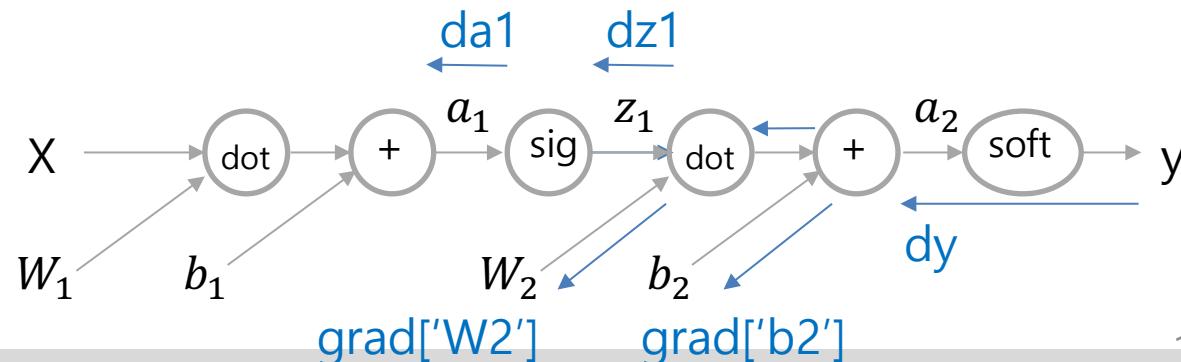
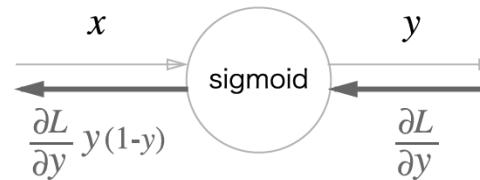


실습 1 : 2 레이어 네트워크 실습

- 2) 구현할 함수 : 2day/prac1_simple_two_Layer_net.py

Sigmoid 역전파 함수를 기억해 보세요

```
68     # backward
69     dy = (y - t) / batch_num
70     ##### Write your codes #####
71     #####
72     grads['W2'] = None
73     grads['b2'] = None
74
75     dz1 = None
76     da1 = None
77     grads['W1'] = None
78     grads['b1'] = None
79
```



실습 1 : 2 레이어 네트워크 실습

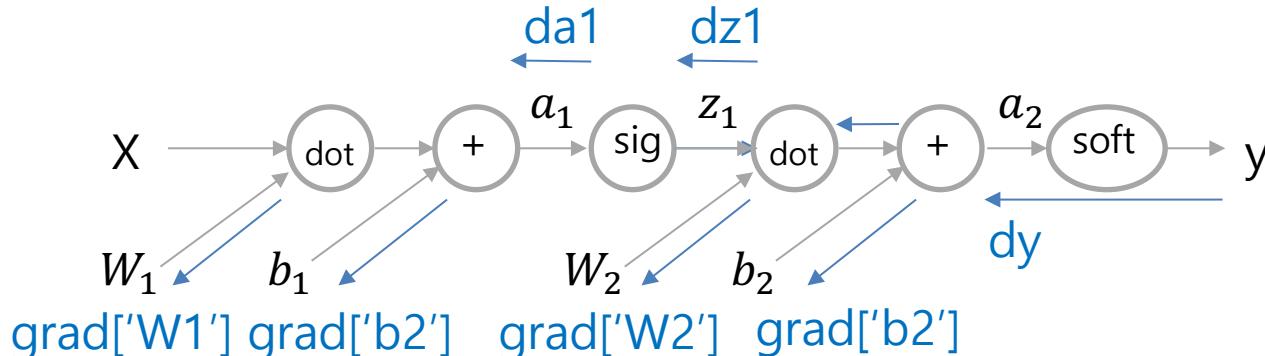
- 2) 구현할 함수 : 2day/prac1_simple_two_Layer_net.py

Affine 계층의 역전파를 기억해 보세요.

```
68  # backward
69  dy = (y - t) / batch_num
70  ##### Write your codes #####
71  #####
72  #####
73  grads['W2'] = None
74  grads['b2'] = None
75
76  dz1 = None
77  da1 = None
78  grads['W1'] = None
79  grads['b1'] = None
```

$$\frac{\partial L}{\partial \mathbf{X}} = \frac{\partial L}{\partial \mathbf{Y}} \cdot \mathbf{W}^T$$

$$\frac{\partial L}{\partial \mathbf{W}} = \mathbf{X}^T \cdot \frac{\partial L}{\partial \mathbf{Y}}$$



지금까지 살펴본 내용



- 분류기의 구성
 - Score function : $Wx+b$
 - Loss function : Score Function의 잘못 분류된 정도를 측정
 - Optimization : Loss function의 값을 줄이는 방향으로 파라미터 업데이트

$$w = w - \eta \frac{\partial L}{\partial w}$$



박 은 수 Research Director

E-mail : es.park@modulabs.co.kr