모두의연구소

Optimizaition

모두의연구소
박은수 Research Director

지난시간 돌아보기 ...

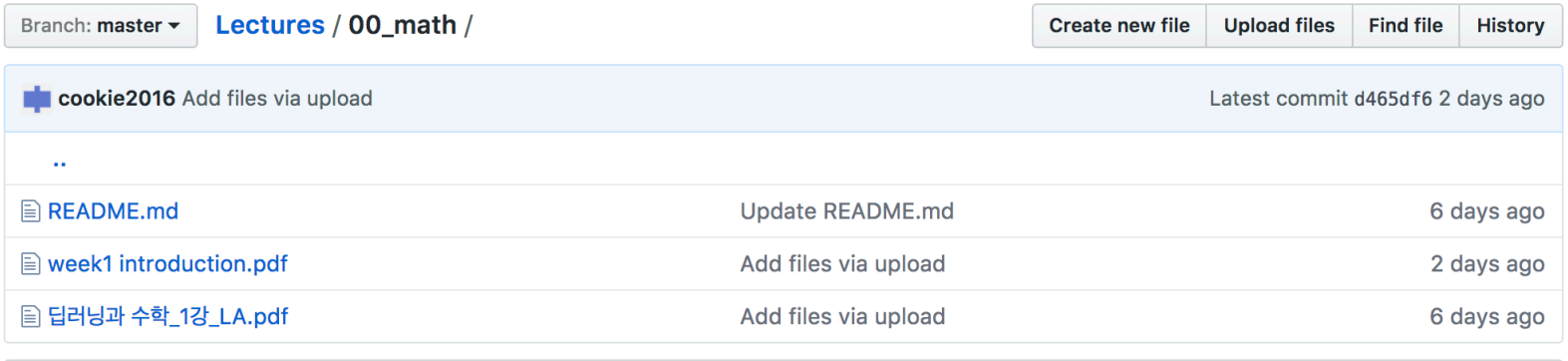
모두의연구소

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

지난시간 돌아보기 ...

모두의연구소

- #lec_math

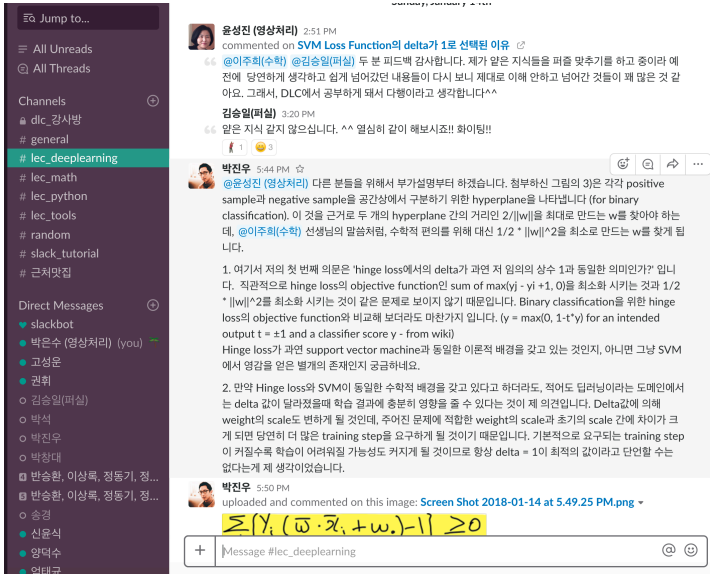


강의 슬라이드 뿐만아니라 그 설명도 볼 수 있습니다

지난시간 돌아보기 ...

모두의연구소

- #lec_deeplearning



엄청난 논의들

환영합니다

지난시간 돌아보기 ...

모두의연구소

- #lec_deeplearning

Distance 를 측정

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

test image			
56	32	10	18
90	23	128	133
24	26	178	200
2	0	255	220

\-

training image			
10	20	24	17
8	10	89	100
12	16	178	170
4	32	233	112

=

pixel-wise absolute value differences			
46	12	14	1
82	13	39	33
12	10	0	30
2	32	22	108

add → 456

가장 간단한
Distance 측정

지난시간 돌아보기 ...

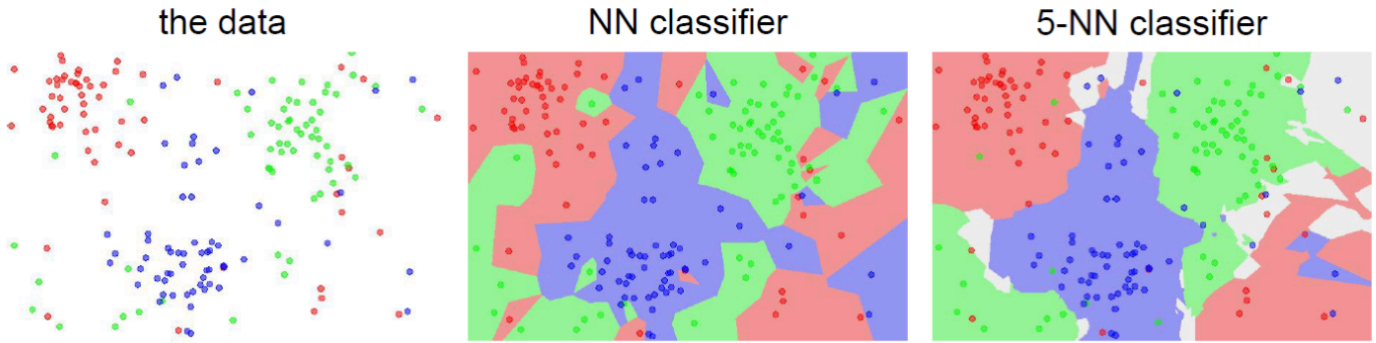
모두의연구소

- #lec_deeplearning

처음 만난
이미지 분류기

k-Nearest Neighbor은 무엇인가?

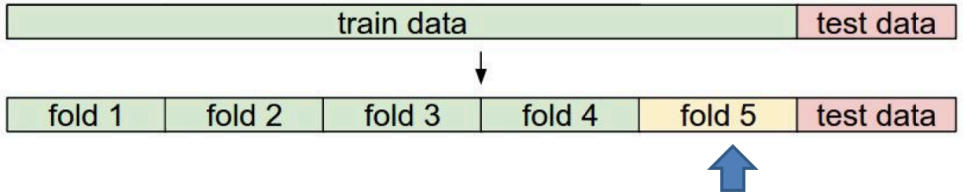
k개의 Nearest Neighbor를 찾고, k개 중 많이 나온 레이블을 선택 (voting)



지난시간 돌아보기 ...

모두의연구소

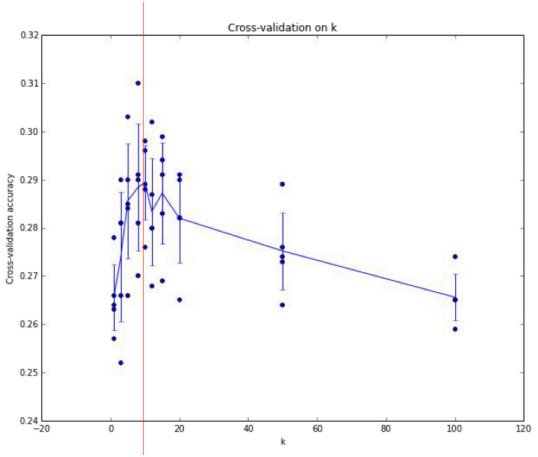
- #lec_deeplearning



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

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

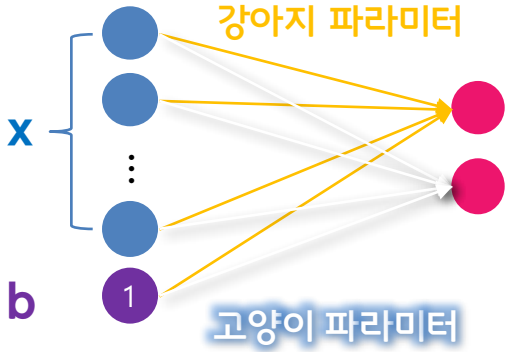
Cross validation



음.. k=7 이 좋겠군

지난시간 돌아보기 …

모두의연구소



$$\underset{2\times1}{f(x, W)} = \underset{2\times3,072}{W}\underset{3,072\times1}{x} + \underset{2\times1}{b}$$

32x32x3 영상 = 3,072 픽셀

강아지 파라미터	0.2	0.1	…	0.3	0.7
고양이 파라미터	0.7	0.8	…	0.9	0.4

W

x

155
200
…
110
78

b

0.1
0.2

0.3	강아지 점수
0.7	고양이 점수

선형분류기

지난시간 돌아보기 ...

모두의연구소

- #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

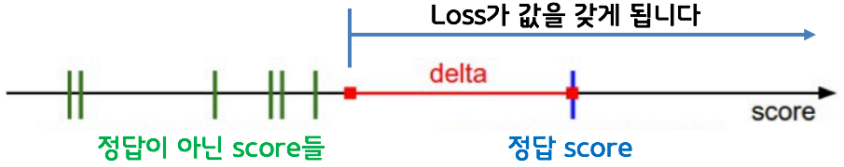
SVM loss

$$L_i = \sum_{j \neq y_i} \max(0, f(x_i, W)_j - f(x_i, W)_{y_i} + \Delta)$$

i 번째 이미지의 Loss

정답이 아닌 것들만 합한다

정답이 아닌 score와 정답 score와의 차이



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

지난시간 돌아보기 ...

모두의연구소

- #lec_deeplearning

Softmax loss

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

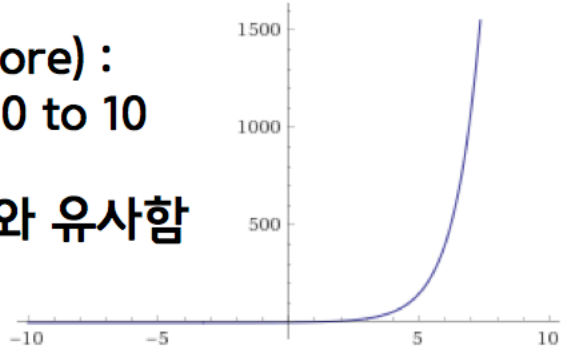
Softmax

강아지 점수	2.3	exp	9.97
고양이 점수	-1.2		0.3

전부다 양수가 됨

exp(score) :
from -10 to 10

Max함수와 유사함



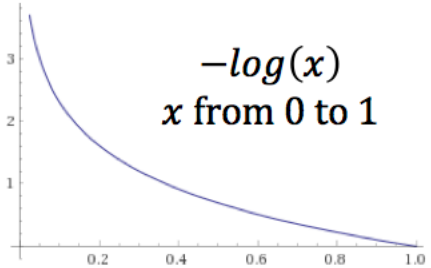
지난시간 돌아보기 ...

모두의연구소

- #lec_deeplearning

Softmax loss

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



강아지 점수	2.3	exp →	9.97	normalize →	0.97
고양이 점수	-1.2		0.3		**0.03**
			전부다 양수가 됨		**확률**

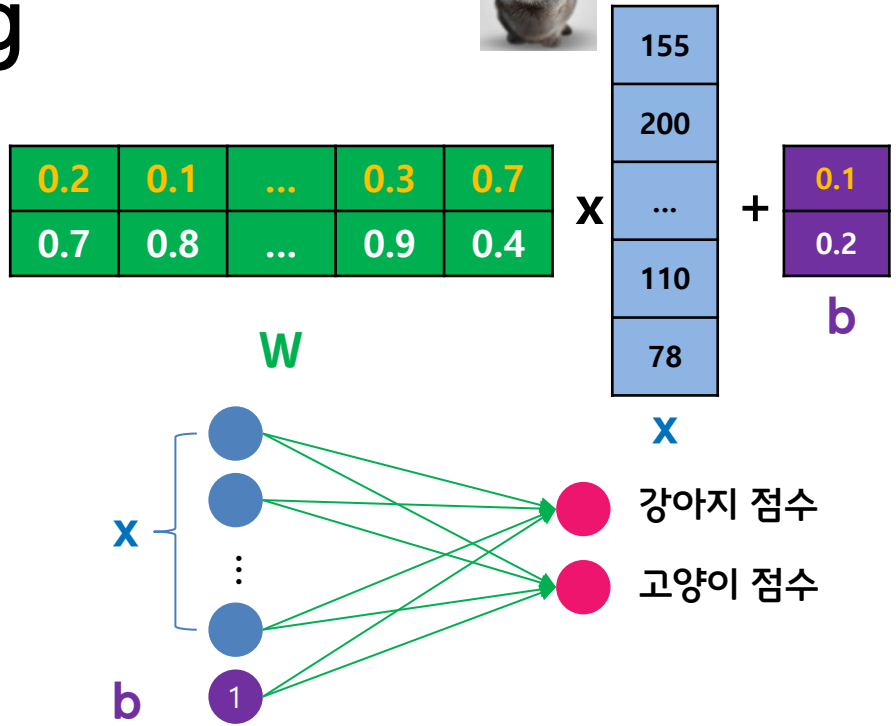
$L_i = -log(0.03) = 3.5$

지난시간 돌아보기 ...

모두의연구소

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

고양이가 입력이면 고양이 점수가 높아야 함



지난시간 돌아보기 ...

모두의연구소

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

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

W

x

155
200
...
110
78

x

+

0.1
0.2

b

=

Cross-entropy loss (Softmax)

강아지 점수	2.3	exp	9.97	normalize	0.97	강아지 확률
고양이 점수	-1.2		0.3		0.03	고양이 확률

$-log(0.03) = 3.5$

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

지난시간 돌아보기 …

모두의연구소

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

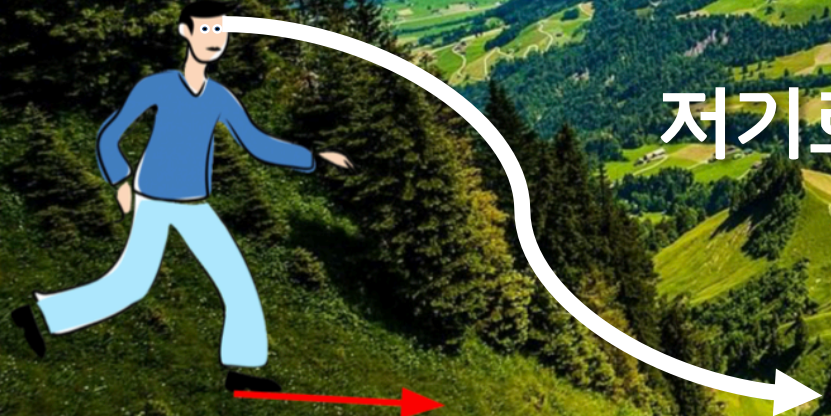
오늘의 주제

Loss를 최소화하는 w와 b를 찾아라

최저점을 향해 가시오

저기 보이네~~~!!!!

저기로 가자~~!

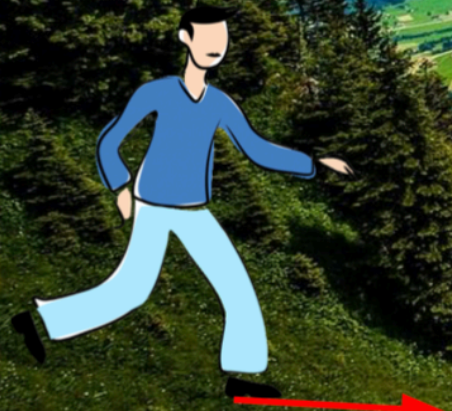


From : lecture note 2 - cs231n stanford university

최저점을 향해 가시오

헉?!

눈이 없으면 ?



최저점을 향해 가시오

넘어질 것 같아..
어디까지 갈 수 있을까..

[대안]
발로 더듬더듬 해서 내리막이면 가자!

내리막을 찾는 방법

모두의연구소

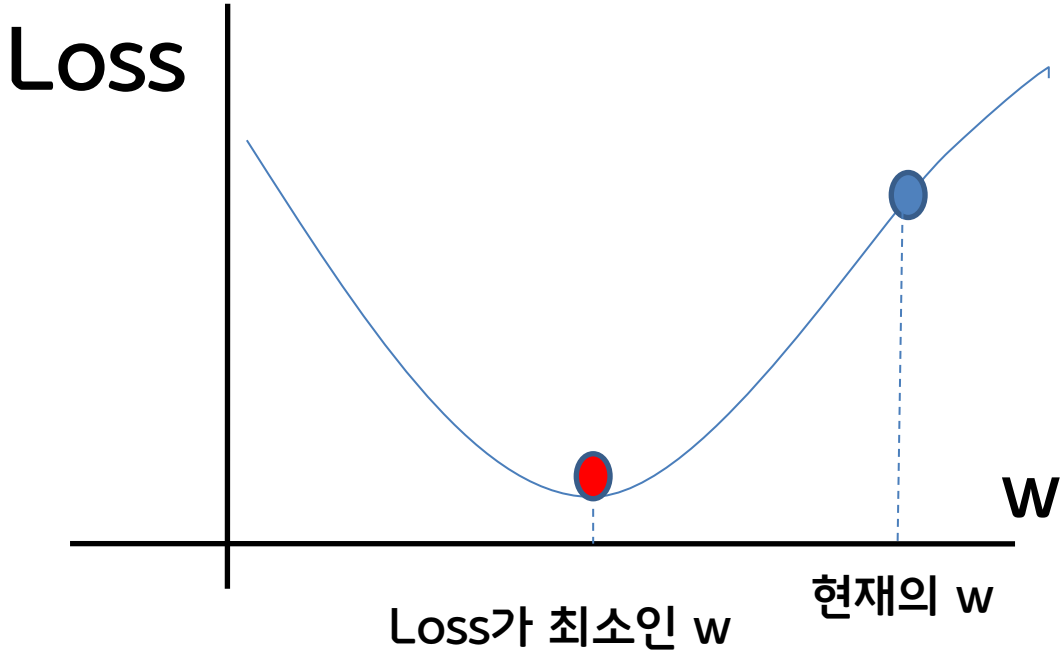
내리막 ?? == 기울기 ??

기울기를 따라 내려 가보자

Gradient Descent

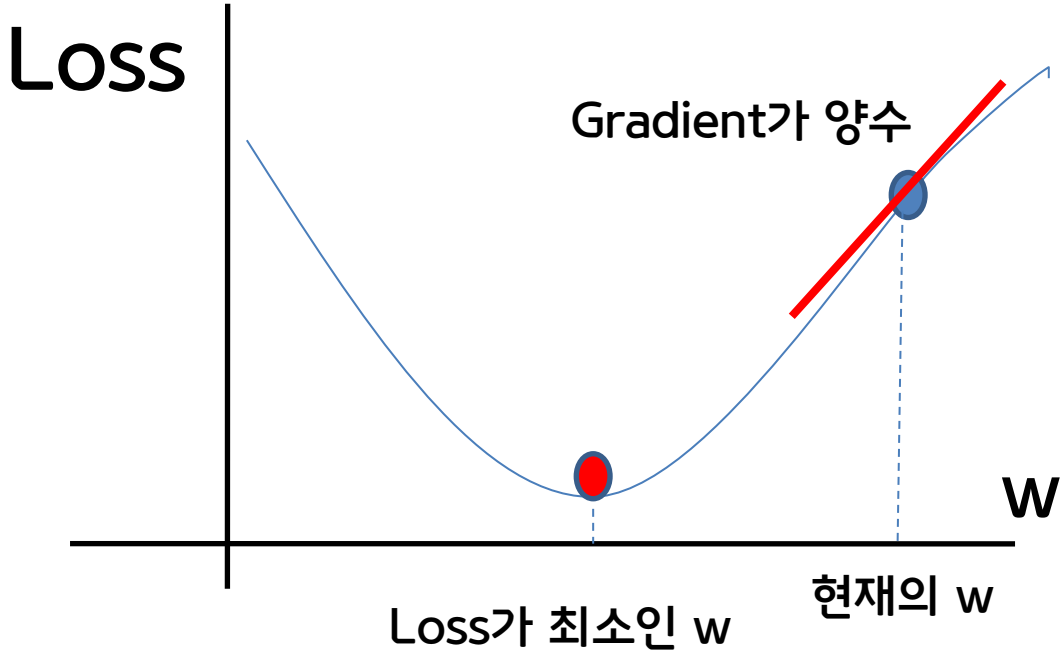
Gradient Descent

모두의연구소



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

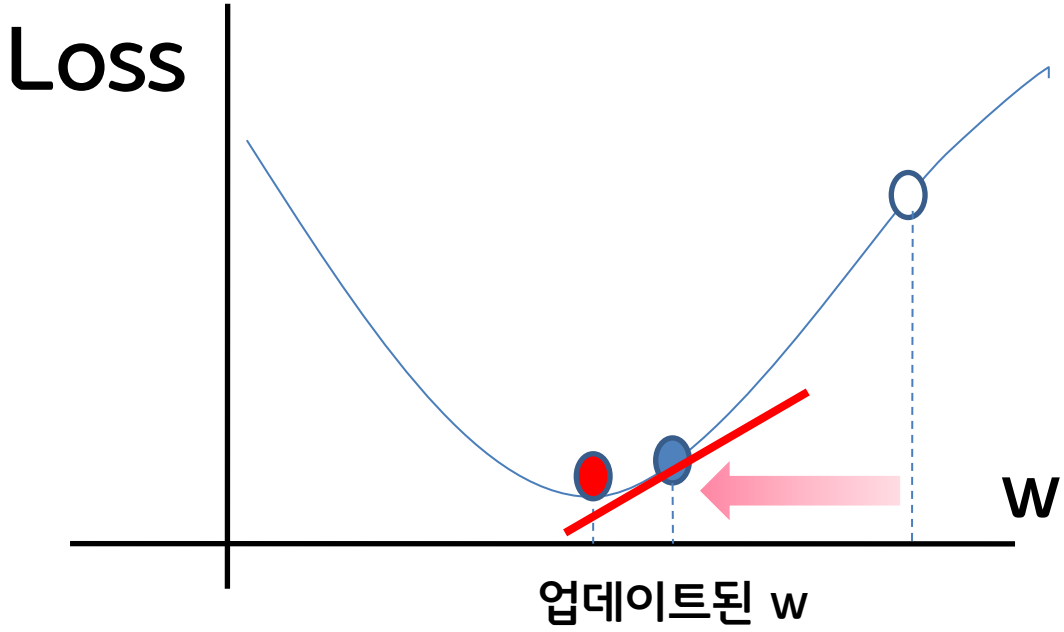
Gradient Descent



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

η : learning rate

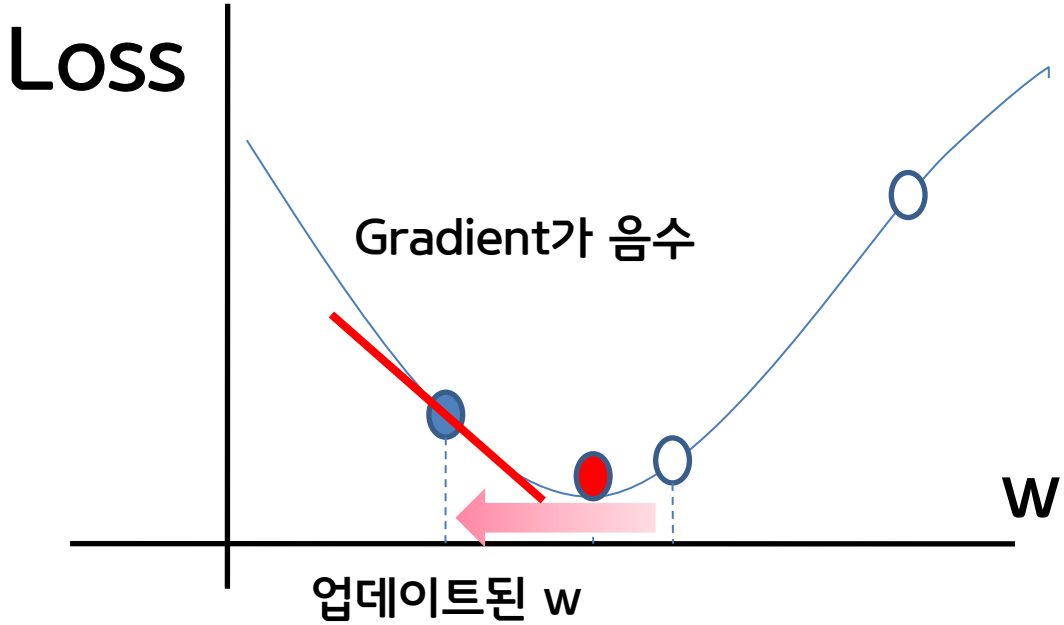
Gradient Descent



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

η : learning rate

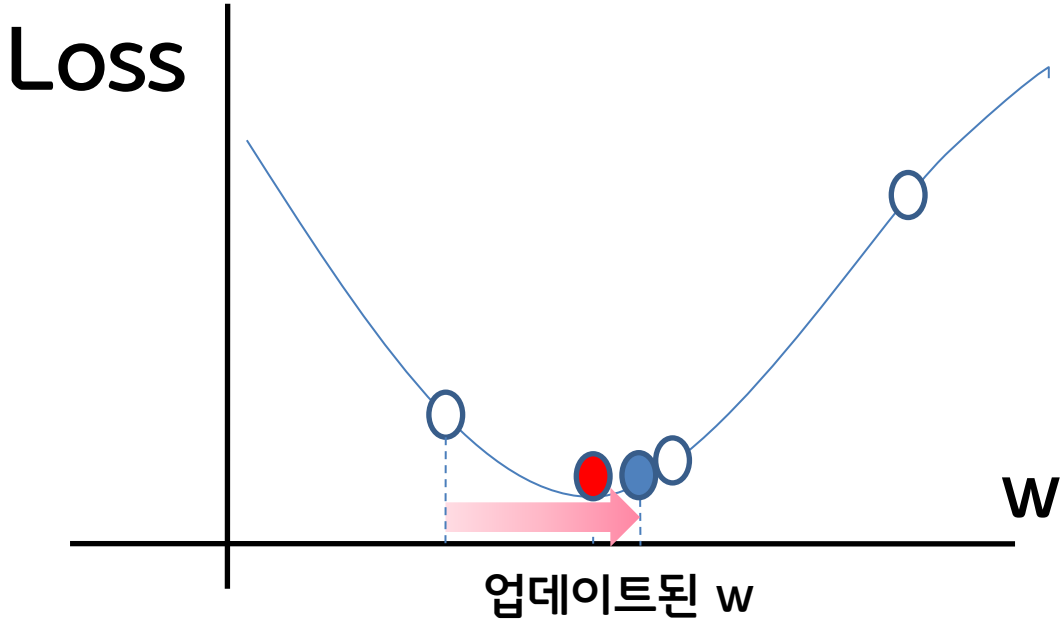
Gradient Descent



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

η : learning rate

Gradient Descent



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

Loss 함수에 대한 w의 음의 Gradient 찾아서 연속적으 업데이트해 주면 되는군요

그런데 어떻게 Gradient를 찾죠?

Gradient Descent

모두의연구소

Strategy #2: **Follow the slope**

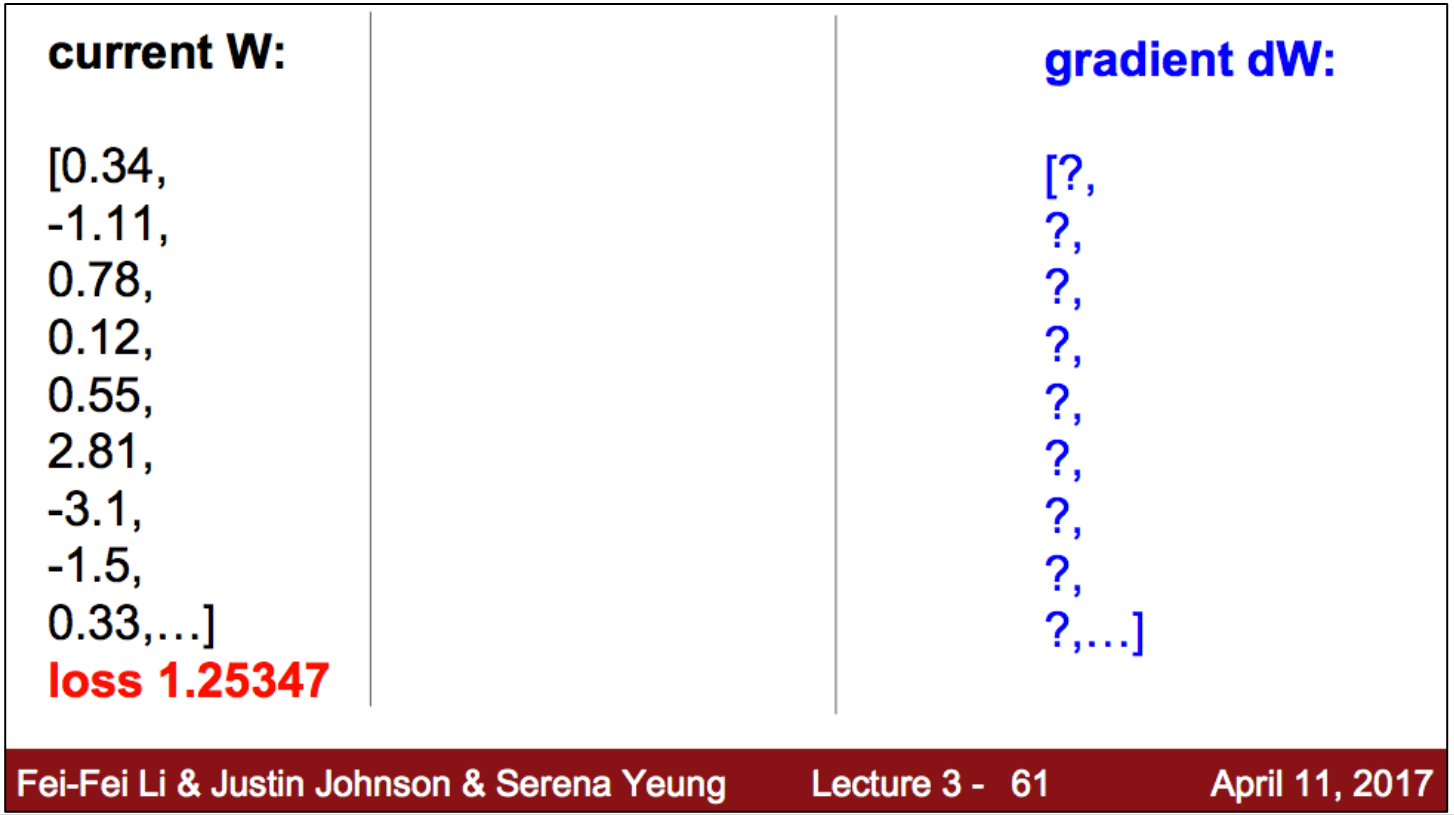
In 1-dimension, the derivative of a function:

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

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



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**,
?,
?,

(1.25322 - 1.25347)/0.0001
= -2.5

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

?,
?,…]

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:	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, **0.6**, ?, ?, ?,…]
loss 1.25347	loss 1.25353	

(1.25353 - 1.25347)/0.0001
= 0.6

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

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:	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, **0**, ?, ?,…]
loss 1.25347	loss 1.25347	

(1.25347 - 1.25347)/0.0001
= 0

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

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:

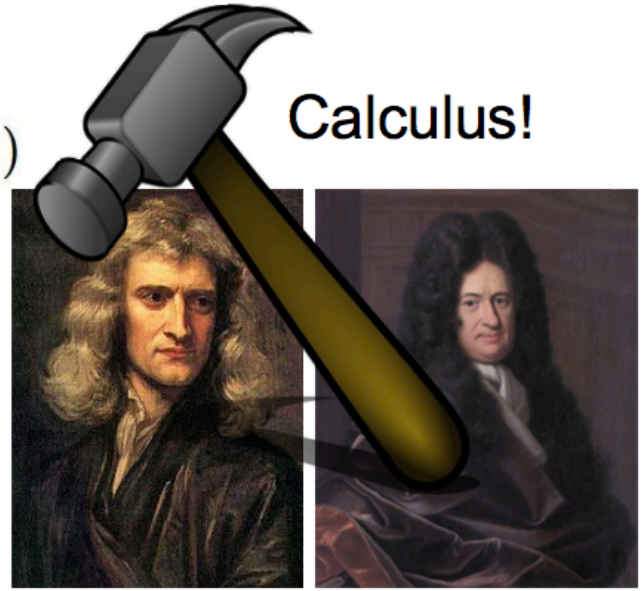
$$L = \frac{1}{N}\sum_{i=1}^{N} L_i + \sum_k W_k^2$$

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

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

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

모두의연구소

수치 미분은 구현은 쉽지만 시간이 오래걸립니다

또한 해석적 방법에 비해 부정확합니다

오차역전파법 (Backpropagation)

Computational Graph

모두의연구소

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

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

z의 x에 대한 미분의 연쇄법칙 표현

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t}\frac{\partial t}{\partial x} \Rightarrow \frac{\partial z}{\partial x} = \frac{\partial z}{\partial t}\frac{\partial t}{\partial x}$$

Reference : "밑바닥부터 시작하는 딥러닝", 사이토 고키

Computational Graph

모두의연구소

- **연쇄법칙이란?**

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

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t}\frac{\partial t}{\partial x} \Rightarrow \begin{matrix} \frac{\partial z}{\partial t} = 2t \\ \frac{\partial t}{\partial x} = 1 \end{matrix}$$

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

Reference : "밑바닥부터 시작하는 딥러닝", 사이토 고키

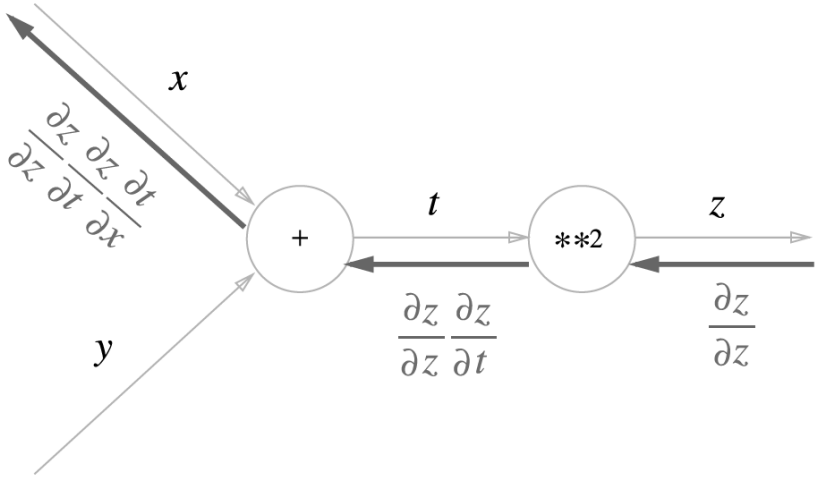
Computational Graph

모두의연구소

- **연쇄법칙과 계산 그래프**

  역전파 과정

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



Reference : "밑바닥부터 시작하는 딥러닝", 사이토 고키

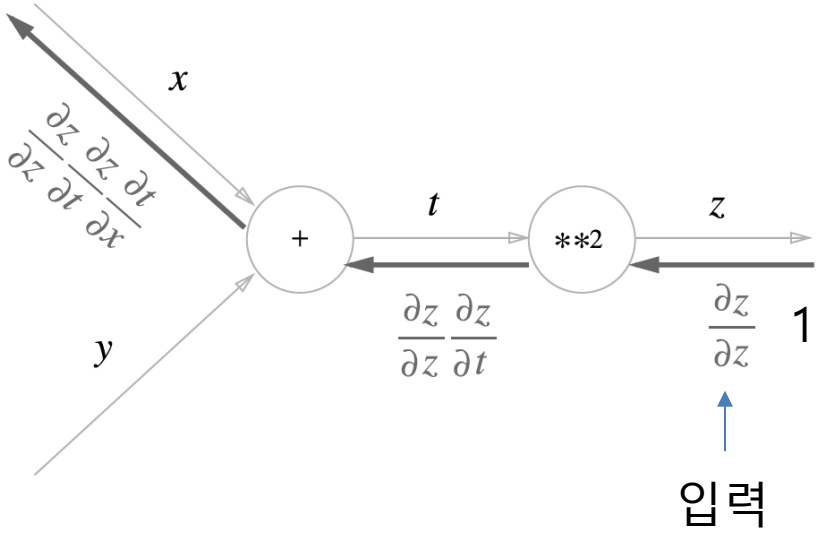
Computational Graph

모두의연구소

- **연쇄법칙과 계산 그래프**

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

역전파 과정



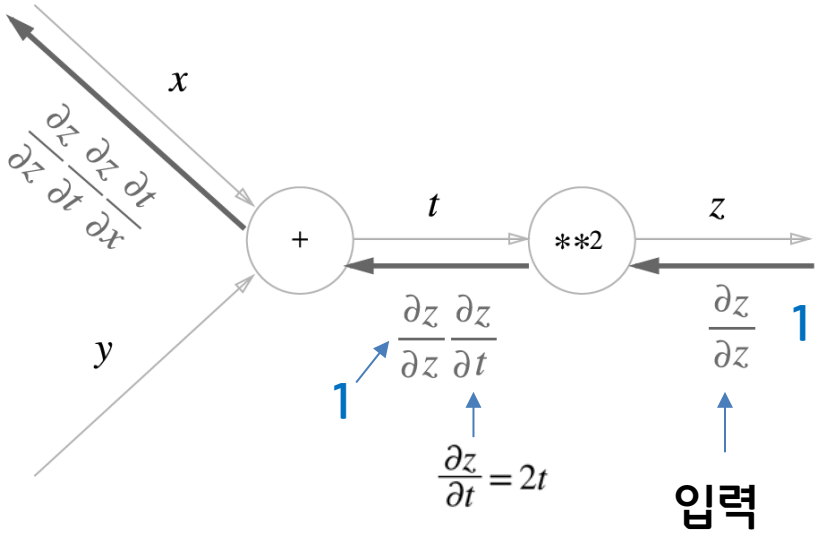
Computational Graph

모두의연구소

- **연쇄법칙과 계산 그래프**

  역전파 과정

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



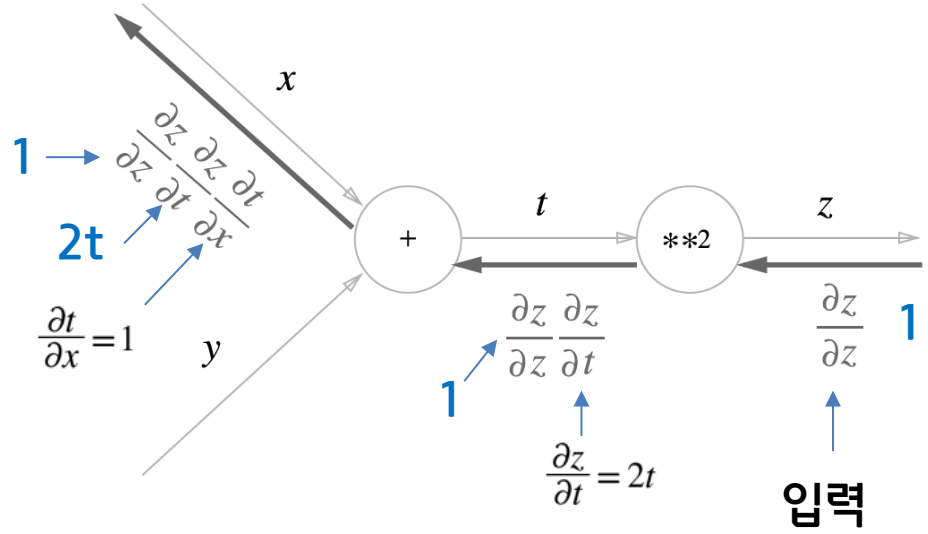
Computational Graph

모두의연구소

- **연쇄법칙과 계산 그래프**

$z = (x + y)^2$ ⇨ $z = t^2$, $t = x + y$

역전파 과정

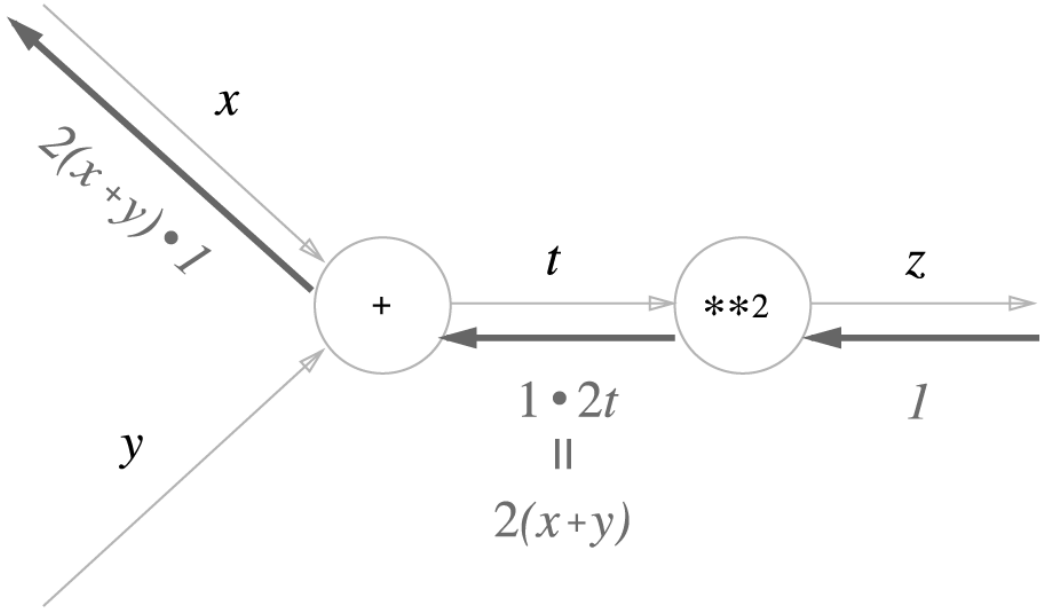


Computational Graph

- **연쇄법칙과 계산 그래프**

$z = (x + y)^2$ ⇨ $z = t^2$, $t = x + y$

역전파 과정



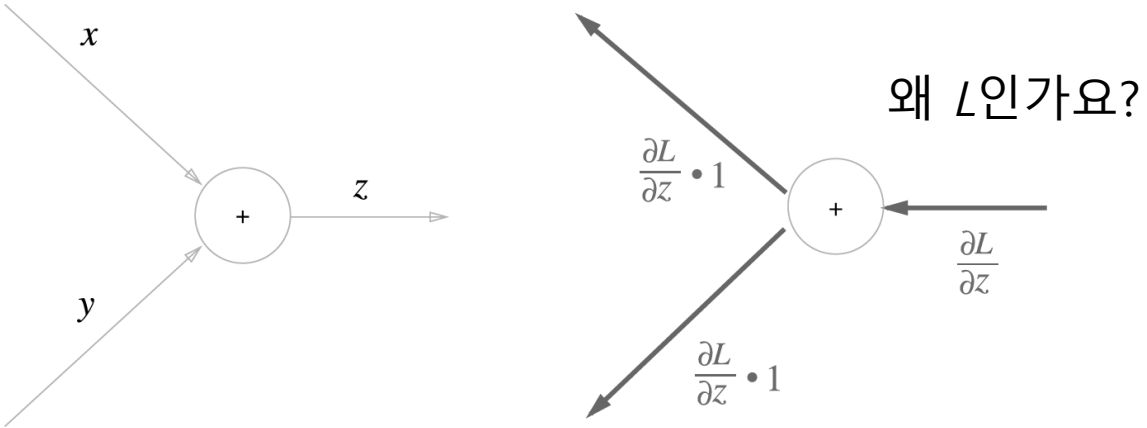
Computational Graph

- **덧셈 노드의 역전파**

$$z = x + y$$

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

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



왜 L인가요?

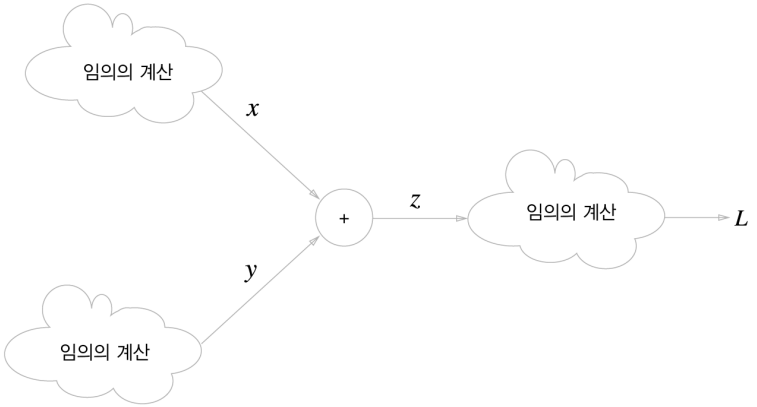
Reference : "밑바닥부터 시작하는 딥러닝", 사이토 고키

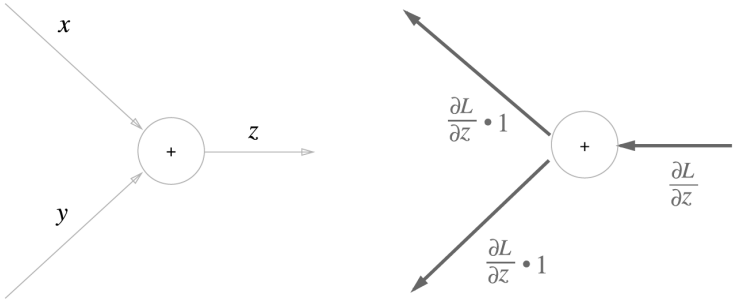
Computational Graph

모두의연구소

- 덧셈 노드의 역전파

실제로 큰 계산을 가정하고 그중 한 노드가 덧셈노드임을 가정한 겁니다





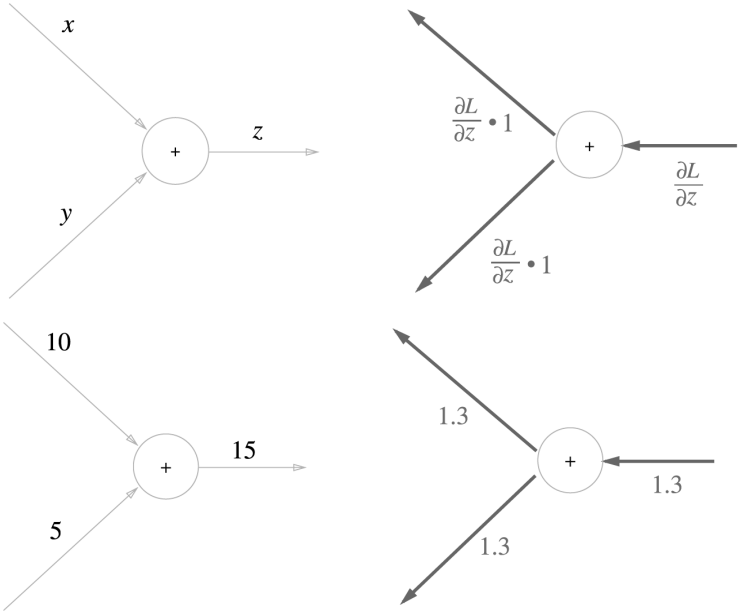
Computational Graph

- **덧셈 노드의 역전파**

$$z = x + y$$

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

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



예시

Computational Graph

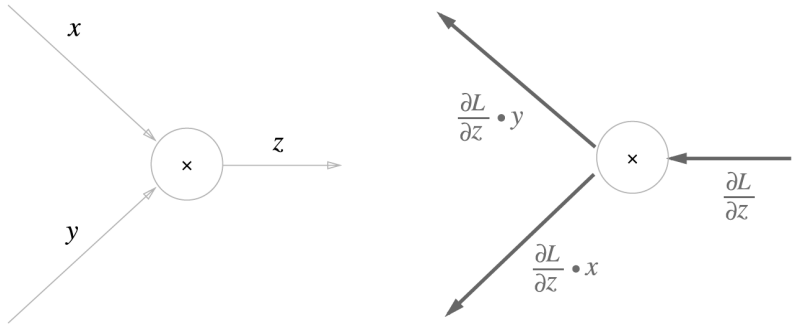
모두의연구소

- **곱셈 노드의 역전파**

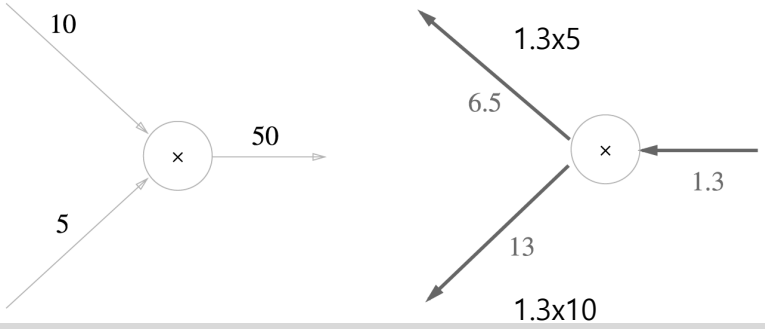
$$z = xy$$

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

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



예시

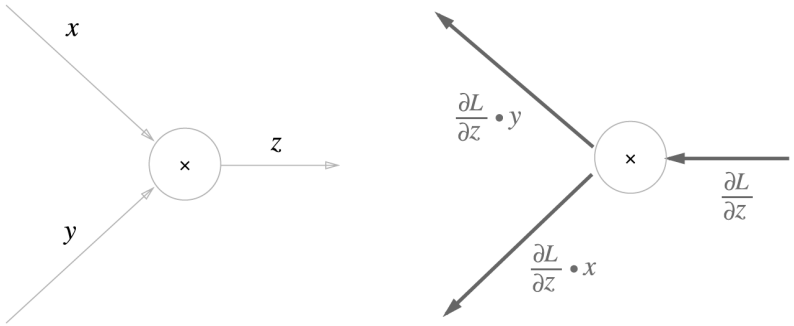


Computational Graph

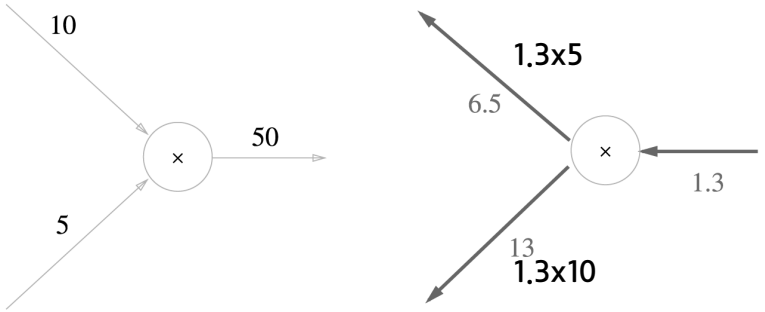
모두의연구소

- 곱셈 노드의 역전파

$$z = xy \Rightarrow \frac{\partial z}{\partial x} = y, \quad \frac{\partial z}{\partial y} = x$$



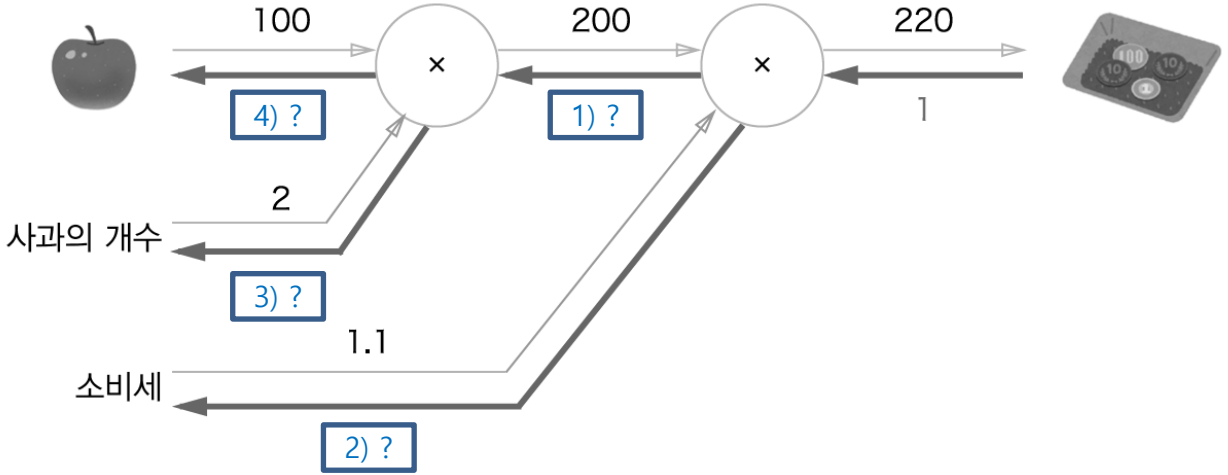
- 곱셈의 역전파는 순방향 입력신호의 값이 필요합니다
- 곱셈노드 구현 시 순전파 입력신호를 변수에 저장해 둡니다



Computational Graph

모두의연구소

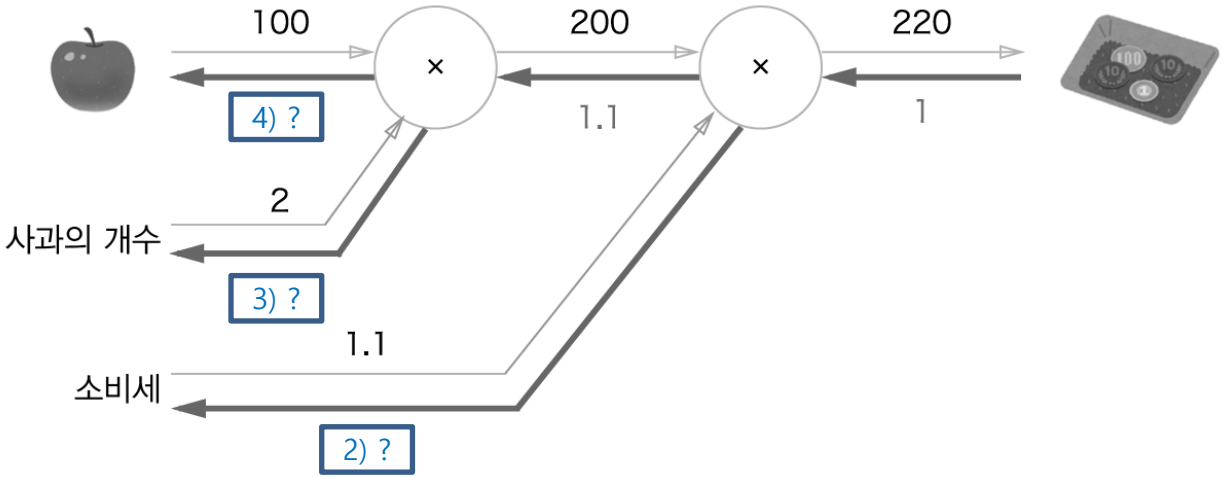
- **사과쇼핑의 예**



Computational Graph

모두의연구소

- **사과쇼핑의 예**

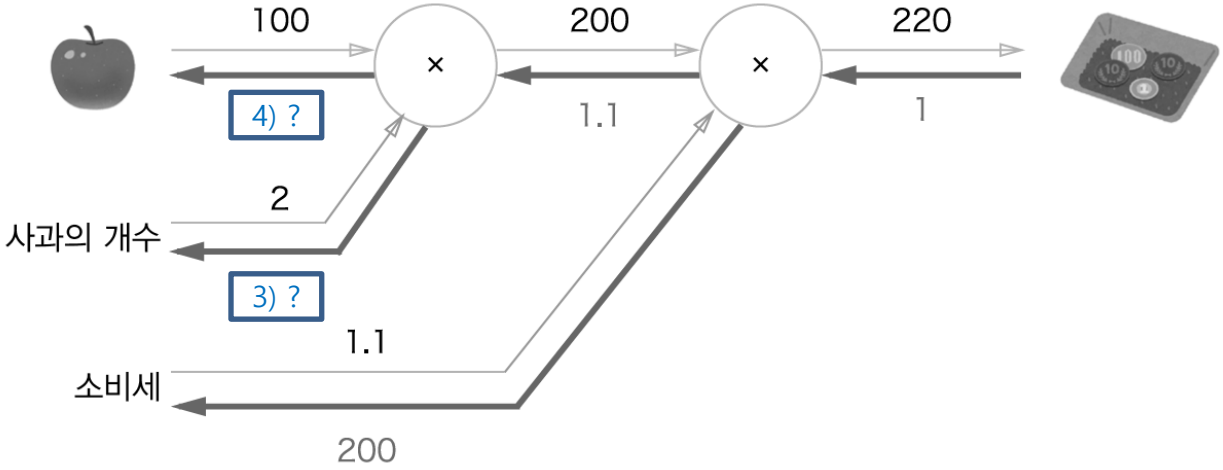


Reference : "밑바닥부터 시작하는 딥러닝", 사이토 고키

Computational Graph

모두의연구소

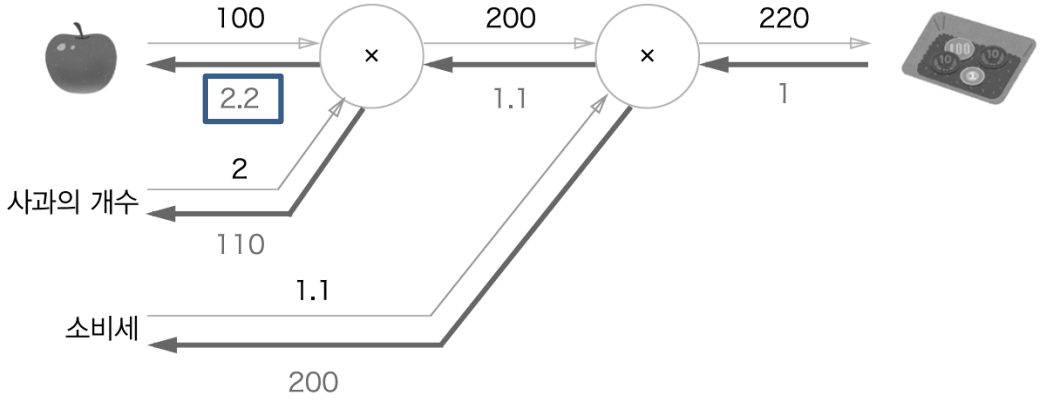
- **사과쇼핑의 예**



Computational Graph

모두의연구소

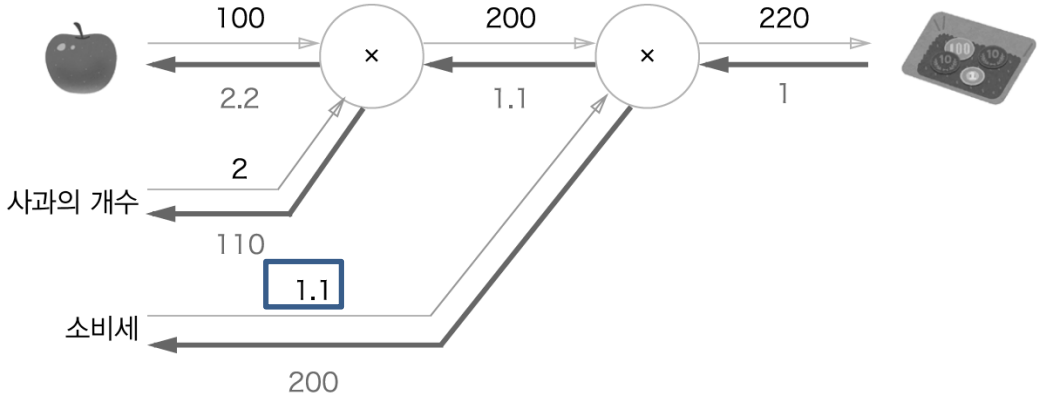
- **사과쇼핑의 예**



- 사과가격이 1원 오르면, 최종 가격 2.2 증가
  - 101(사과 가격)x2x1.1 = 222.2 원

Reference : "밑바닥부터 시작하는 딥러닝", 사이토 고키

Computational Graph



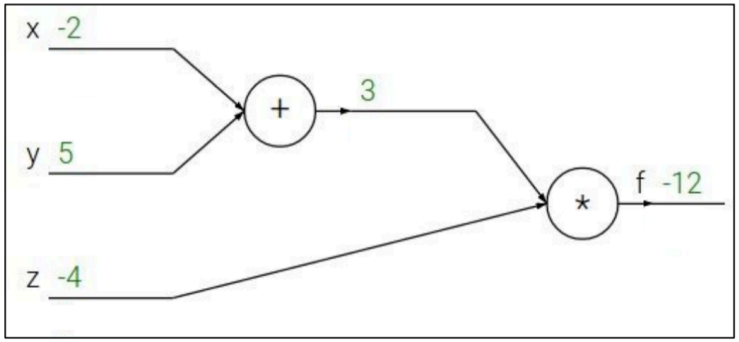
- 사과가격이 1원 오르면, 최종 가격 2.2 증가
  - 101(사과 가격)x2x1.1 = 222.2 원
- 소비세가 100%(1.1+1 = 2.1)오르면 가격 200증가
  - 100x2x2.1(소비세) = 420 원

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

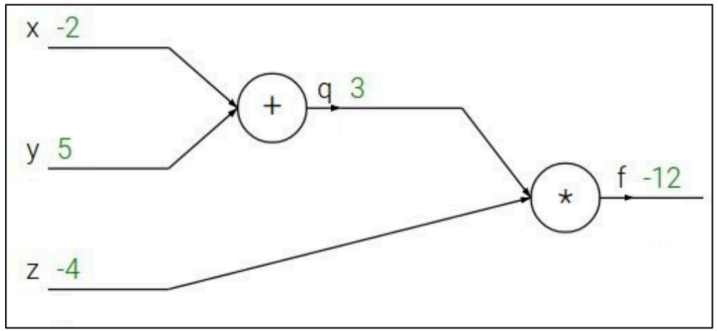
$$f(x, y, z) = (x + y)z$$

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

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

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

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



Computational Graph

Backpropagation: a simple example

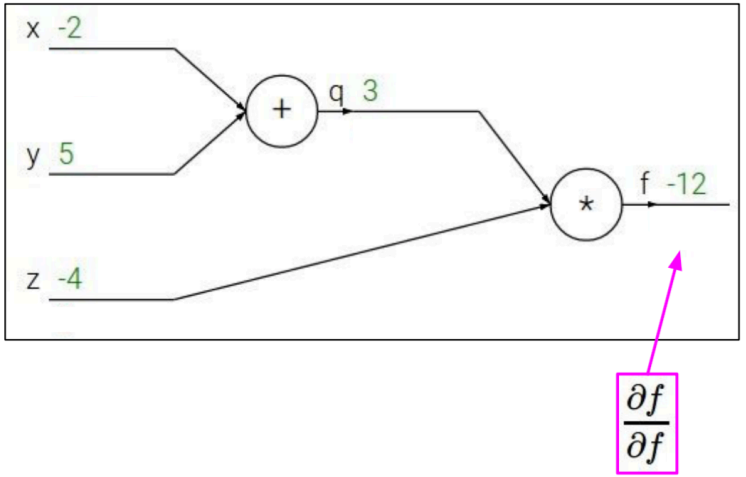
$$f(x, y, z) = (x + y)z$$

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

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

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

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

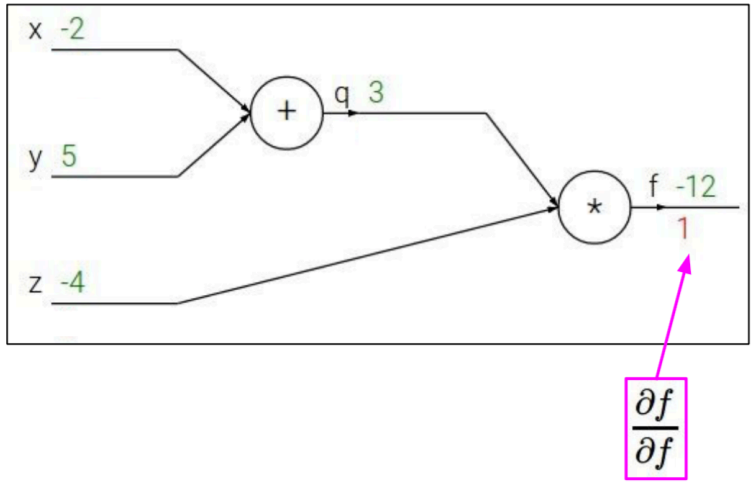


Computational Graph

Backpropagation: a simple example

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

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



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

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

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

Computational Graph

Backpropagation: a simple example

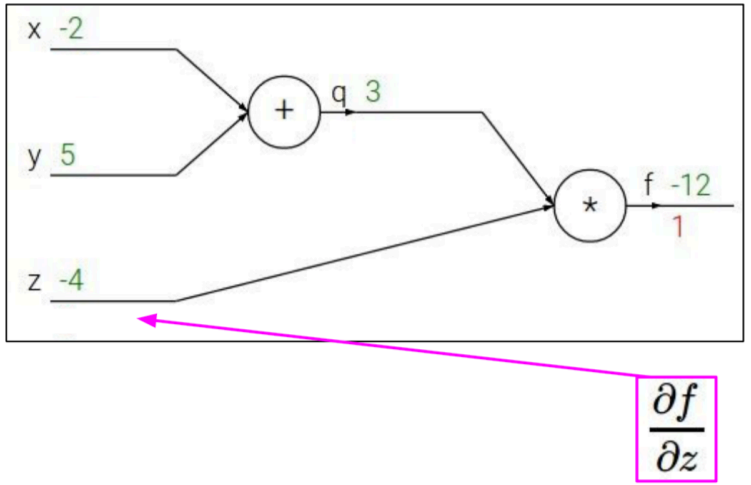
$$f(x, y, z) = (x + y)z$$

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

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

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

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



Computational Graph

Backpropagation: a simple example

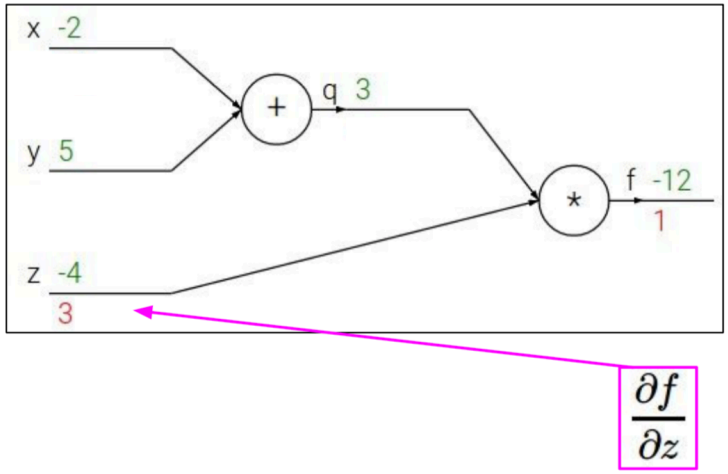
$$f(x, y, z) = (x + y)z$$

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

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

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

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



Computational Graph

Backpropagation: a simple example

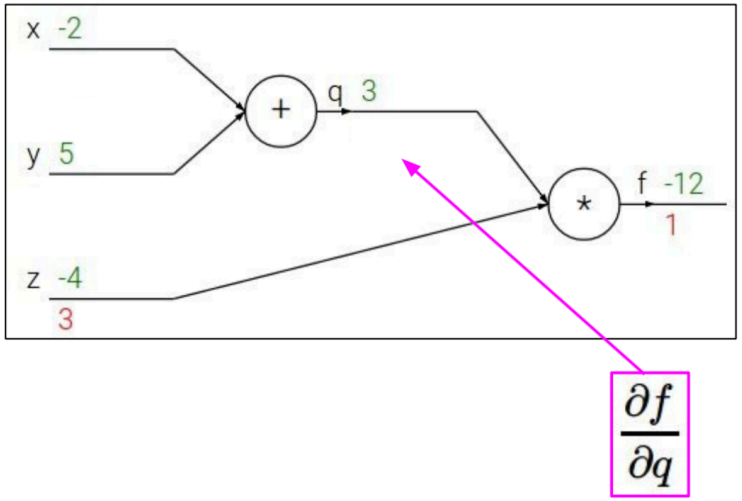
$$f(x,y,z) = (x+y)z$$

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

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

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

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



Computational Graph

Backpropagation: a simple example

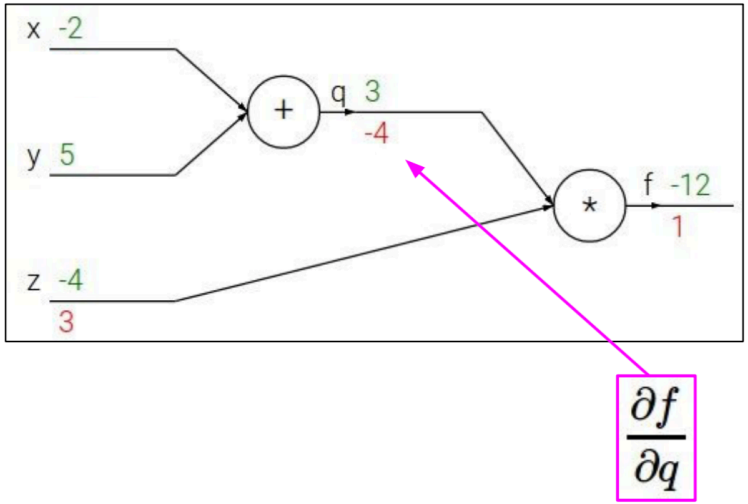
$$f(x, y, z) = (x + y)z$$

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

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

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

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



Computational Graph

Backpropagation: a simple example

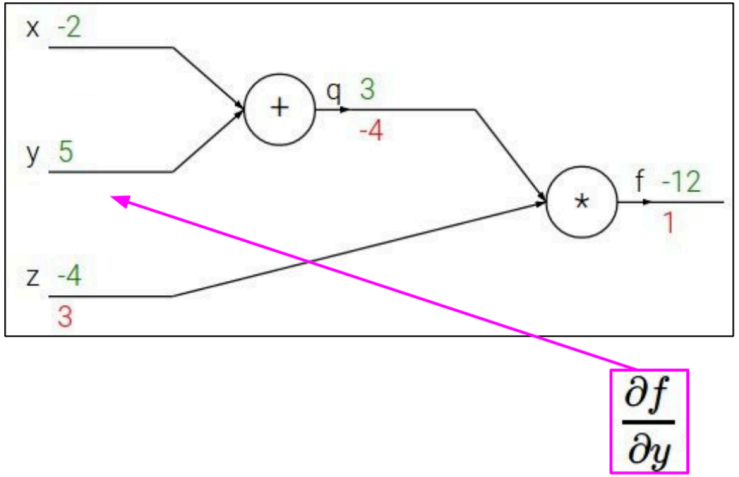
$$f(x, y, z) = (x + y)z$$

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

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

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

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



Computational Graph

Backpropagation: a simple example

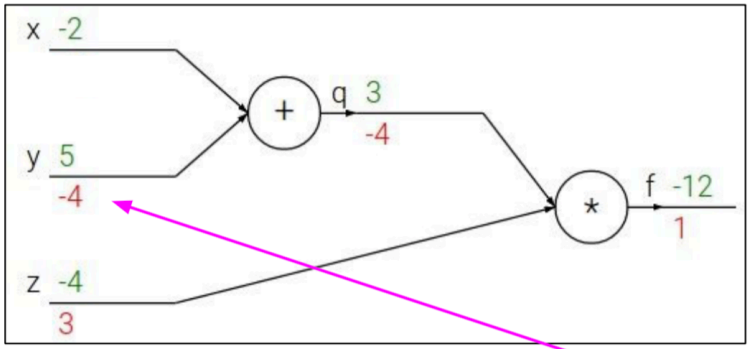
$$f(x, y, z) = (x + y)z$$

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

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

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

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



Chain rule:

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

Computational Graph

Backpropagation: a simple example

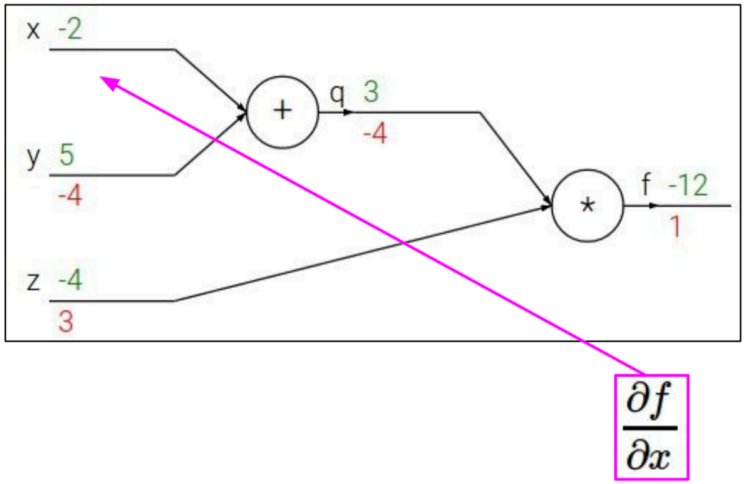
$$f(x, y, z) = (x + y)z$$

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

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

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

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



Computational Graph

Backpropagation: a simple example

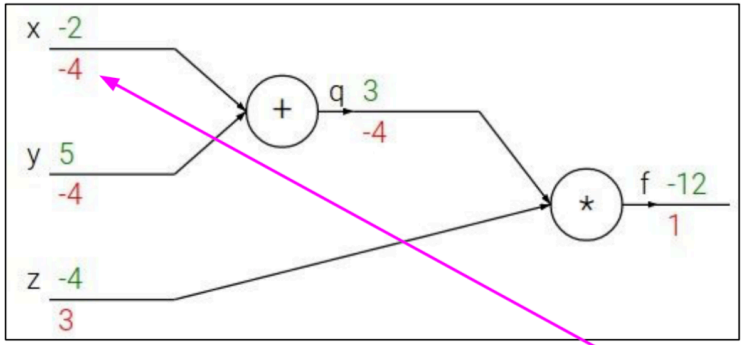
$$f(x,y,z) = (x+y)z$$

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

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

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

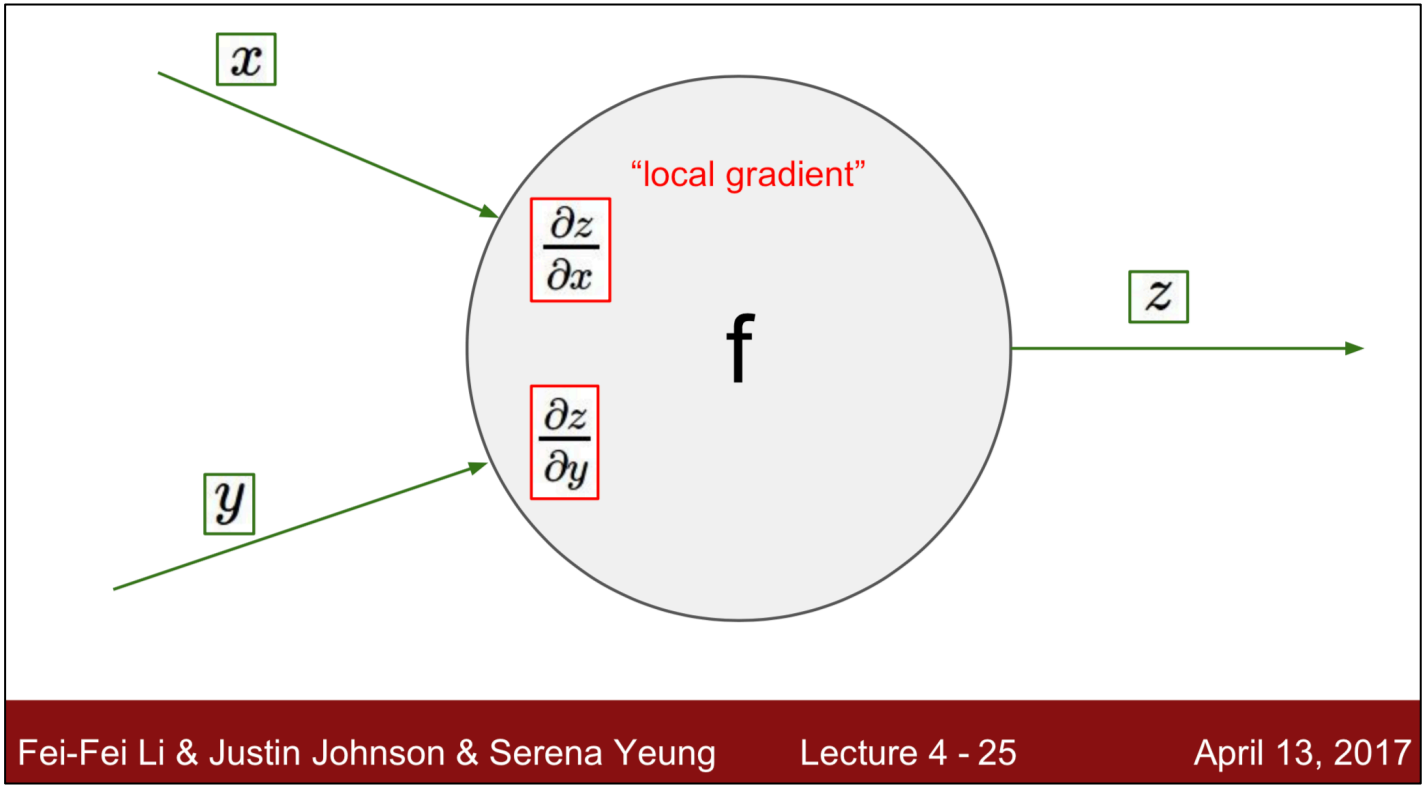
Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



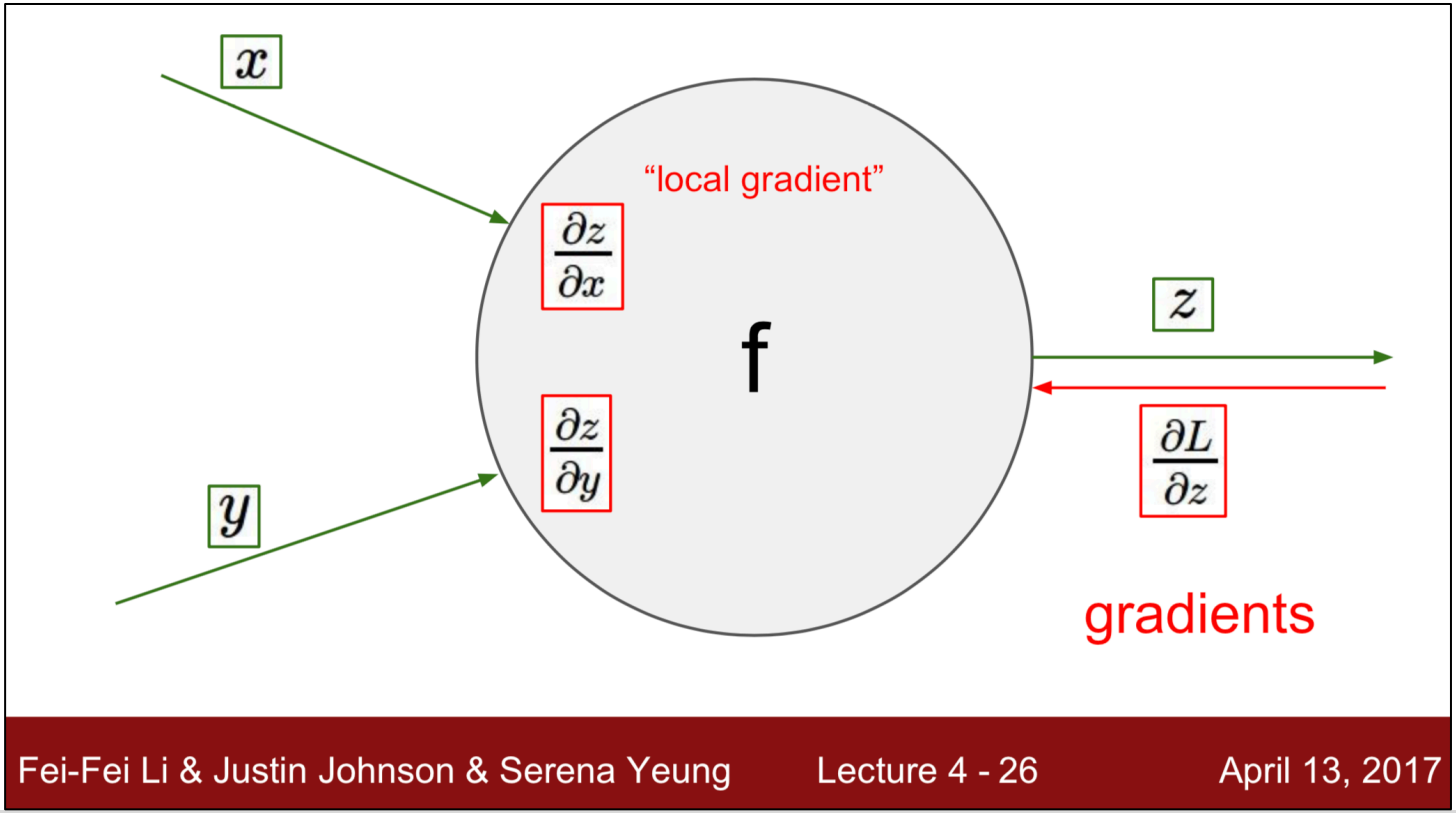
Chain rule:

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

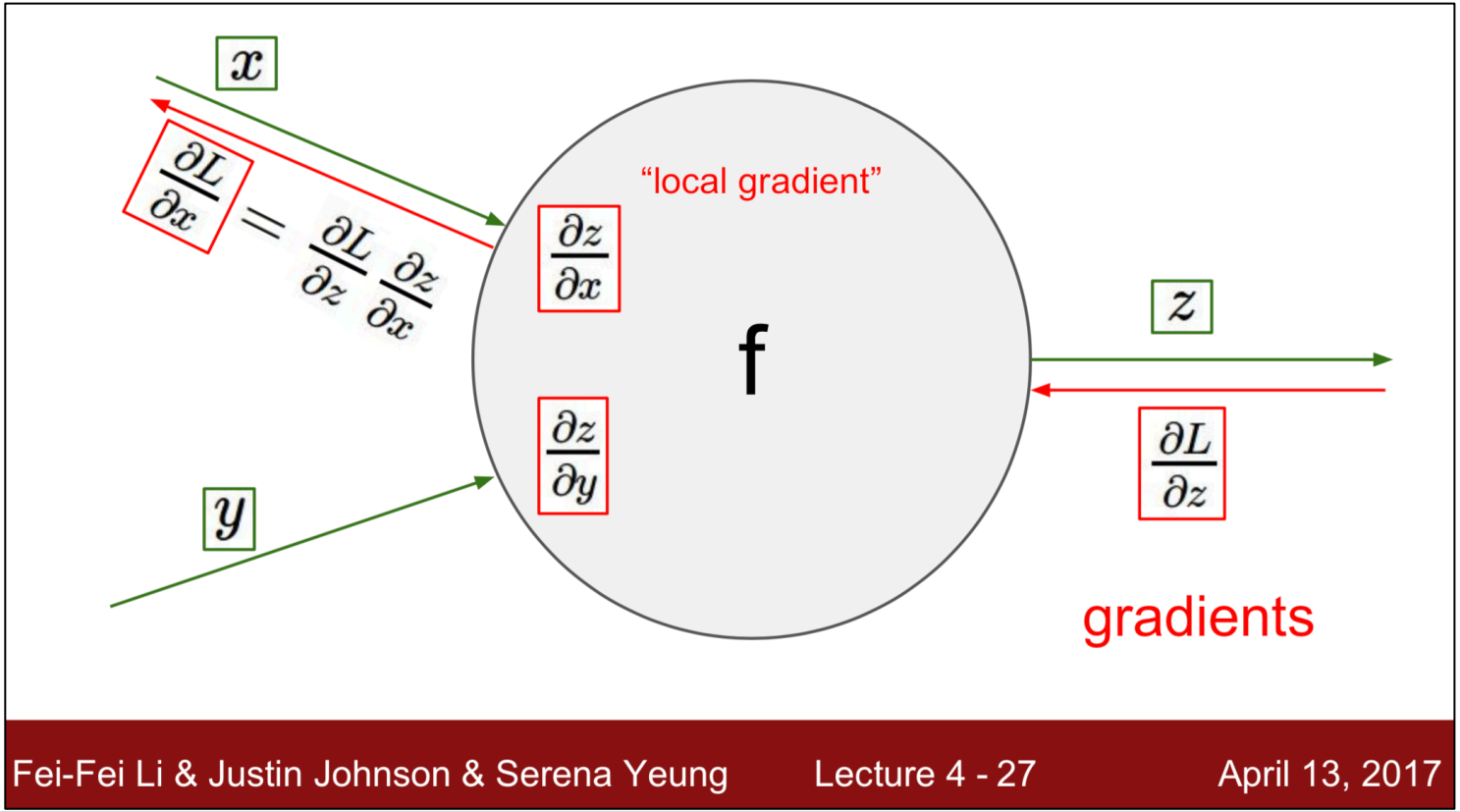
Computational Graph



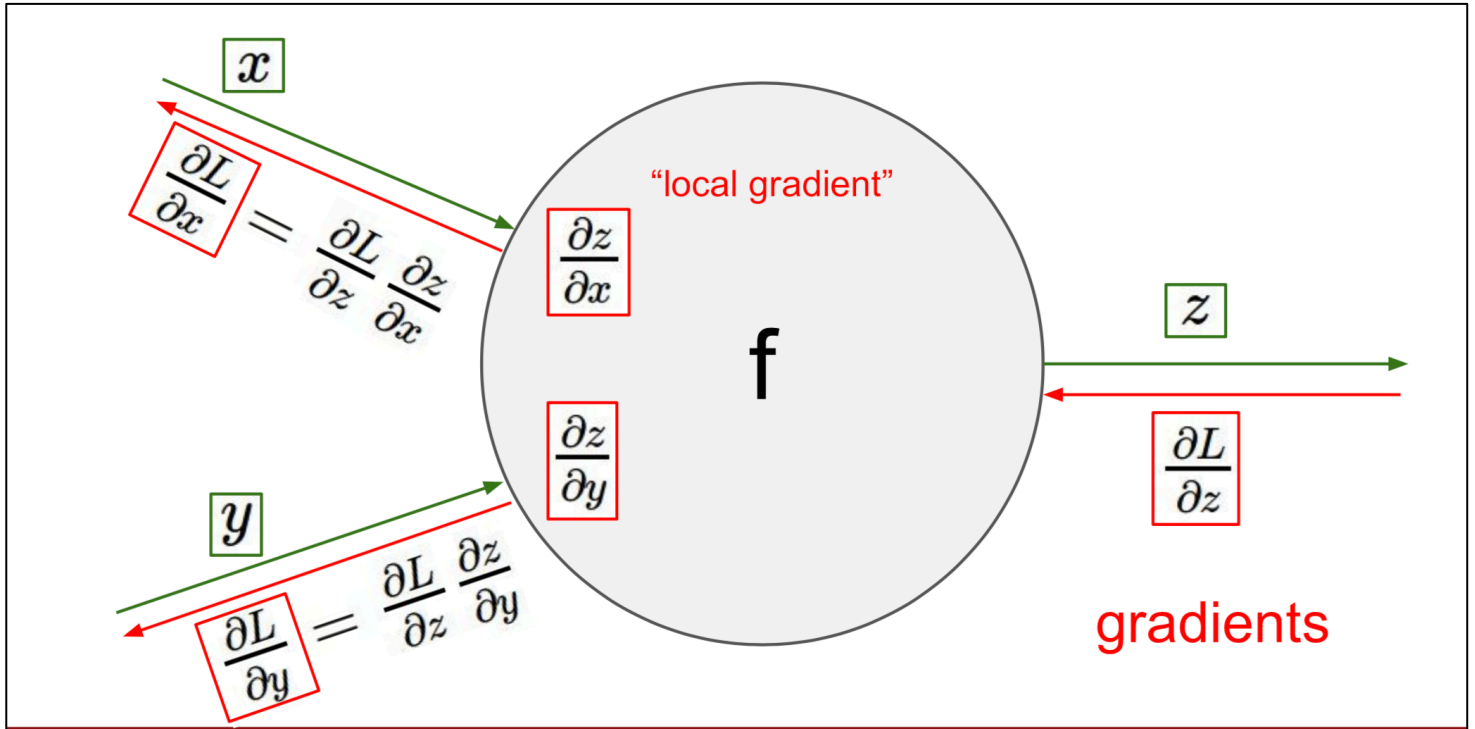
Computational Graph



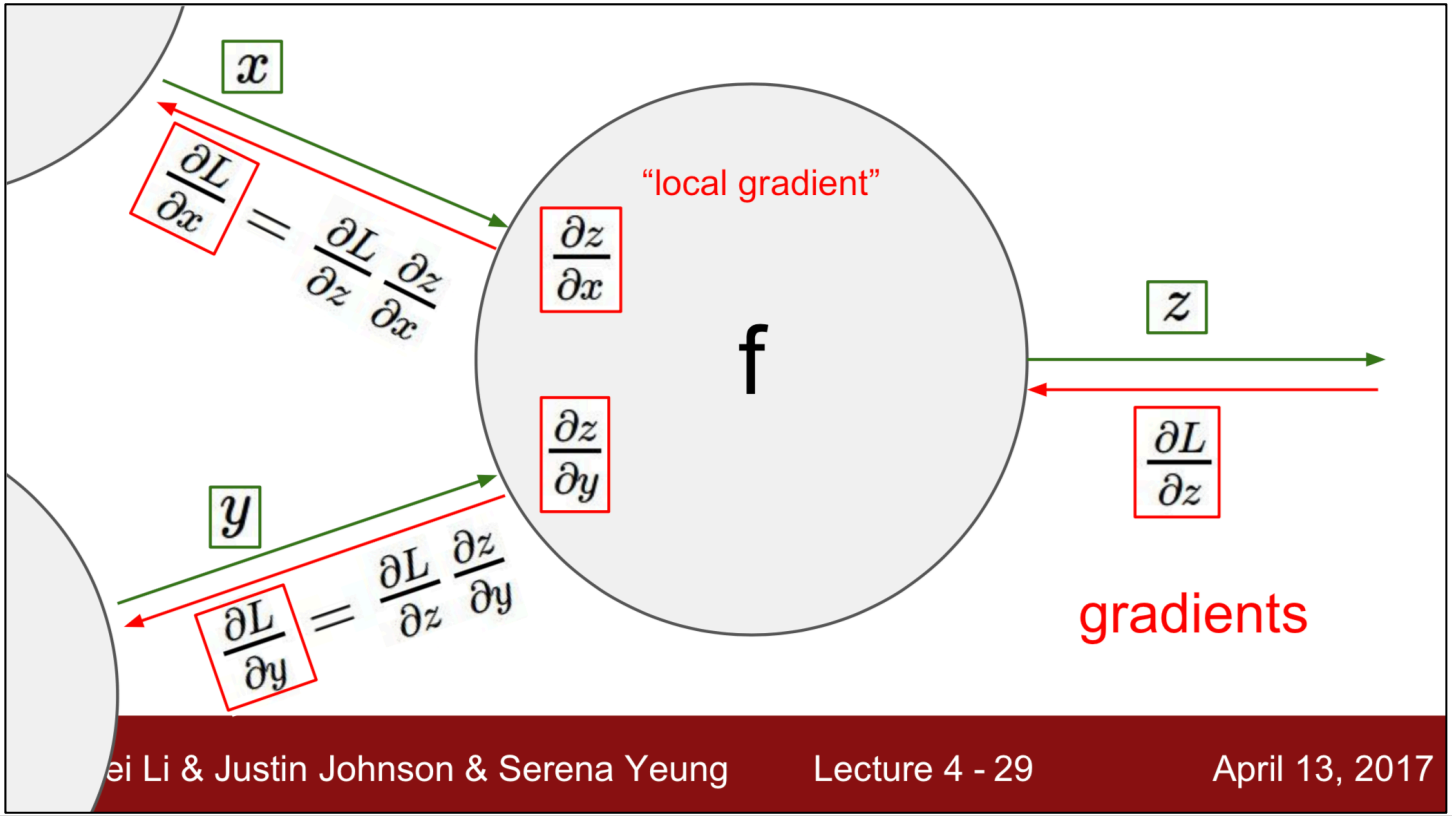
Computational Graph



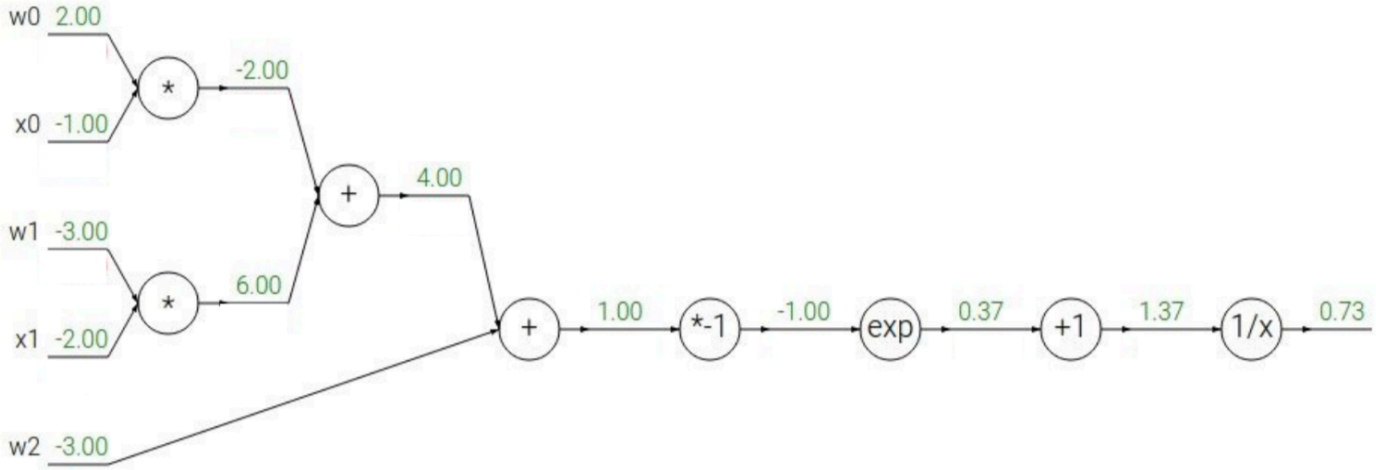
Computational Graph



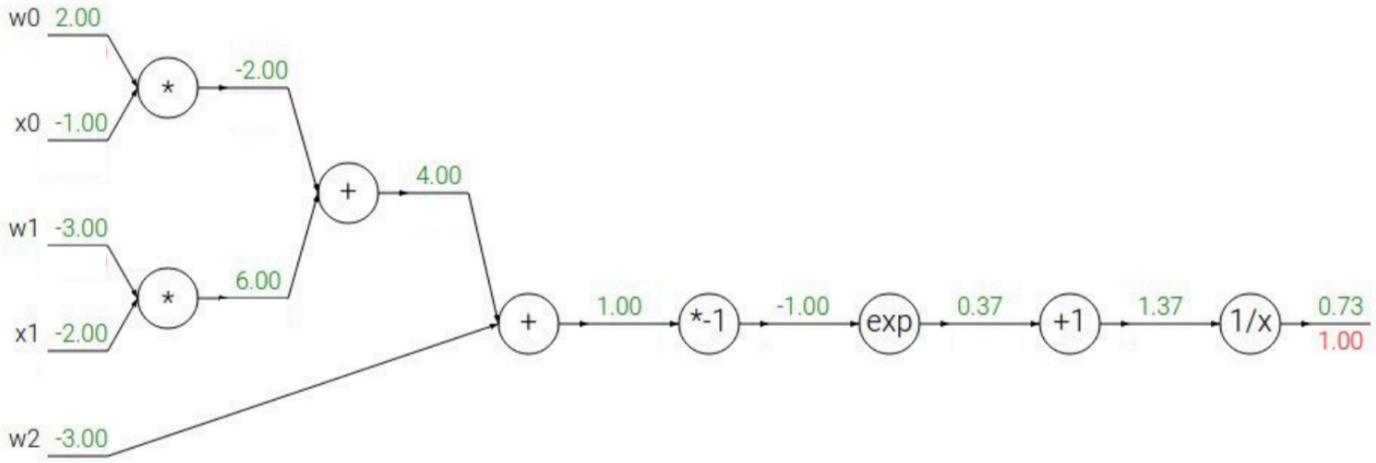
Computational Graph



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



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



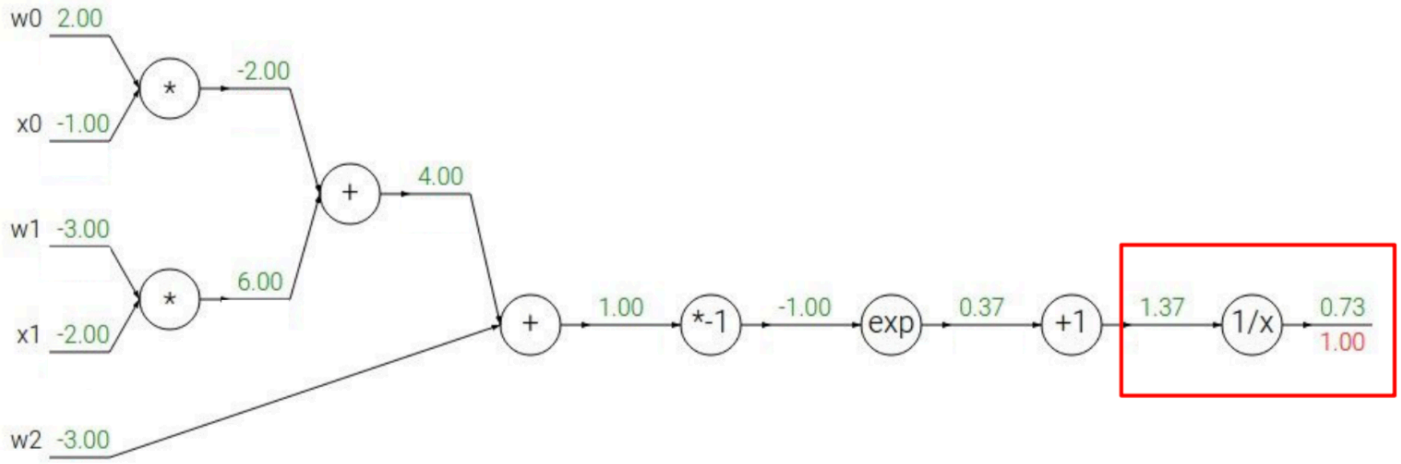
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

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



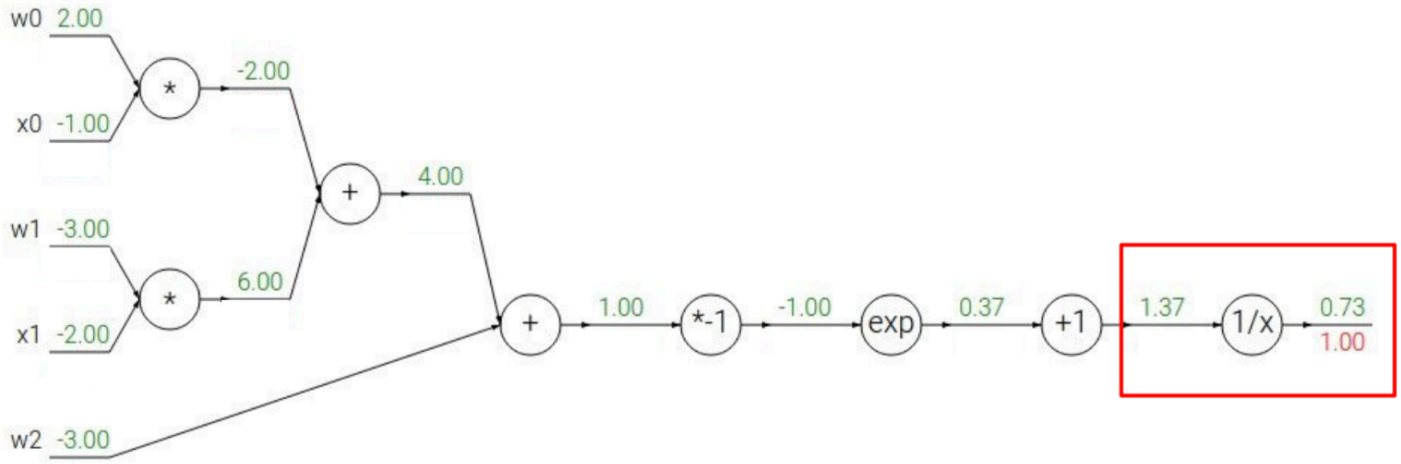
$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$

$f_a(x) = ax \rightarrow \frac{df}{dx} = a$

$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$

$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$

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



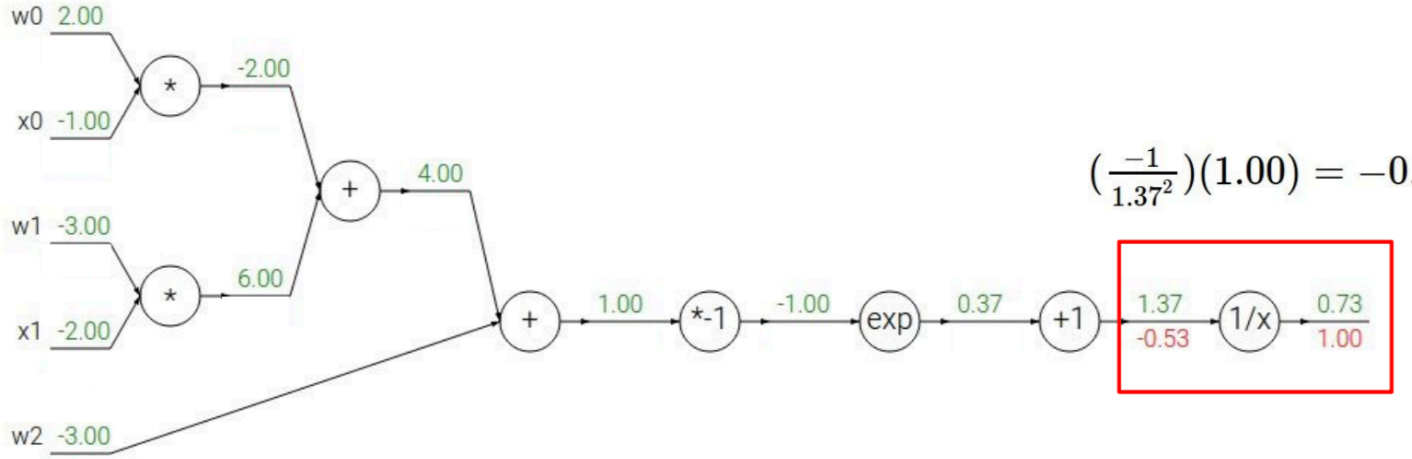
$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$

$f_a(x) = ax \rightarrow \frac{df}{dx} = a$

$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$

$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$

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



$$(\frac{-1}{1.37^2})(1.00) = -0.53$$

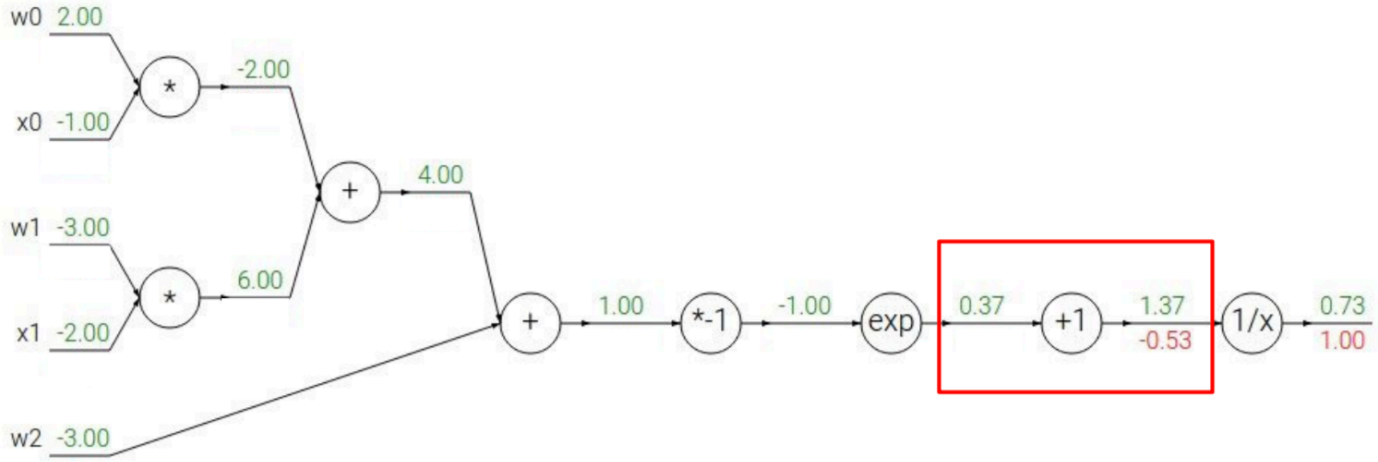
$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$

$f_a(x) = ax \rightarrow \frac{df}{dx} = a$

$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$

$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$

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



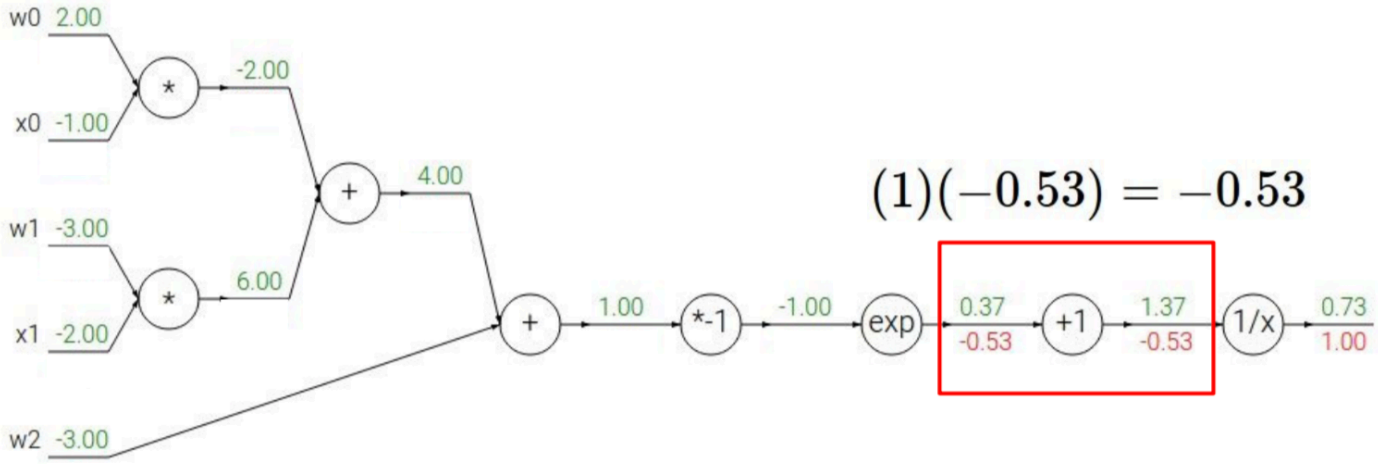
$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$

$f_a(x) = ax \rightarrow \frac{df}{dx} = a$

$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$

$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$

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



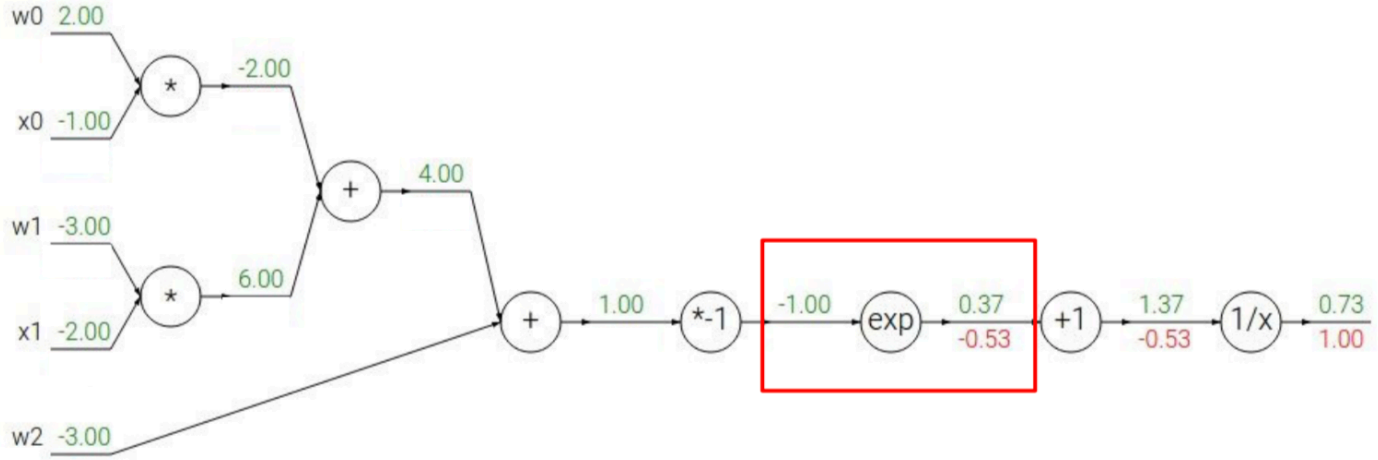
$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$

$f_a(x) = ax \rightarrow \frac{df}{dx} = a$

$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$

$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$

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



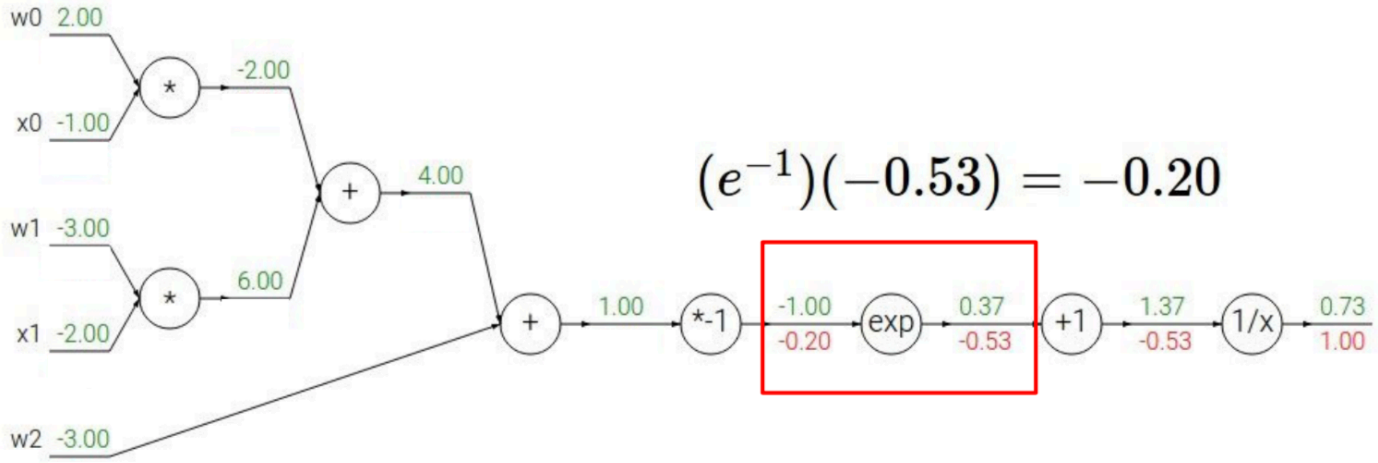
$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$

$f_a(x) = ax \rightarrow \frac{df}{dx} = a$

$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$

$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$

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



$$(e^{-1})(-0.53) = -0.20$$

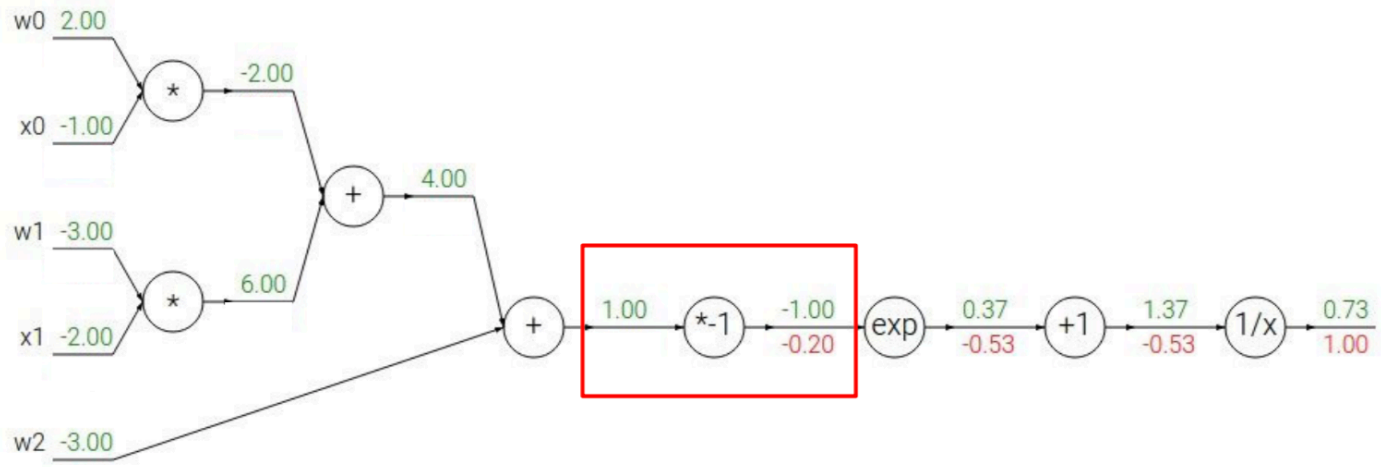
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

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



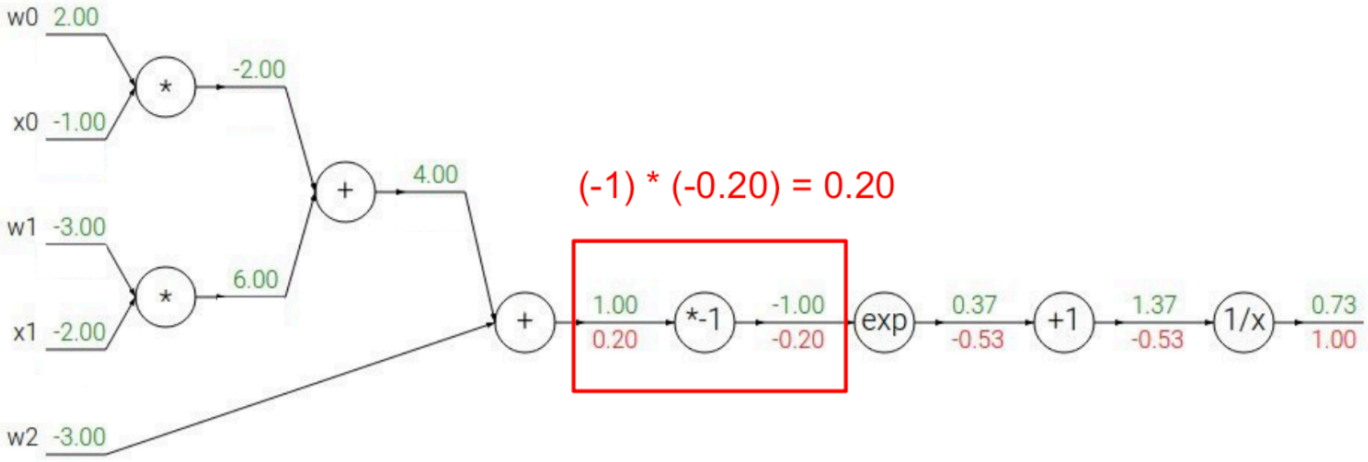
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

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



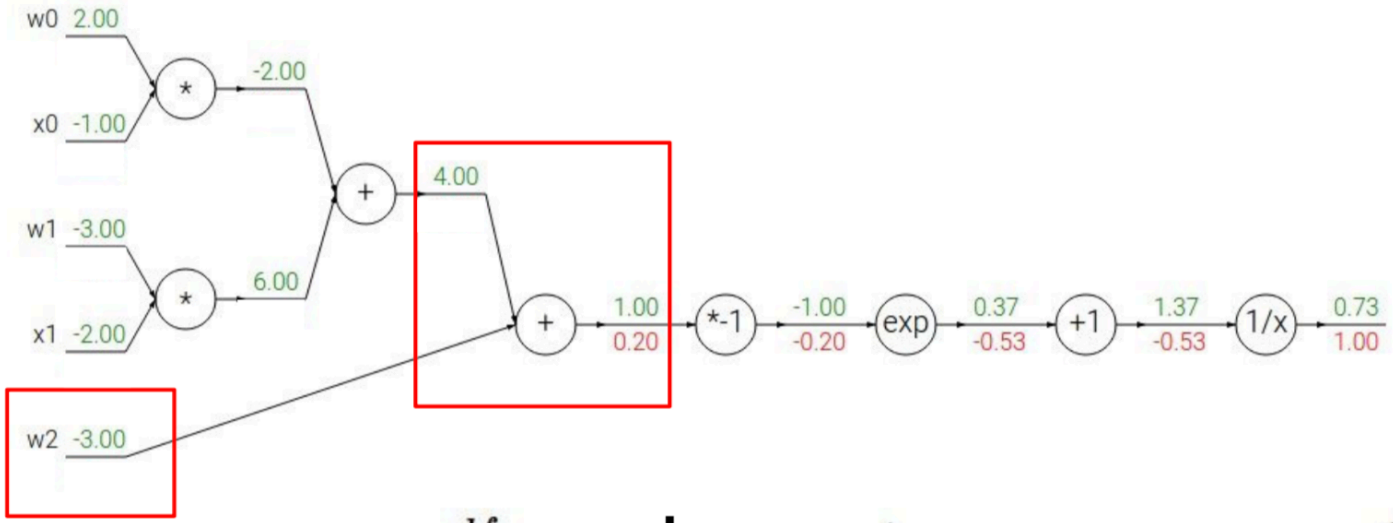
$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$

$f_a(x) = ax \rightarrow \frac{df}{dx} = a$

$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$

$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$

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



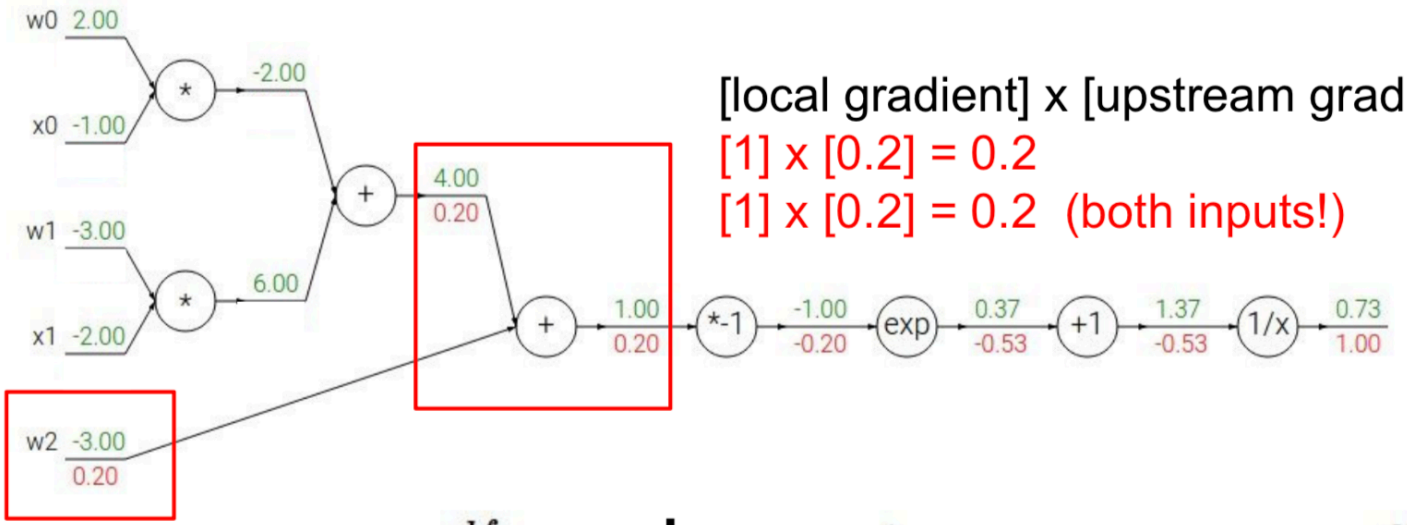
$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$

$f_a(x) = ax \rightarrow \frac{df}{dx} = a$

$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$

$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$

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



[local gradient] x [upstream gradient]

[1] x [0.2] = 0.2

[1] x [0.2] = 0.2 (both inputs!)

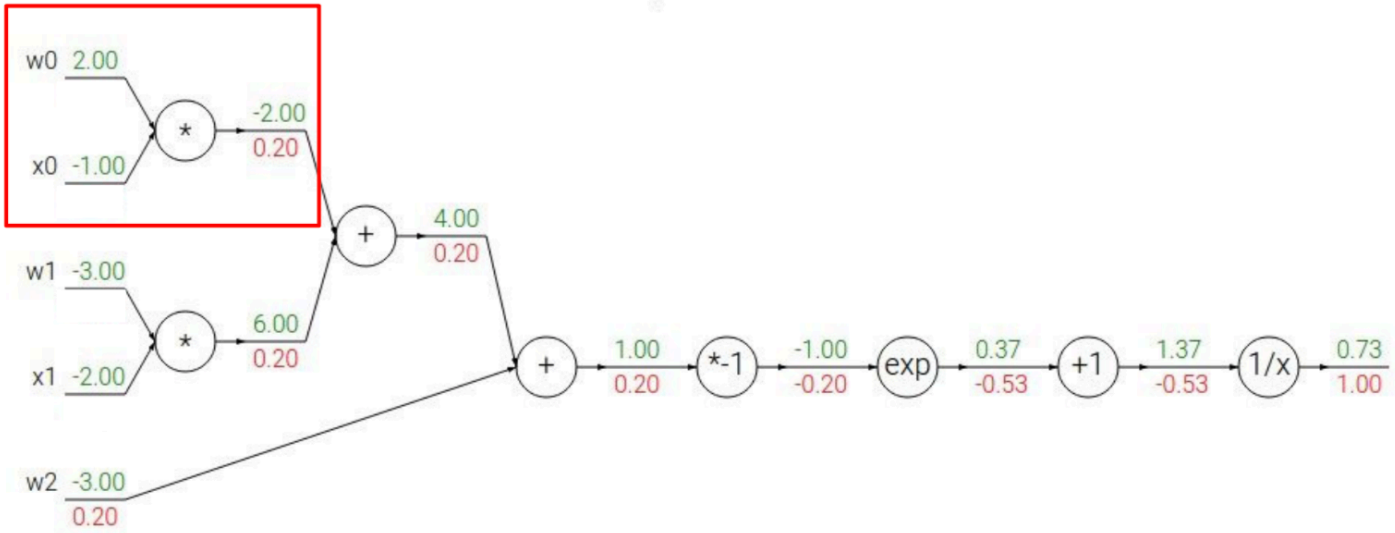
$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$

$f_a(x) = ax \rightarrow \frac{df}{dx} = a$

$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$

$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$

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



$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

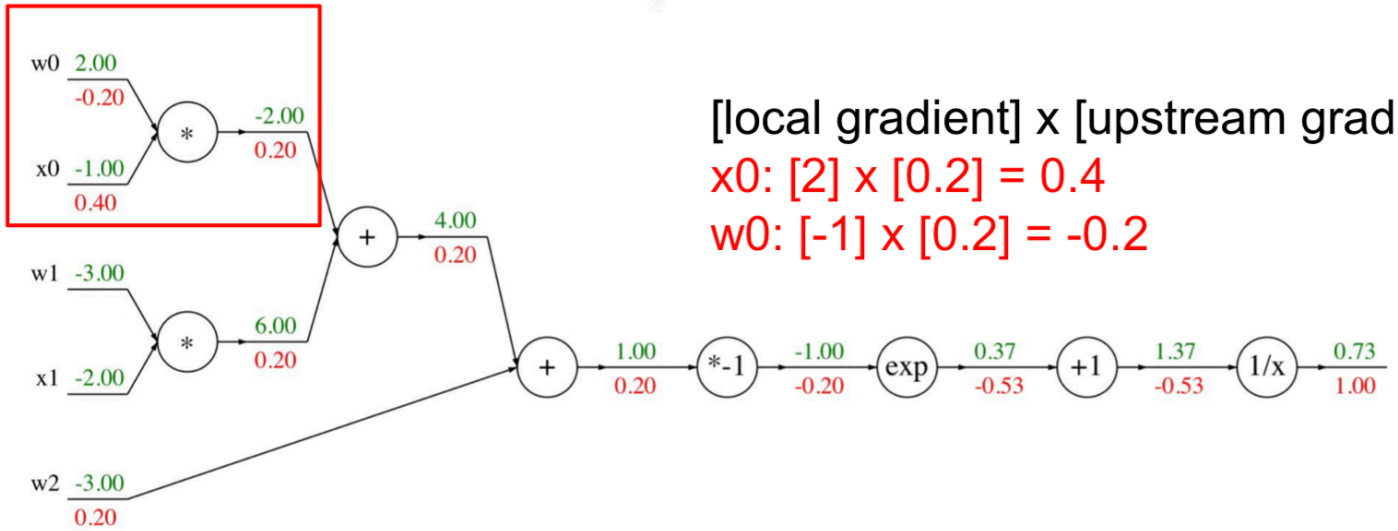
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example:

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



[local gradient] x [upstream gradient]

x0: [2] x [0.2] = 0.4

w0: [-1] x [0.2] = -0.2

$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$

$f_a(x) = ax \rightarrow \frac{df}{dx} = a$

$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$

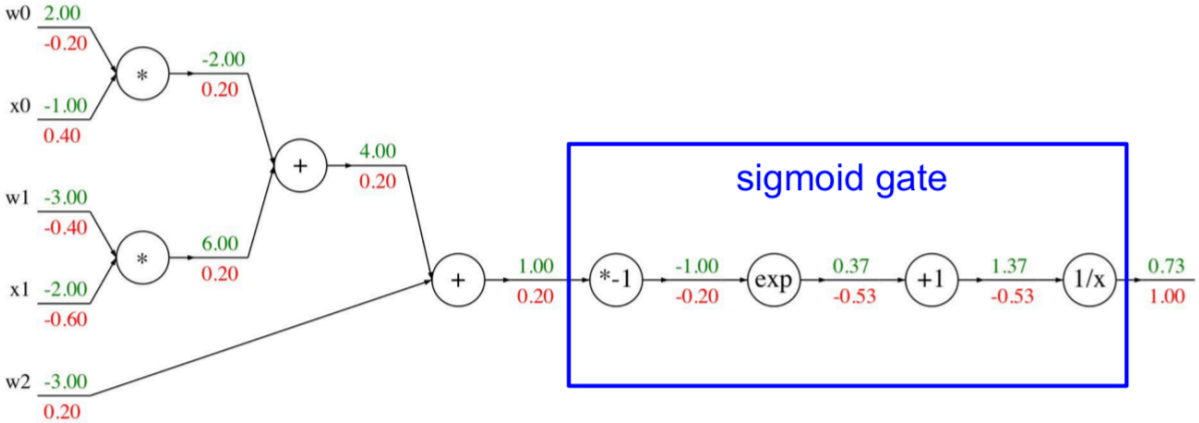
$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$

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

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

sigmoid function

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

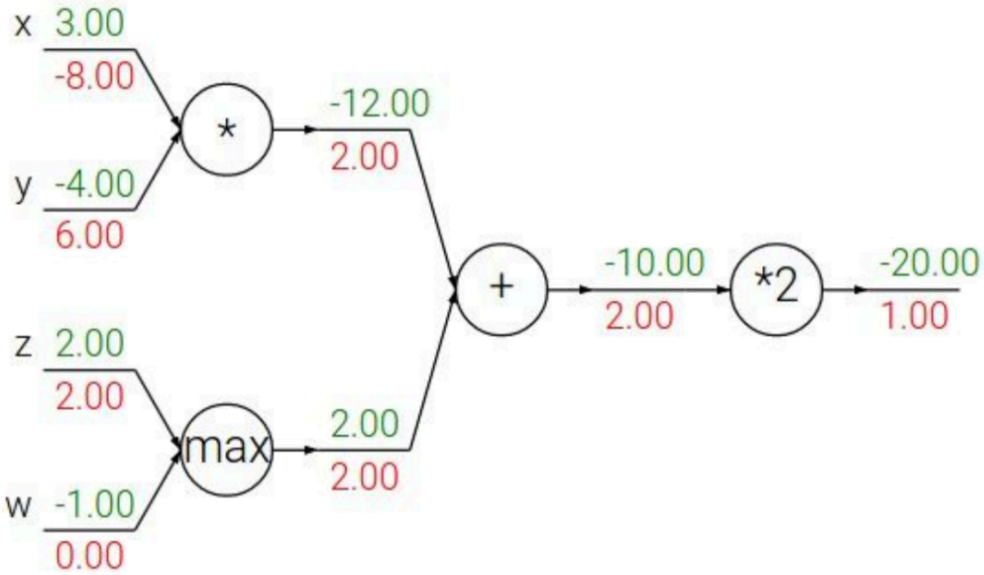


Patterns in backward flow

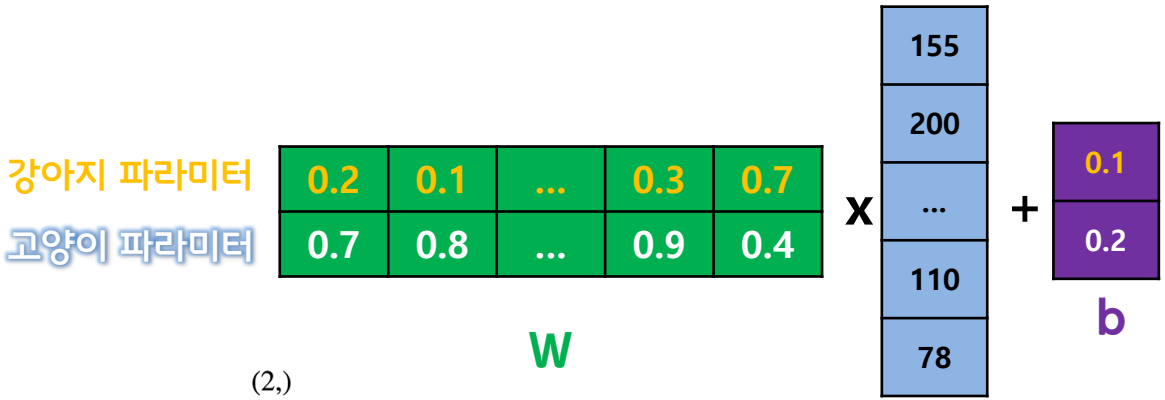
add gate: gradient distributor

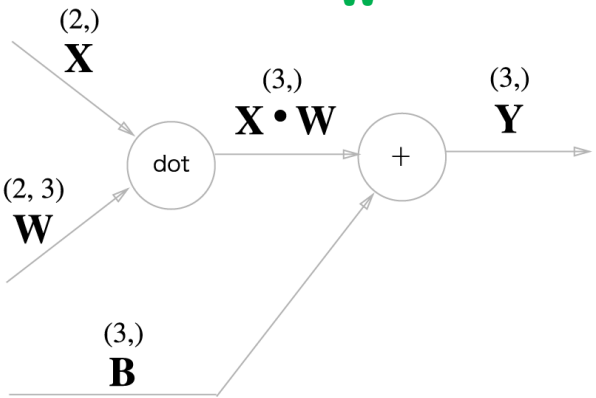
max gate: gradient router

mul gate: gradient switcher



Computational Graph





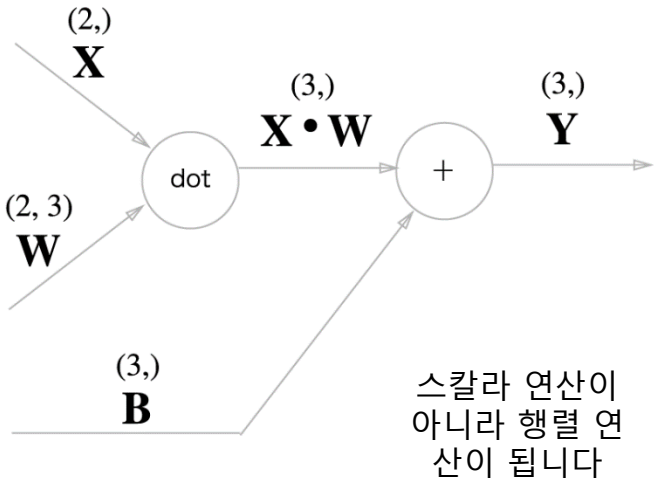
- Affine 계층

Affine / Softmax 계층 구현하기

모두의연구소

- 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
```

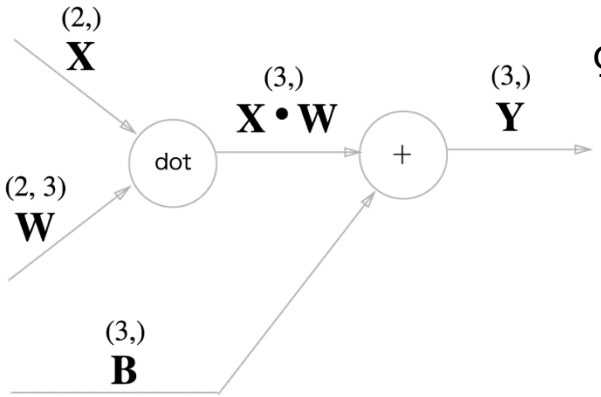


- 신경망의 순전파 때 수행하는 행렬의 내적은 기하학에서 어파인 변환(affine transformation)이라고 합니다.

Affine / Softmax 계층 구현하기

모두의연구소

- Affine 계층



역전파

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

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

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

Affine / Softmax 계층 구현하기

모두의연구소

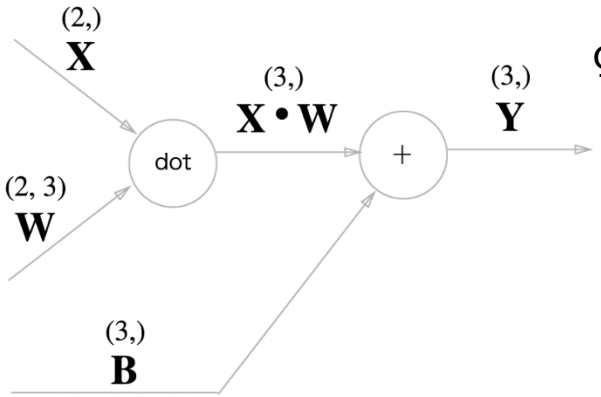
- Affine 계층
  - $\mathbf{W}^{\mathrm{T}}$의 T는 전치(transpose)행렬을 뜻합니다. 전치행렬은 $\mathbf{W}$의 (i, j) 위치의 원소를 (j, i) 위치로 바꾼 것을 말합니다.

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

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

Affine / Softmax 계층 구현하기

- Affine 계층



역전파

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

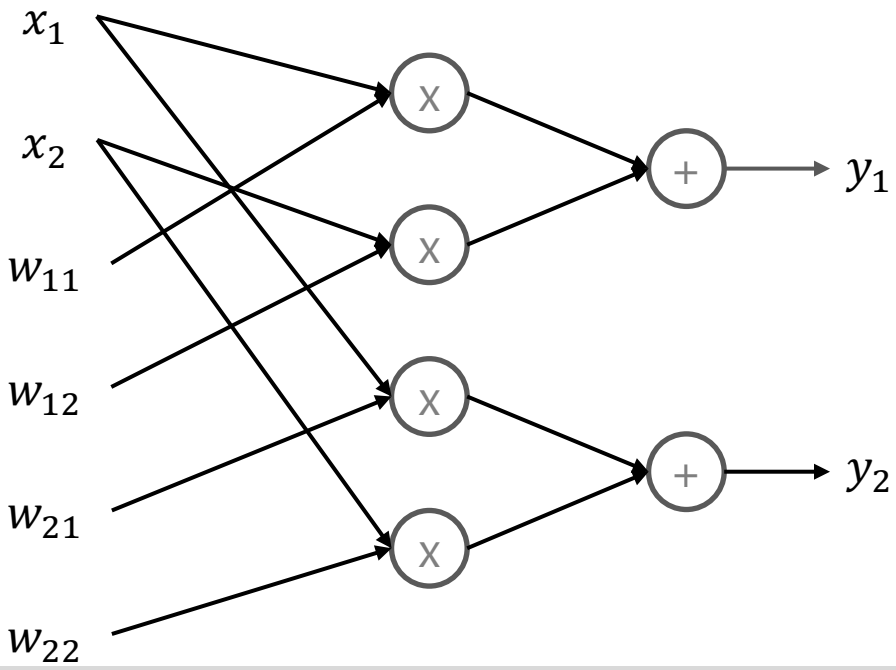
$$\frac{\partial L}{\partial \mathbf{W}} = \mathbf{X}^{\mathrm{T}} \cdot \frac{\partial L}{\partial \mathbf{Y}}$$

이건 왜 이렇게 된건가요?

Computational Graph

- Affine 계층

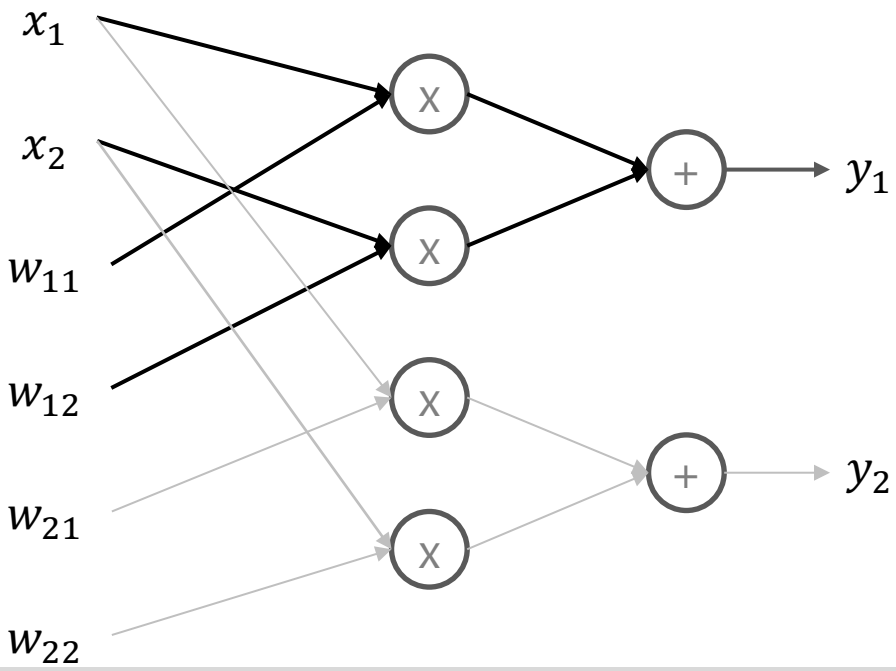
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

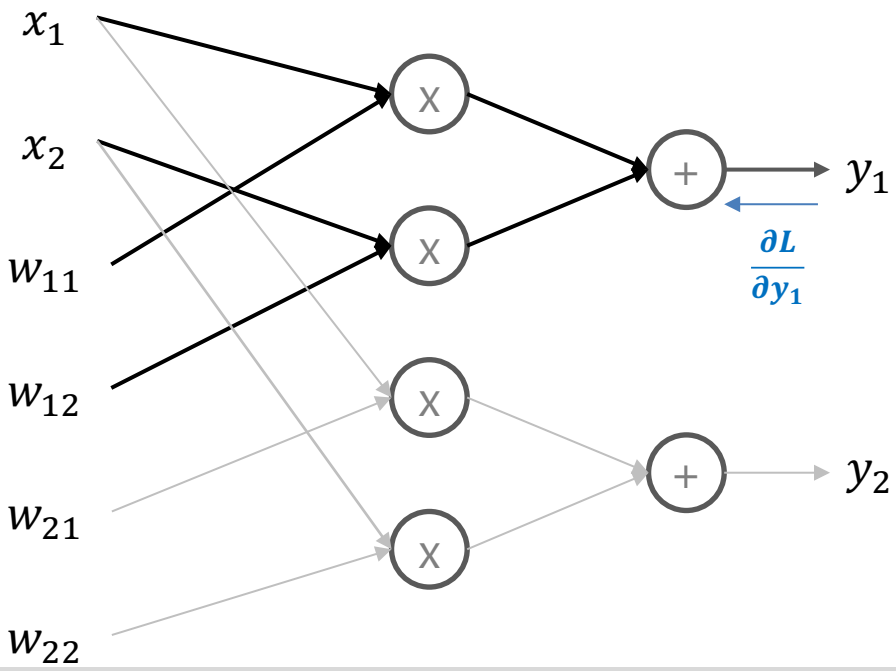
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

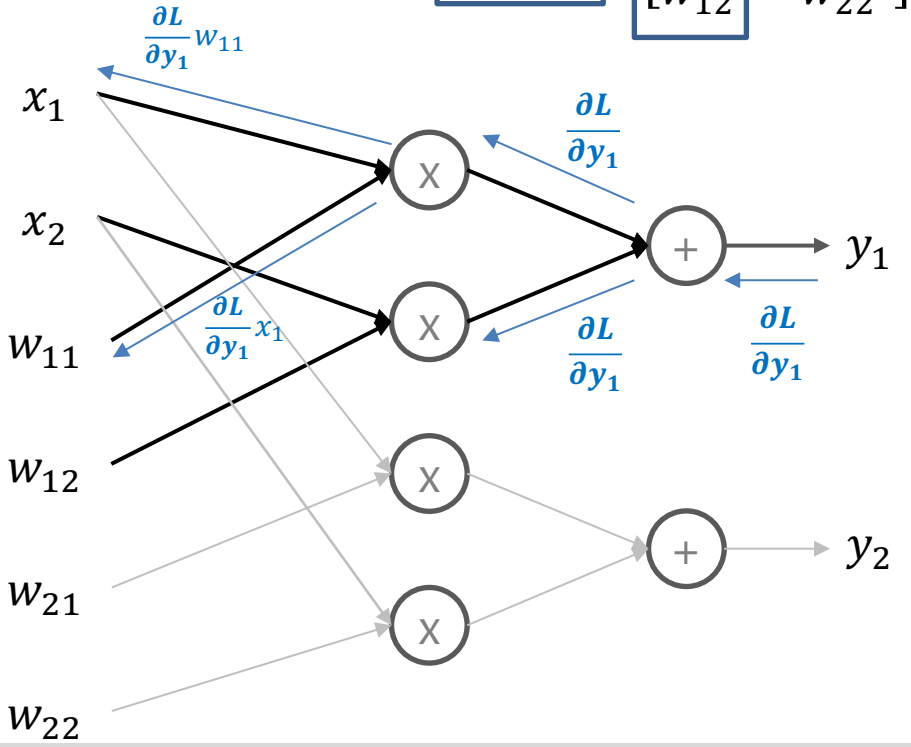
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

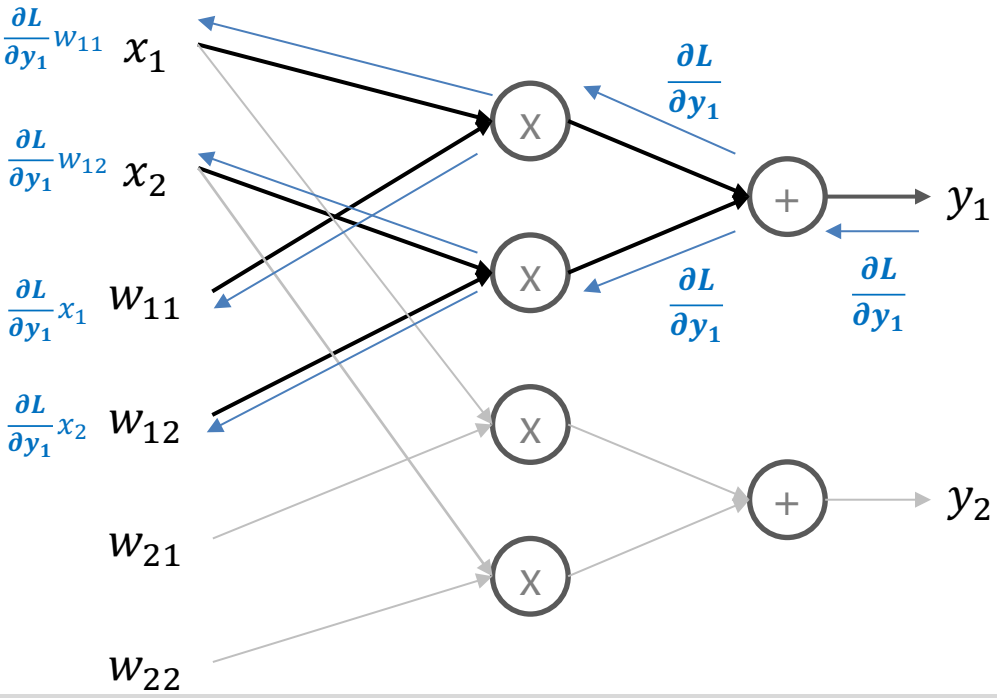
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

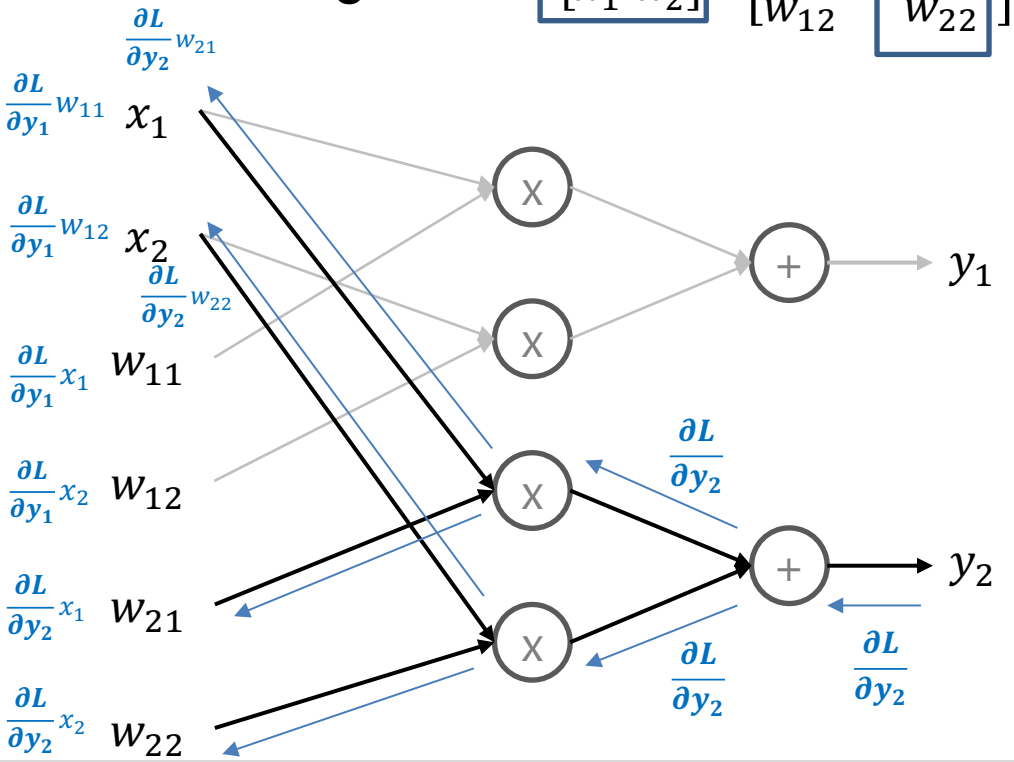
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

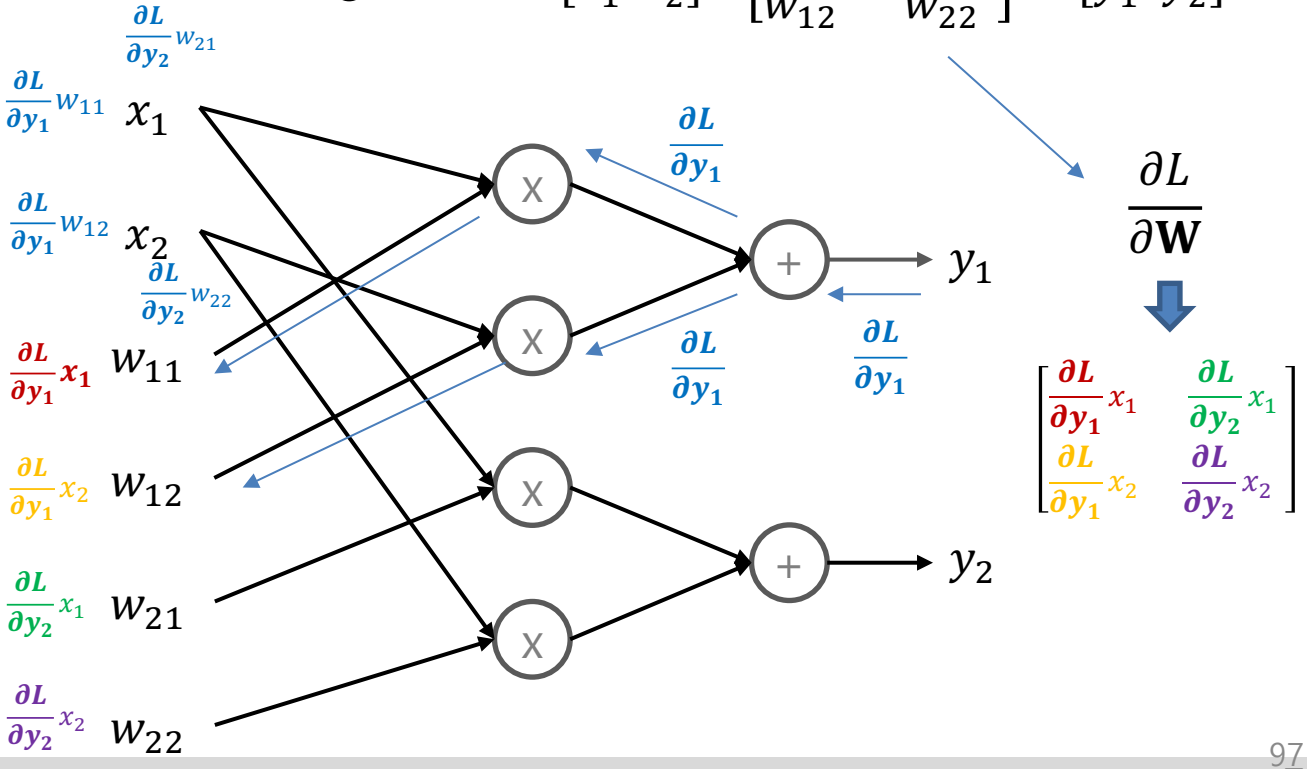
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

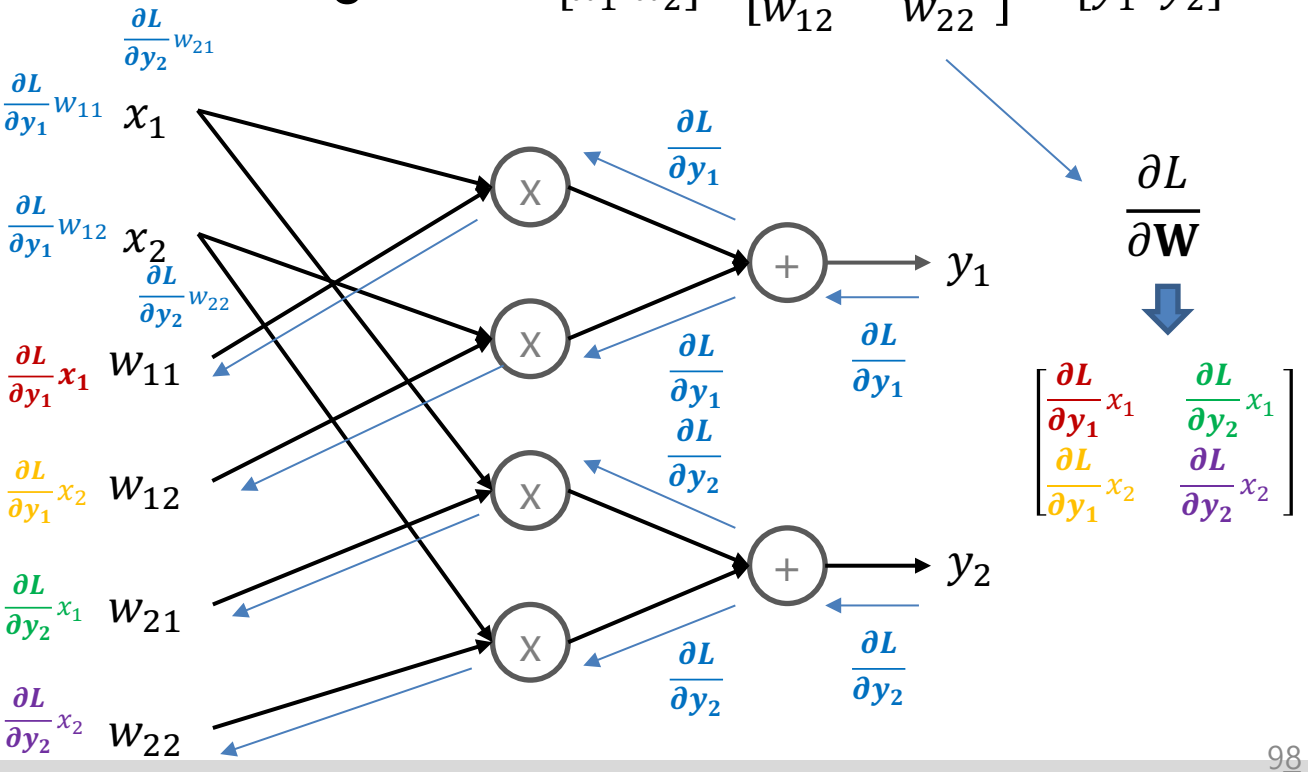
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

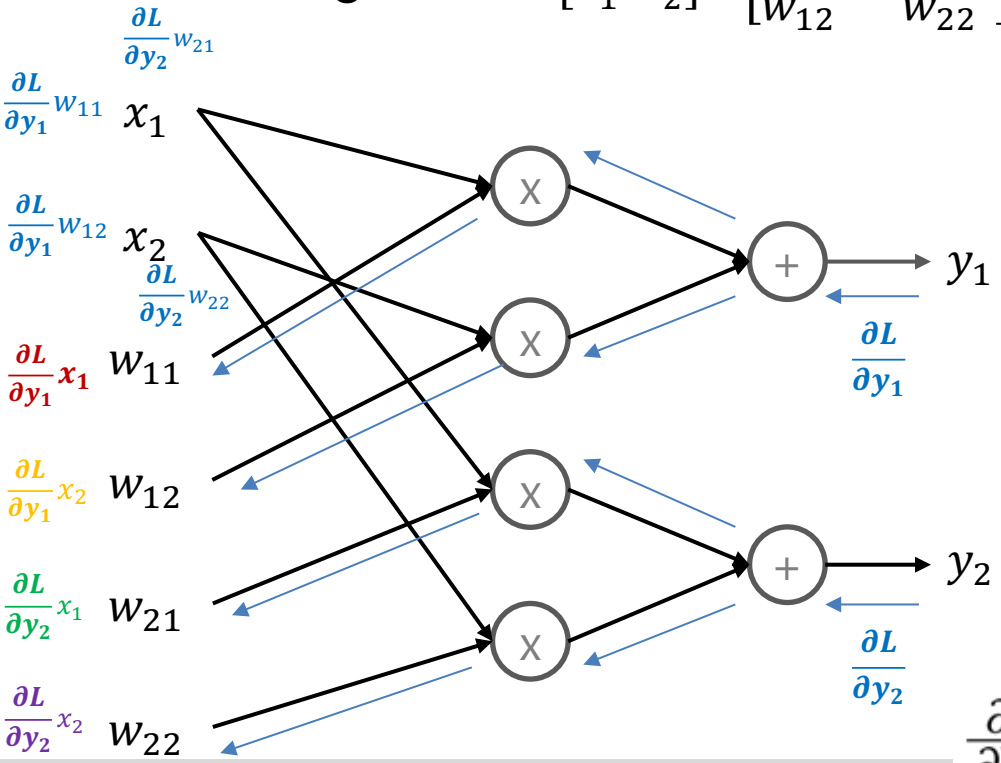
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



Computational Graph

- Affine 계층

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



$$\frac{\partial L}{\partial \mathbf{W}}$$

$$\begin{bmatrix} \frac{\partial L}{\partial y_1} x_1 & \frac{\partial L}{\partial y_2} x_1 \\ \frac{\partial L}{\partial y_1} x_2 & \frac{\partial L}{\partial y_2} x_2 \end{bmatrix}$$

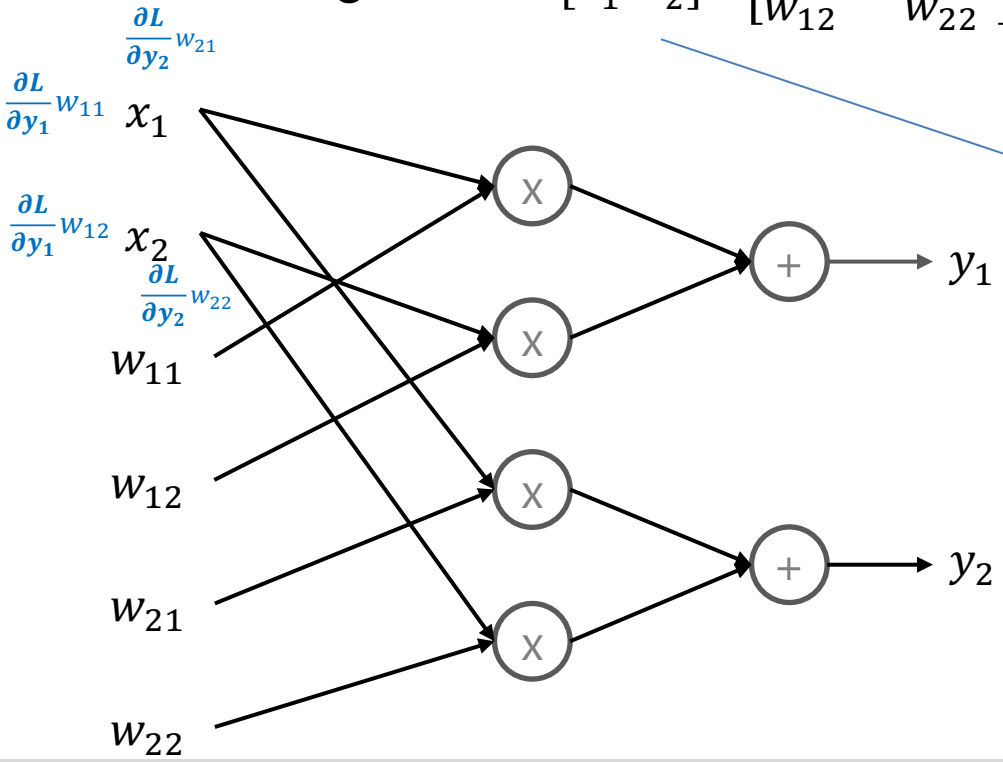
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial L}{\partial y_1} & \frac{\partial L}{\partial y_2} \end{bmatrix}$$

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

Computational Graph

- Affine 계층

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



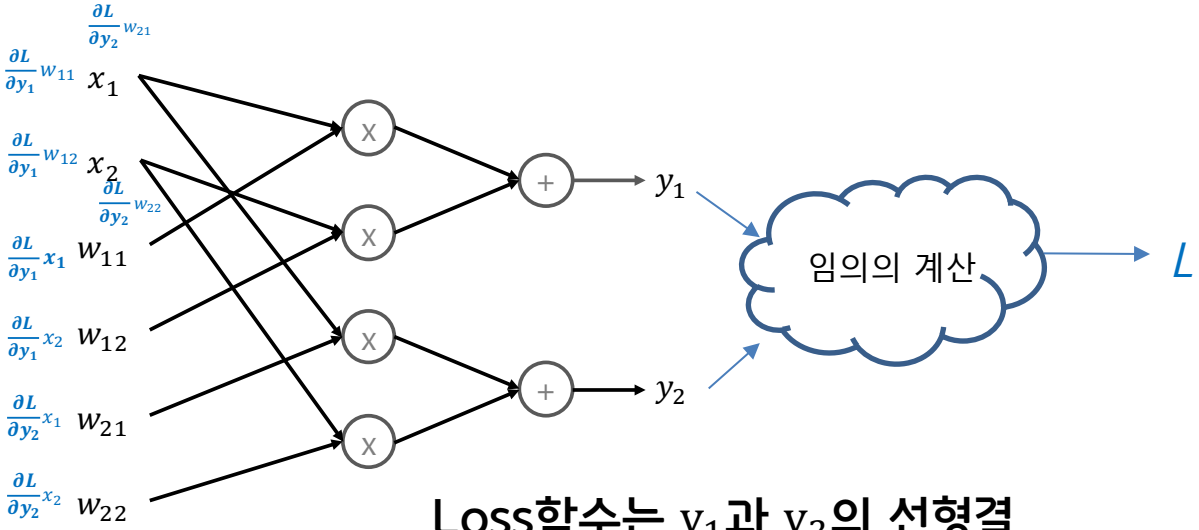
$$\frac{\partial L}{\partial \mathbf{X}}$$

2개씩 나왔어요
ㅠㅠ

Computational Graph

- Affine 계층

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



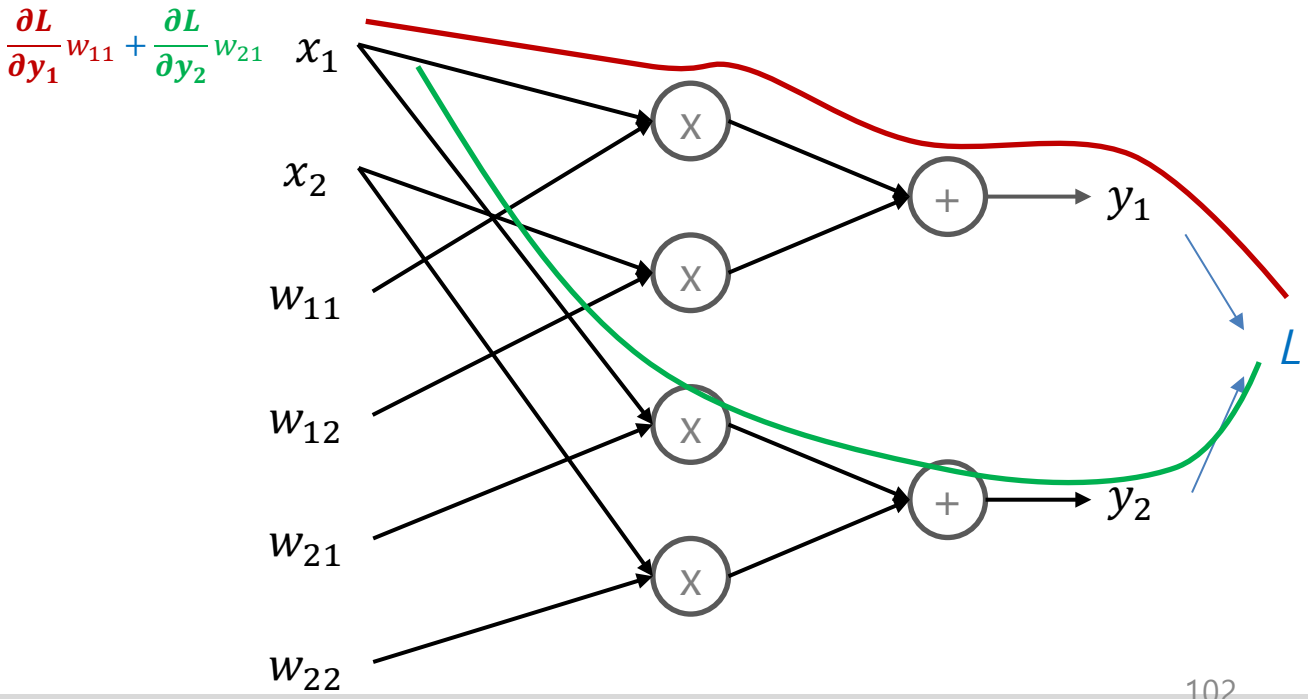
Loss함수는 y_1과 y_2의 선형결합일 겁니다

예) $L = ay_1 + by_2$

Computational Graph

- Affine 계층

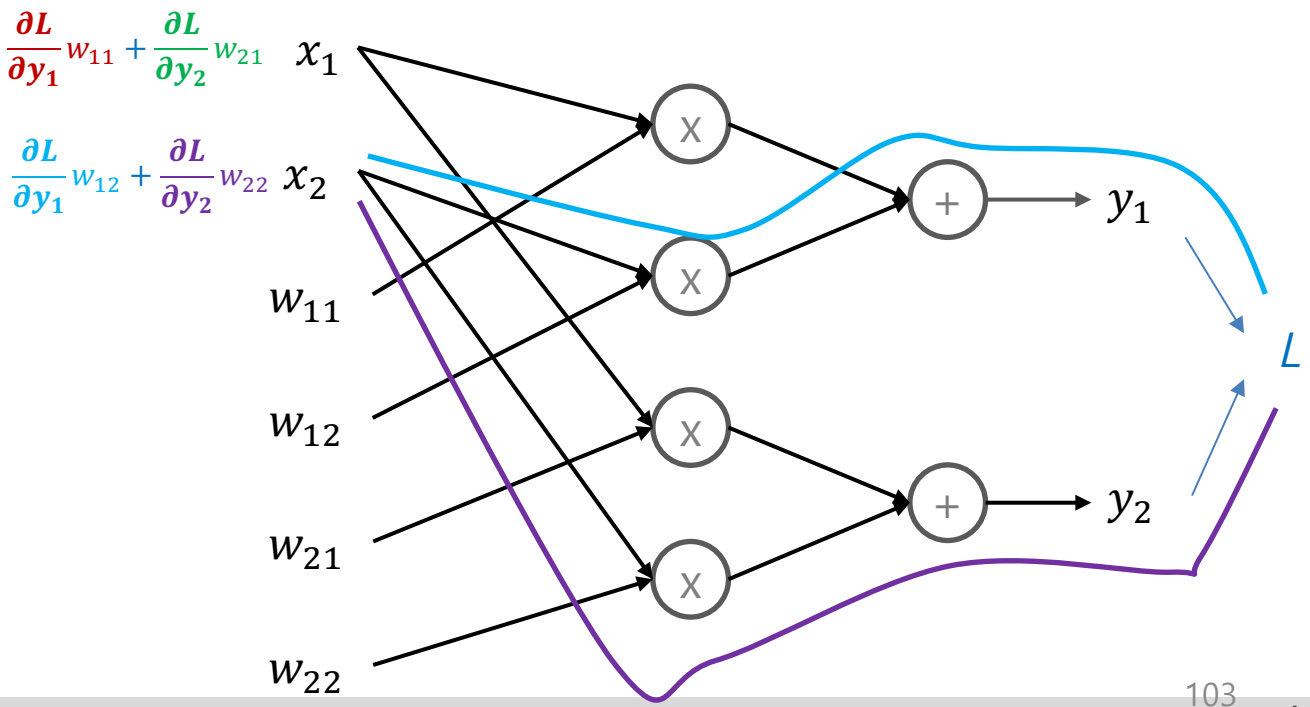
$$[x_1 \ x_2] \times \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} = [y_1 \ y_2]$$



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

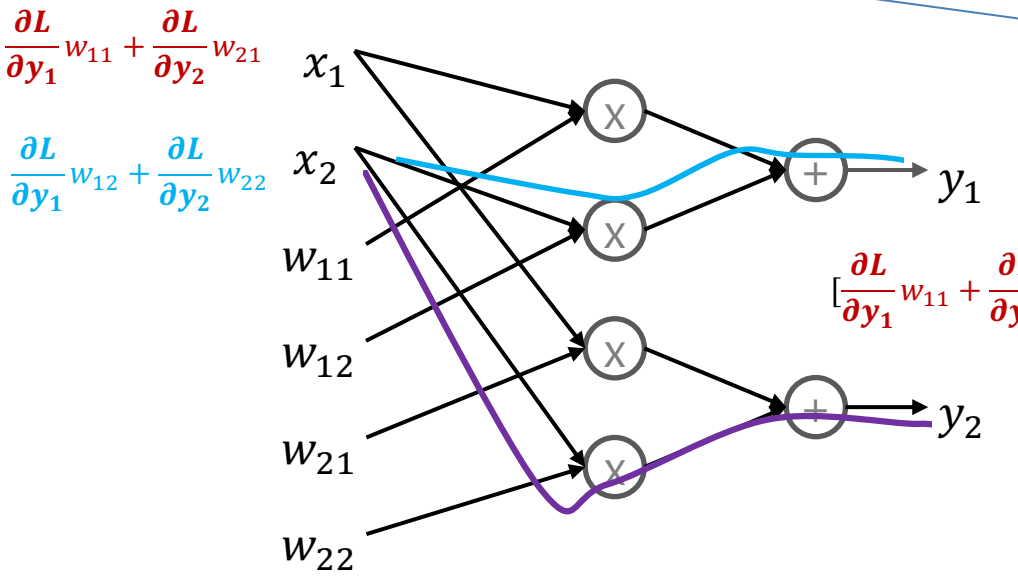
모두의연구소

- Affine 계층

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

$$\frac{\partial L}{\partial \mathbf{X}}$$

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



Computational Graph

- Affine 계층

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

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

⇩

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

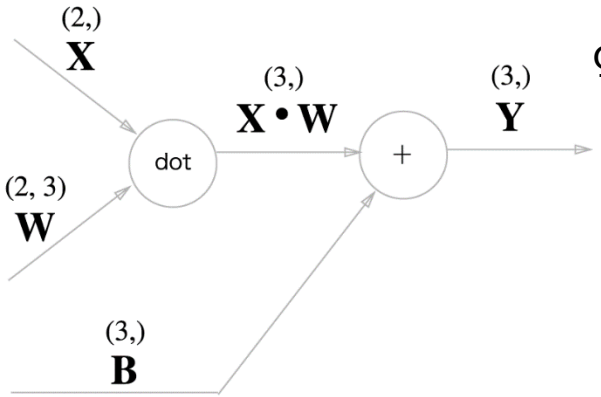
⇩

이제 다시 예제로 돌아와 봅시다

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

Computational Graph

- Affine 계층



역전파

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

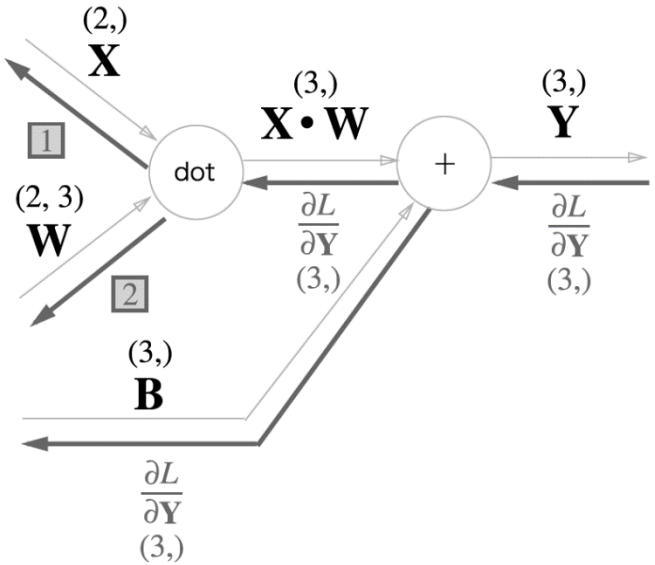
$$\frac{\partial L}{\partial \mathbf{W}} = \mathbf{X}^{\mathrm{T}} \cdot \frac{\partial L}{\partial \mathbf{Y}}$$

Computational Graph

- Affine 계층

$$\boxed{1}\ \underset{(2,)}{\frac{\partial L}{\partial \mathbf{X}}} = \underset{(3,)}{\frac{\partial L}{\partial \mathbf{Y}}}\ \underset{(3,2)}{\mathbf{W}^{\mathrm{T}}}$$

$$\boxed{2}\ \underset{(2,3)}{\frac{\partial L}{\partial \mathbf{W}}} = \underset{(2,1)}{\mathbf{X}^{\mathrm{T}}}\ \underset{(1,3)}{\frac{\partial L}{\partial \mathbf{Y}}}$$



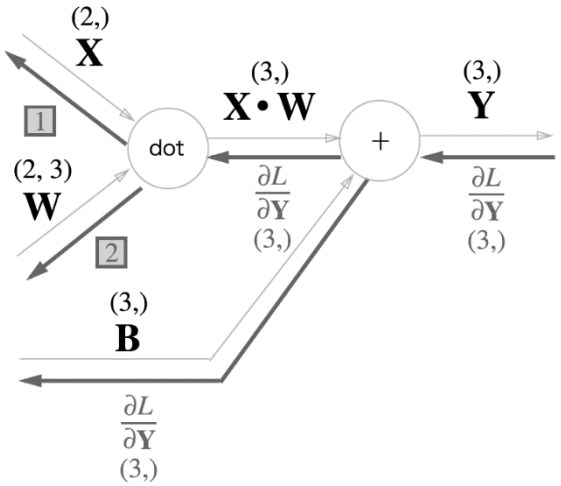
매트릭스의 형상(shape)을 잘 생각해 보면 쉽습니다

Computational Graph

- Affine 계층

$$\boxed{1}\ \underset{(2,)}{\frac{\partial L}{\partial \mathbf{X}}} = \underset{(3,)}{\frac{\partial L}{\partial \mathbf{Y}}}\ \underset{(3,2)}{\mathbf{W}^{\mathrm{T}}}$$

$$\boxed{2}\ \underset{(2,3)}{\frac{\partial L}{\partial \mathbf{W}}} = \underset{(2,1)}{\mathbf{X}^{\mathrm{T}}}\ \underset{(1,3)}{\frac{\partial L}{\partial \mathbf{Y}}}$$



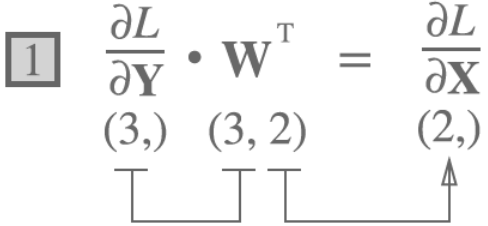
$\boldsymbol{X}$와 $\frac{\partial L}{\boldsymbol{X}}$와 형상이 같아야 하고 $\boldsymbol{W}$와 $\frac{\partial L}{\boldsymbol{W}}$는 형상이 같아야 합니다

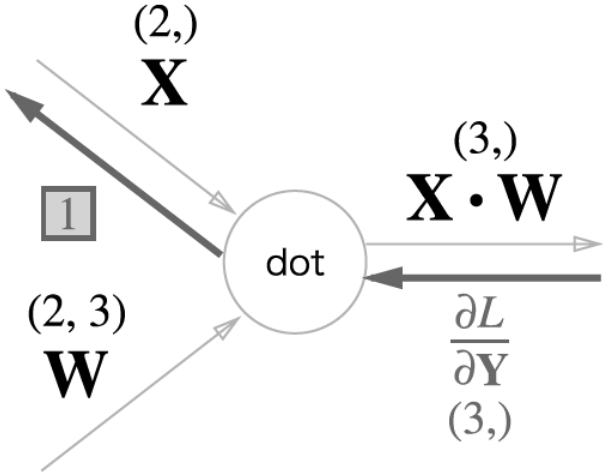
$$\mathbf{X} = (x_0, x_1, \cdots, x_n)$$

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

Computational Graph

- Affine 계층

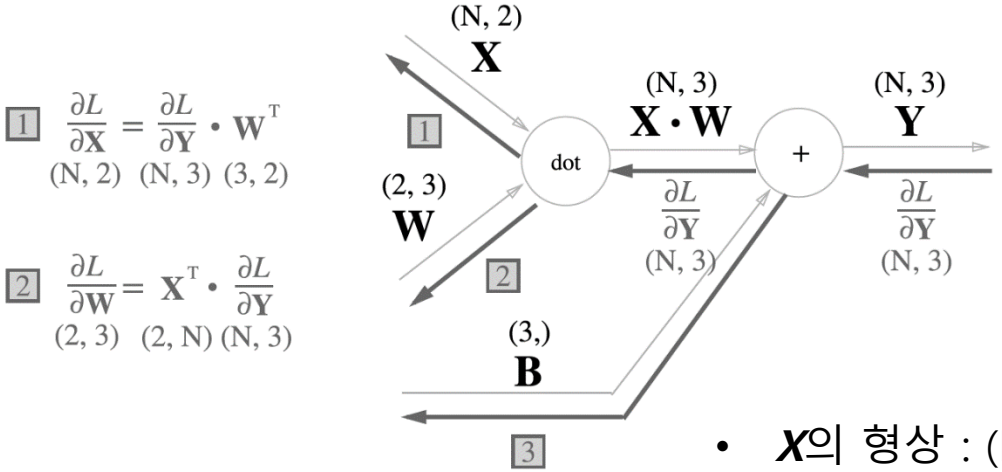




그냥 곱셈계층으로 이해하고 형상을 맞춰주면 됩니다

Computational Graph

- 배치용 Affine 계층



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

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

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

- ***X***의 형상 : (N, 2)

- 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

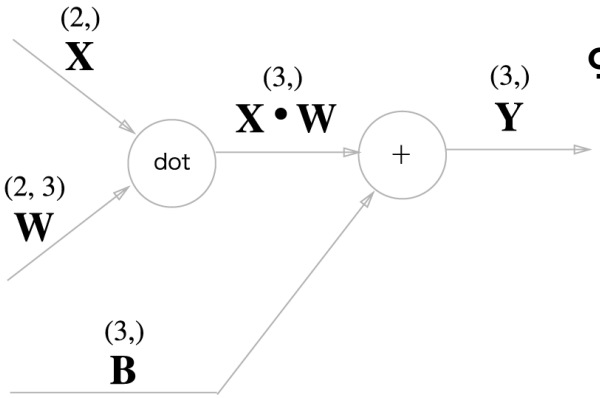
역전파의 경우

```
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

모두의연구소

- Affine 계층



역전파

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

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

w 매트릭스가 업데이트 되겠군요

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

모두의연구소

신경망 관련 계층구현

활성화 함수계층 구현하기

모두의연구소

- ReLU 계층

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

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


```
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 dout[self.mask] = 0
스크 이용  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  dout[self.mask] = 0
스크 이용  dx = dout

           return dx
```

활성화 함수계층 구현하기

모두의연구소

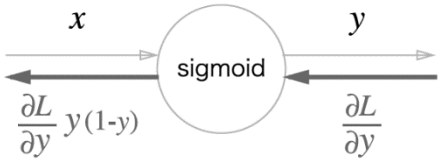
- Sigmoid 계층 : $y = \frac{1}{1 + \exp(-x)}$

  - /codes/common/layers.py

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

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

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

출력 y를 저장

계산



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 경과 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          ###################################
71          ######### Write your codes #######
72          ###################################
73          grads['W2'] = None
74          grads['b2'] = None
75
76          da1 = None
77          dz1 = None
78          grads['W1'] = None
79          grads['b1'] = None
80
81          return grads
```

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

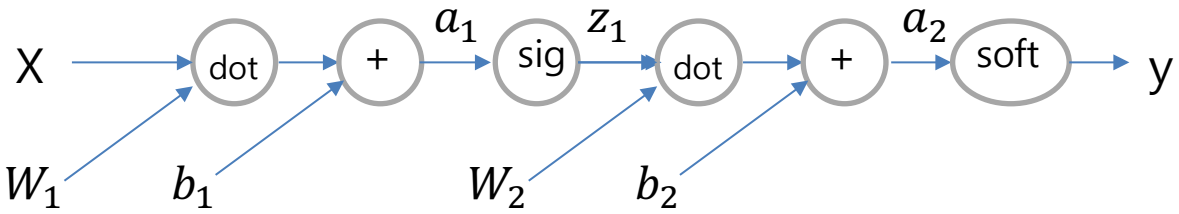
```
55      def gradient(self, x, t):
56          W1, W2 = self.params['W1'], self.params['W2']
57          b1, b2 = self.params['b1'], self.params['b2']
58          grads = {}
59
60          batch_num = x.shape[0]
61
62          # forward
63          a1 = np.dot(x, W1) + b1
64          z1 = sigmoid(a1)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

실습 1 : 2 레이어 네트워크 실습

모두의연구소

- 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)
```

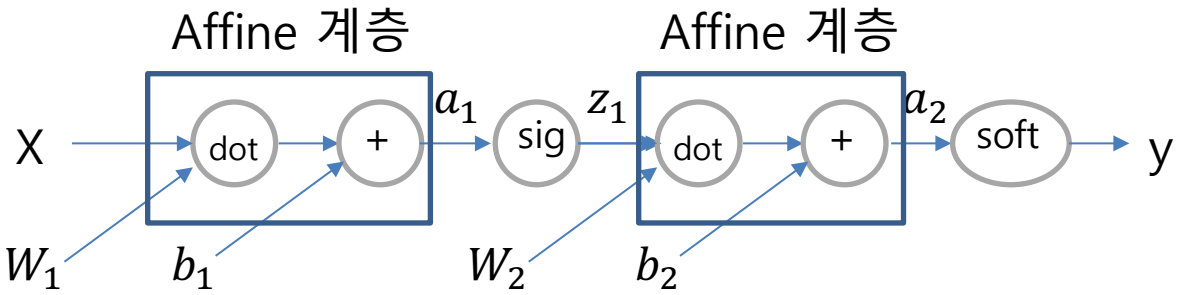


실습 1 : 2 레이어 네트워크 실습

모두의연구소

- 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)
```



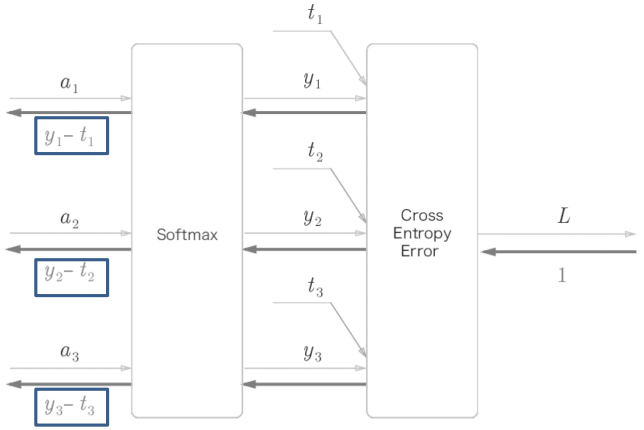
실습 1 : 2 레이어 네트워크 실습

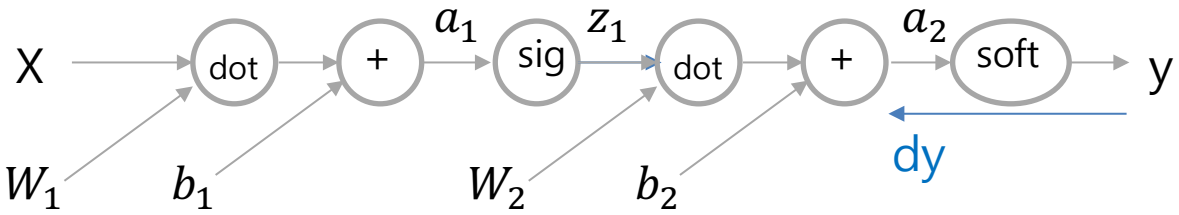
모두의연구소

- 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        ###################################
71        ######### Write your codes ########
72        ###################################
73        grads['W2'] = None
74        grads['b2'] = None
75
76        da1 = None
77        dz1 = None
78        grads['W1'] = None
79        grads['b1'] = None
80
81        return grads
```

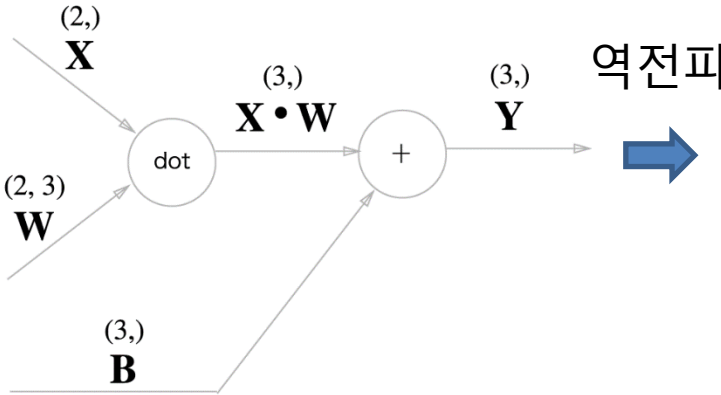




실습 1 : 2 레이어 네트워크 실습

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

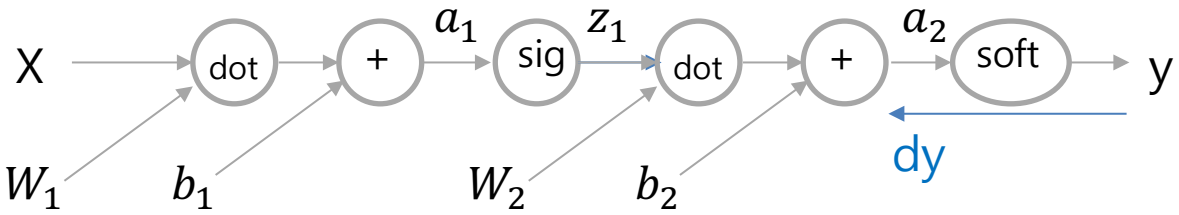
Affine 계층의 역전파를 기억해 봅시다



역전파

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

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



실습 1 : 2 레이어 네트워크 실습

모두의연구소

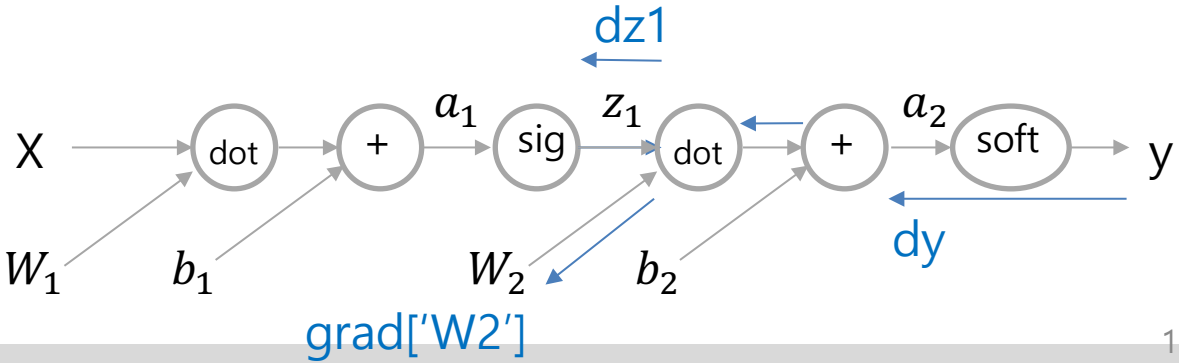
- 2) 구현할 함수 : 2day/prac1_simple_two_Layer_net.py

  Affine 계층의 역전파를 기억해 봅시다

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

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

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



실습 1 : 2 레이어 네트워크 실습

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

  Affine 계층의 배치데이터 편향을 기억해 봅시다

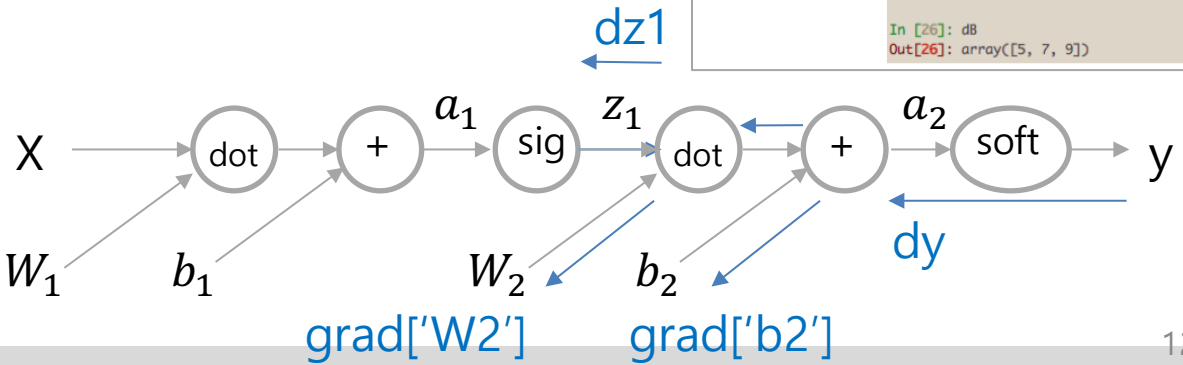
```
68    # backward
69    dy = (y - t) / batch_num
70    ##################################
71    ######### Write your codes #######
72    ##################################
73    grads['W2'] = None
74    grads['b2'] = None
75
76    dz1 = None
77    da1 = None
78    grads['W1'] = None
79    grads['b1']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

Affine / Softmax 계층 구현하기

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

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])
```



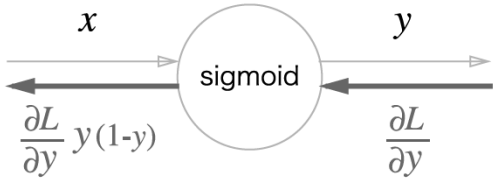
실습 1 : 2 레이어 네트워크 실습

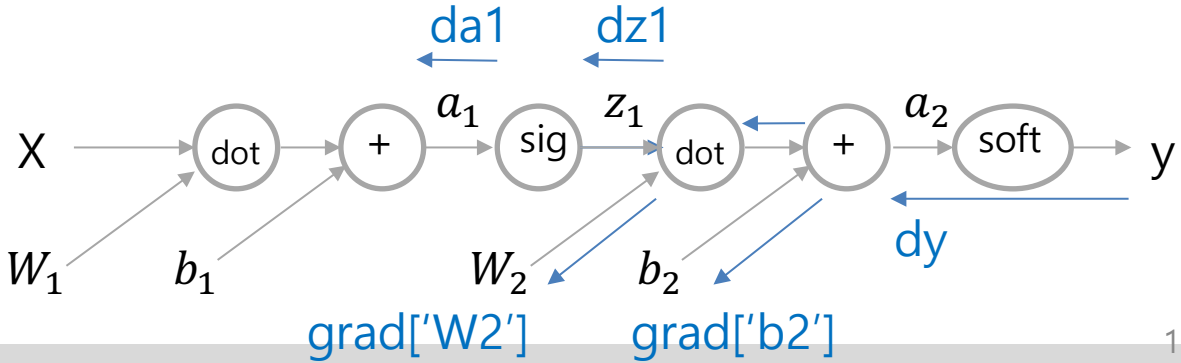
모두의연구소

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

  Sigmoid 역전파 함수를 기억해 보세요

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





실습 1 : 2 레이어 네트워크 실습

모두의연구소

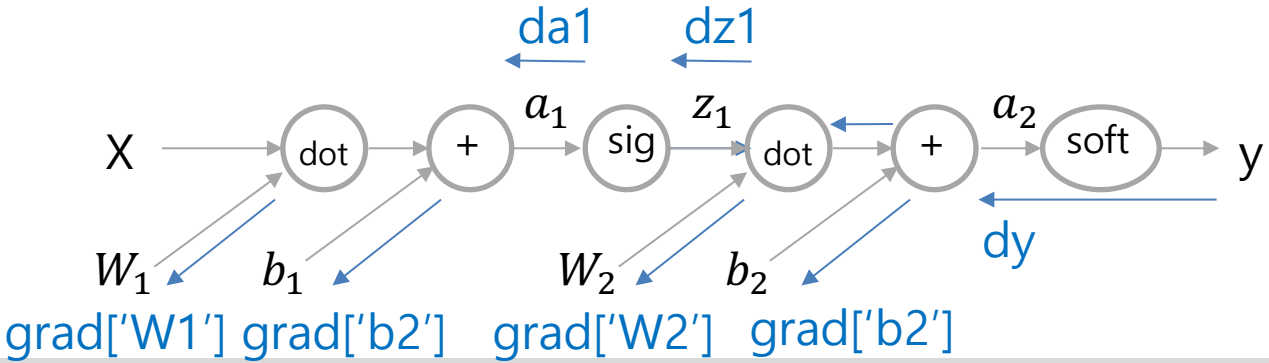
- 2) 구현할 함수 : 2day/prac1_simple_two_Layer_net.py

  Affine 계층의 역전파를 기억해 보세요.

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

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

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



지금까지 살펴본 내용

모두의연구소

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

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

박 은 수 Research Director

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