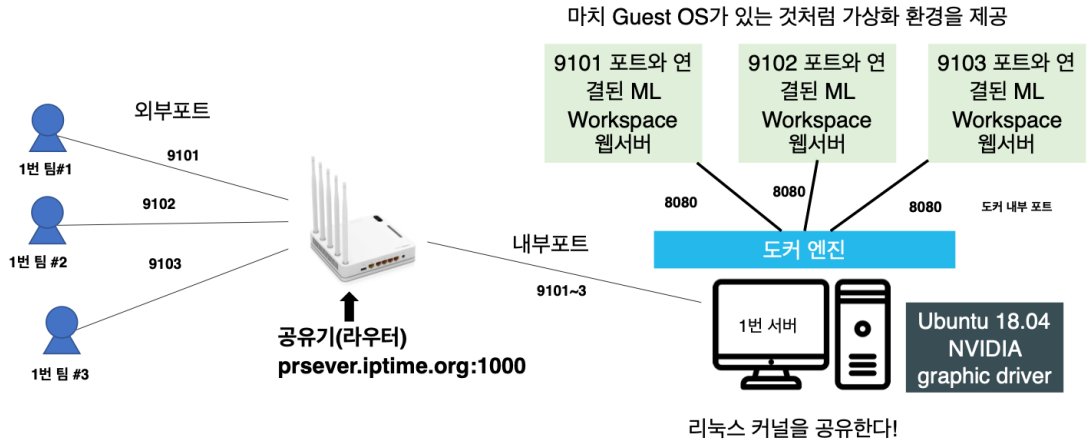
ML workspace_03

1. 지금까지 배운 스터디 내용 정리

중요! 네트워크 연결 과정



첫번째 포트 포워딩은 공유기에서, 두번째 포트 포워딩은 도커 명령을 통해 수행됨

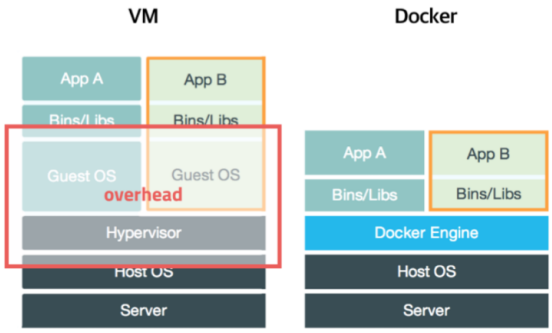
1. 지금까지 배운 스터디 내용 정리

도커 사용해서 웹서버 띄우기

Docker 란?

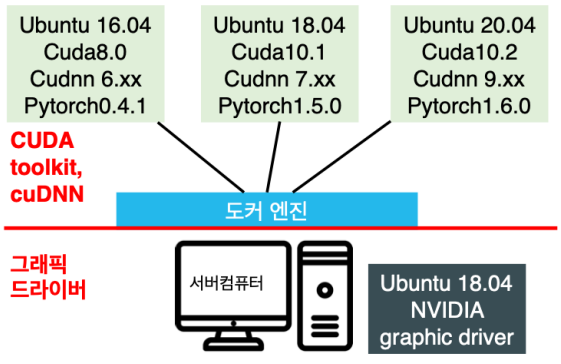
서버 관리 및 운영을 위한 가상화 기술

2014년부터 구글은 모든 서비스(지메일, 클라우드, 유튜브 등)를 도커로 유지 및 관리, 운영했으며 1주일동안 평균 20억 개의 도커 컨테이너가 사용됨.



가상머신과 도커

마치 Guest OS가 있는 것처럼 가상화 환경을 제공하지만 메모리, CPU 사용의 오버헤드가 없다!

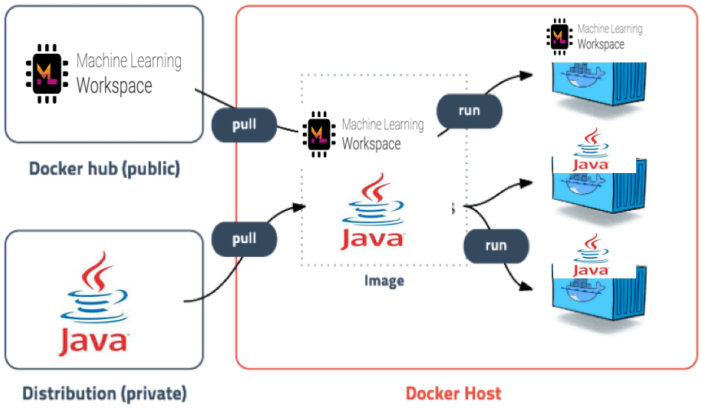


리눅스 커널을 공유한다!

1. 지금까지 배운 스터디 내용 정리 ML workspace는 하나의 이미지일 뿐이다.

도커 이미지와 컨테이너

- 중요 ML workspace 웹서버는 수 많은 이미지 중 하나다.
- 이미지는 저장된 파일일 뿐 컨테이너로 실행되어야 가상환경이 만들어진다.



Docker image 도커 이미지 공유 사이트: https://hub.docker.com/

도커이미지는 ML workspace뿐만 아니라 다른 이미지도 올 수 있다.

1. 지금까지 배운 스터디 내용 정리 현재 도커 컨테이너 이미지로 저장해보기

docker commit은 컨테이너의 환경을 이미지로 저장해준다.

$ docker commit [컨테이너명] [이미지이름]

ex) docker commit ky_study ky_study

commit된 이미지를 컨테이너로 띄워보자!

$ nvidia-docker run --ipc=host --restart always -p 9201:8080 -v /home/prlab/kunyoung:/workspace --name ky_study --env WORKSPACE_AUTH_USER=kunyoung --env WORKSPACE_AUTH_PASSWORD=1234 [자신이 저장한 이미지 이름]

1. 지금까지 배운 스터디 내용 정리 도커 최종 정리

ML Workspace 환경 기반 리눅스 서버 사용을 위한 필수 도커 명령어 정리

$docker images

$docker run

$docker ps –a

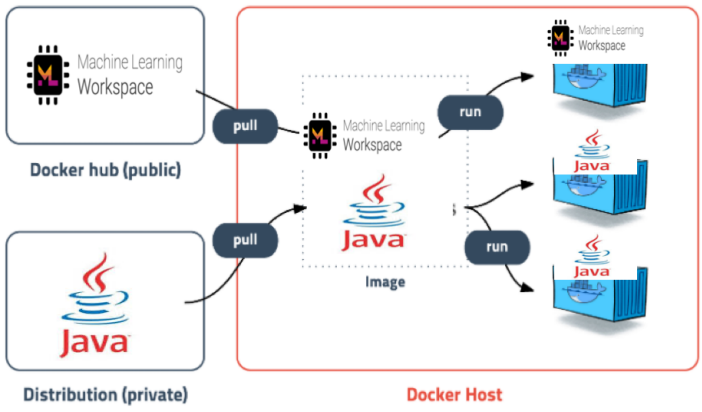
$docker commit

$docker stop

$docker start

$docker rm

$docker rmi

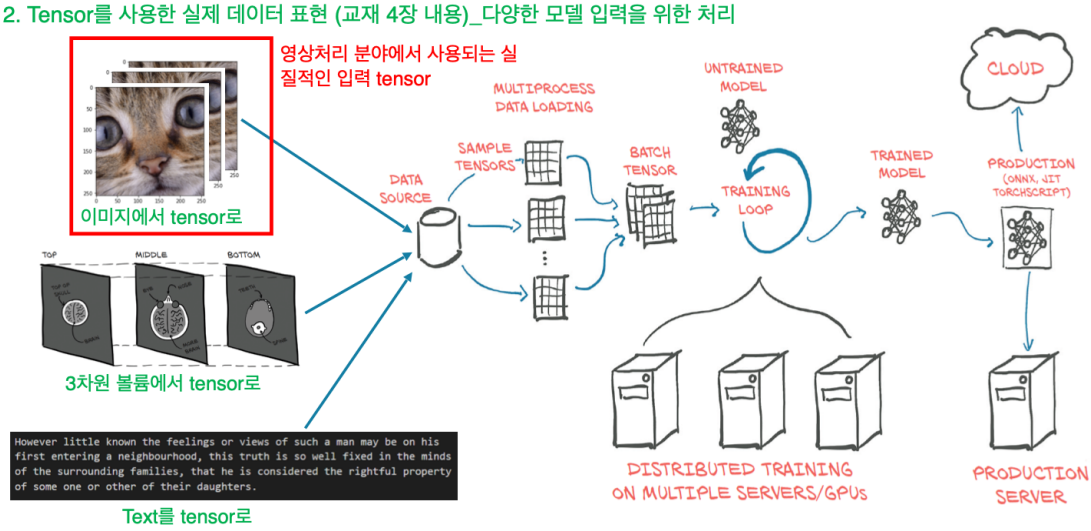


Docker image

1. 지금까지 배운 스터디 내용 정리 Tensor 연산 및 모델 입력을 위한 tensor 변환

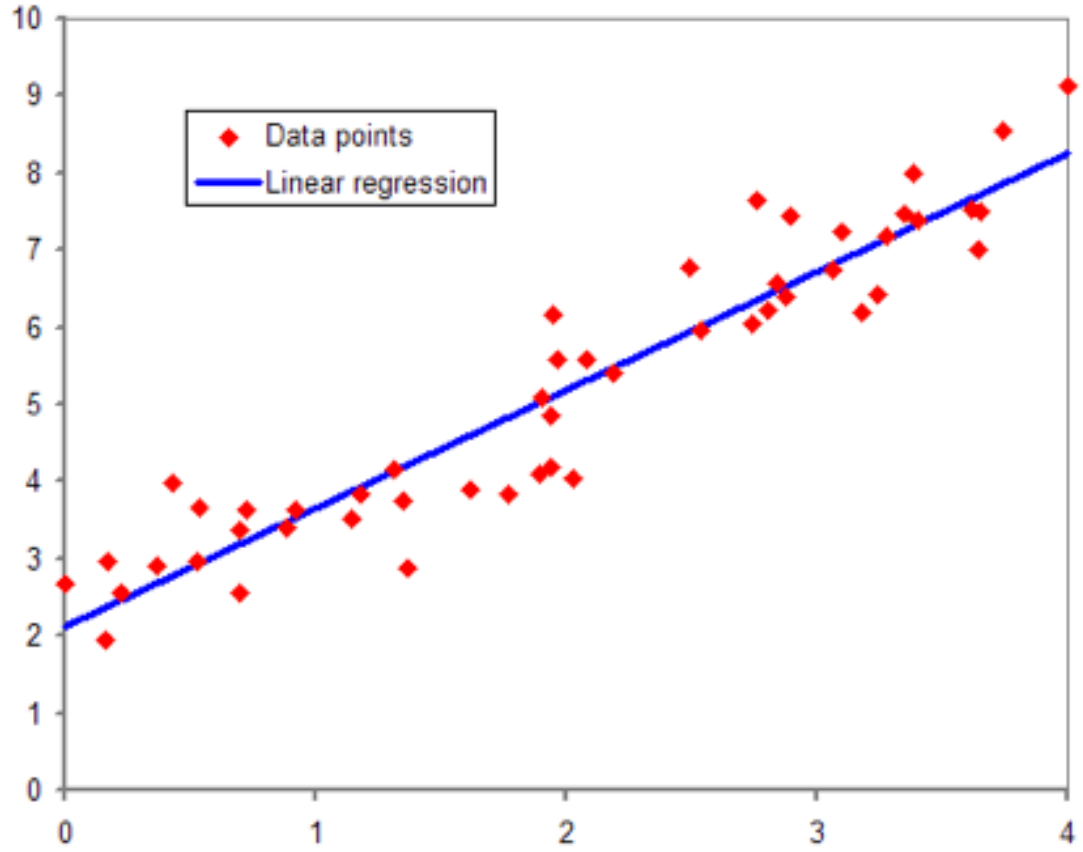
1. PyTorch 기초 tensor 연산 (교재 3장 내용)_Tensor가 무엇인지, tensor를 사용한 연산들

2. Tensor를 사용한 실제 데이터 표현 (교재 4장 내용)_다양한 모델 입력을 위한 처리



머신러닝

머신 러닝에서 사용되는
대부분의 Linear Model (선형 모델)예시



선형모델은 예측하기 쉬운 특징이있다.

1) 훈련 데이터셋과 테스트 데이터셋

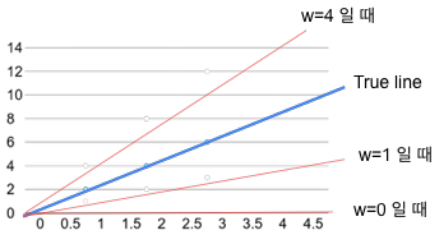
머신러닝 예제)

어떤 학생이 1시간 공부를 했더니 2점, 다른 학생이 2시간 공부를 했더니 4점, 또 다른 학생이 3시간을 공부했더니 6점을 맞았습니다. 그렇다면, **내가 4시간을 공부한다면 몇 점을 맞을 수 있을까요?**

Hours (x)	Points (y)
1	2
2	4
3	6
4	?

이 질문에 대답하기 위해서 1시간, 2시간, 3시간을 공부했을 때 각각 2점, 4점, 6점이 나왔다는 앞서 나온 정보를 이용한다.
예측을 위해 사용하는 데이터를 **훈련 데이터셋(training dataset)**
학습 후, 이 모델이 얼마나 잘 작동하는지 판별하는 데이터셋을 **테스트 데이터셋(test dataset)**

LOSS 함수: 예측한 답이 최적인지 판별하기위한 함수



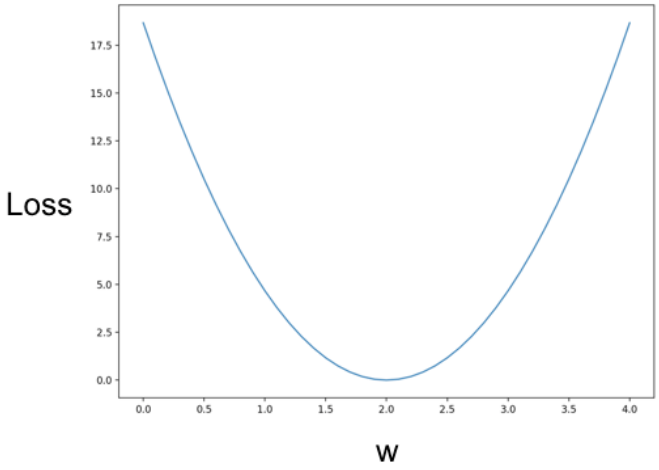
$$loss = (\hat{y} - y)^2 = (x * w - y)^2$$

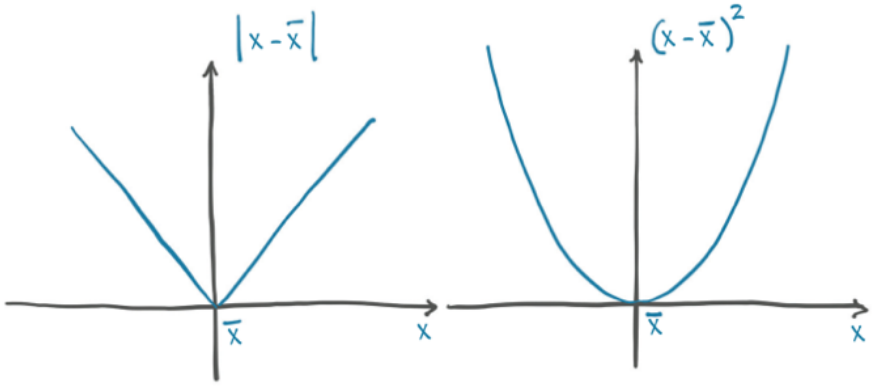
$$loss = \frac{1}{N}\sum_{n=1}^{N}(\hat{y_n} - y_n)^2$$

MSE, mean square error

Hours, x	Loss (w=0)	Loss (w=1)	Loss (w=2)	Loss (w=3)	Loss (w=4)	Points, y
1	4	1	0	1	4	2
2	16	4	0	4	16	4
3	36	9	0	9	36	6
	MSE=56/3=18.7	MSE=14/3=4.7	MSE=0	MSE=14/3=4.7	MSE=56/3=18.7	

LOSS 함수 그래프





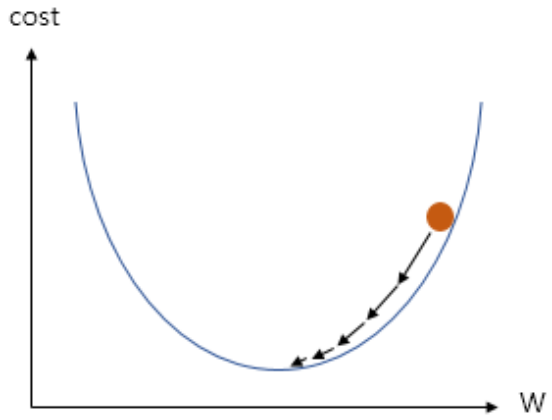
LOSS가 작을수록 최적의 답이다
꼭지점이 있는 그래프는 미분이 어렵다
그래프의 최저점을 잘 찾기위해 둥근모양의 그래프가 좋다

2) 옵티마이저(Optimizer)

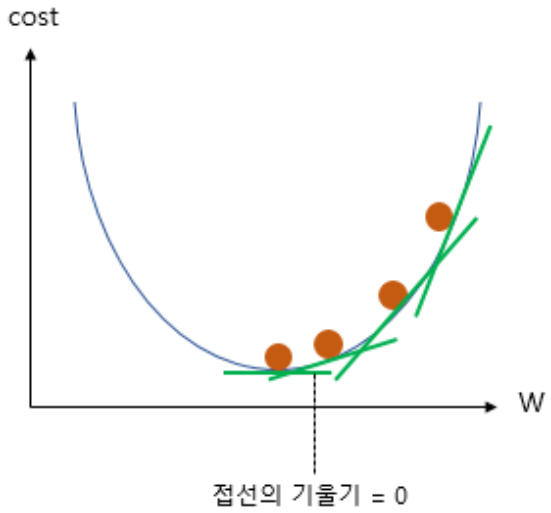
최적화 알고리즘이라고도 부르며, 머신러닝 학습에 사용되는 알고리즘
비용함수의 값을 최소로 하는 **W(가중치)**와 **b(편향)**를 찾는 과정을 머신러닝에서 **학습**이라 한다.

2-1) 경사 하강법(Gradient Descent)

: 가장 기본적인 옵티마이저 알고리즘 중 하나



W(가중치)나 b(편향)이 너무 크거나 작으면 오차가 커져 비용함수 값 cost(W)이 커진다.
기울기 W가 무한대로 커지면 커지거나 무한대로 작아지면 cost의 값 또한 무한대로 커진다.
cost가 가장 작을 때는 맨 아래의 볼록한 부분
기계가 해야할 일은 **cost가 가장 최소값을 가지게 하는 W를 찾는 일**



cost가 최소화가 되는 지점은 **접선의 기울기가 0** 또는 **미분값이 0**이 되는 지점

경사 하강법은 비용 함수를 미분하여 현재 W에서의 접선의 기울기를 구하고,
접선의 기울기가 낮은 방향으로 W의 값을 변경하는 작업을 반복한다.

$$\text{기울기} = \frac{\partial cost(W)}{\partial W}$$

학습률 α(learning rate) : 학습률 α은 W의 값을 변경할 때, 얼마나 크게 변경할지(얼마나 큰 폭으로 이동할지)를 결정한다.

학습률 α를 무작정 크게하면 접선의 기울기가 최소값이 되는 W를 빠르게 찾을 수 있을 것 같지만 그렇지 않다.
반대로 α가 지나치게 낮은 값을 가지면 학습속도가 느려지므로 **적당한 α의 값을 찾아내는 것도 중요**하다.

LOSS 함수를 W로 편미분하여 기울기를 구한다.

$$loss = (\hat{y} - y)^2 = (x * w - y)^2$$

$$w = w - \alpha \frac{\partial loss}{\partial w}$$

- **기울기가 양수일 때 : W의 값이 감소**

  $W := W - \alpha \times (\text{양수기울기})$

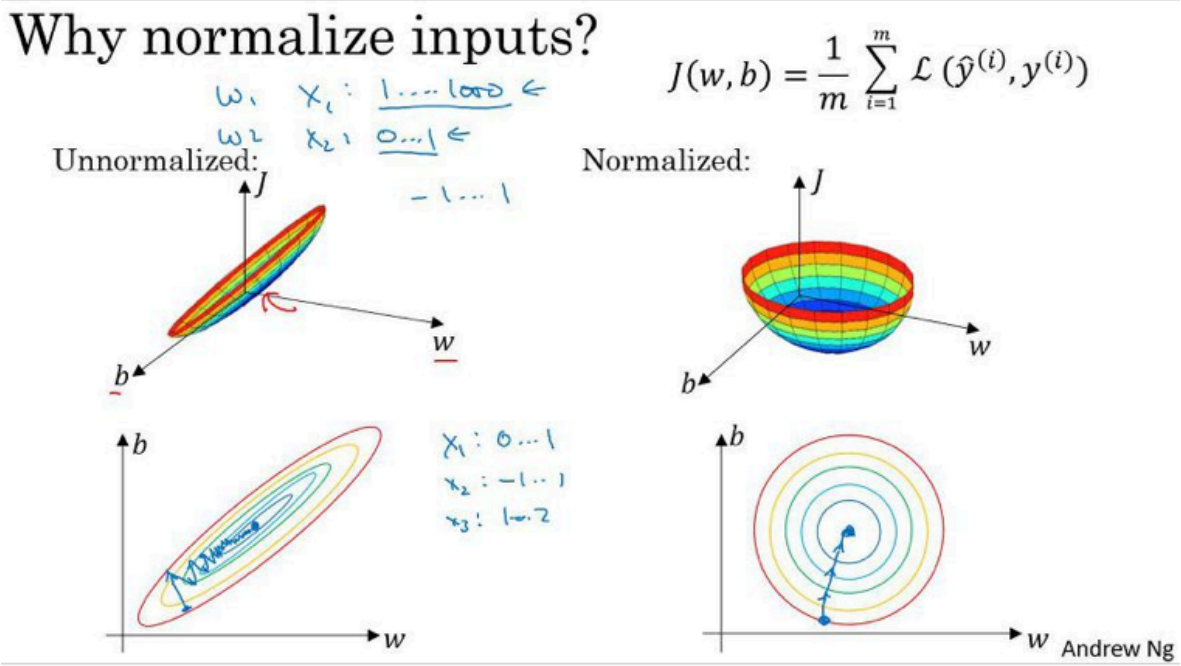
- **기울기가 음수일 때 : W의 값이 증가**

$$W := W - \alpha \times (\text{음수기울기}) = W + \alpha \times (\text{양수기울기})$$

3) 이미지 정규화

정규화(Data Normalization): 데이터의 범위를 사용자가 원하는 범위로 제한하는 것

이미지를 정규화하는 이유?
: 학습을 더 빨리하고 Local optimum에 빠지는 가능성을 줄이기 때문이다.

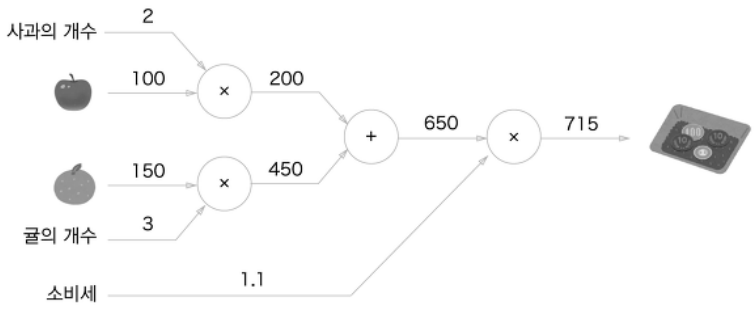


4) 오차역전파법(Backpropagation)

신경망 학습에서는 가중치 매개변수의 기울기를 미분을 이용해 구했다. 이는 간단하지만 시간이 오래 걸리는 단점이 있다.
오차역전파법(backpropagation)은 가중치 매개변수의 기울기를 효율적으로 계산하는 알고리즘이다.

4-1) 계산그래프

계산그래프: 계산과정을 그래프로 나타낸 것으로 노드와 엣지로 표현한다.
노드는 연산을 정의하며, 엣지는 데이터가 흘러가는 방향을 나타낸다.

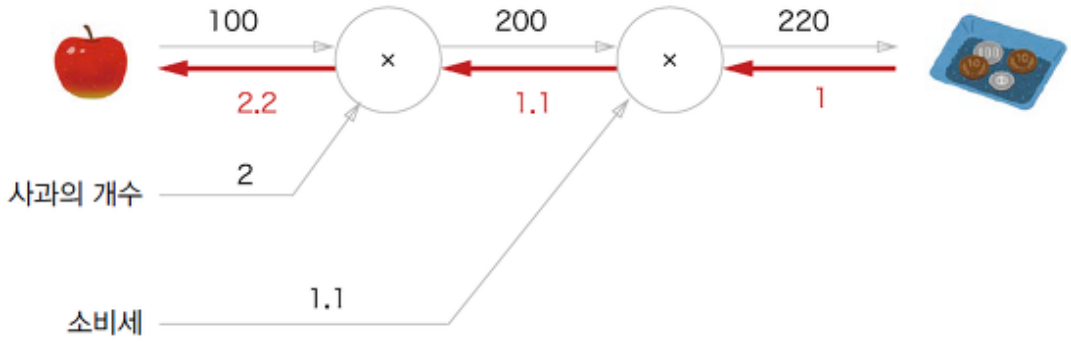


=> **순전파** (계산진행이 왼쪽에서 오른쪽으로 진행)

계산그래프의 특징은 **국소적 계산**을 통해 최종결과를 얻는 것이다. 즉, **자신과 직접 관계된 범위 내**에서만 계산이 이뤄진다.

계산그래프의 장점

- **국소적 계산**을 통해 각 노드의 계산에 집중하여 문제를 단순화할 수 있다.
- **역전파**를 통해 미분을 효율적으로 계산할 수 있다.

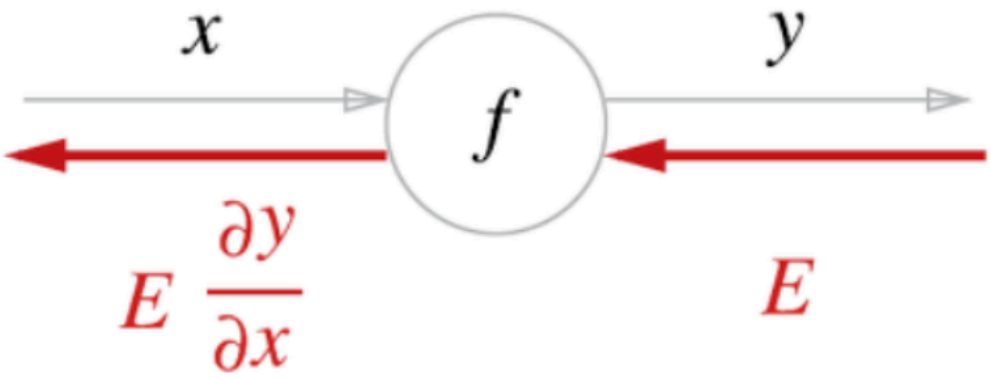


'사과 가격이 오르면 최종 금액에 어떠한 영향을 주는가'에 대해서 **사과 가격에 대한 지불 금액의 미분**을 구해 계산할 수 있다.

사과의 값을 x, 지불 금액을 L라 했을 때, $\frac{\partial L}{\partial x}$를 구하는 것이다.

4-2) 연쇄법칙(Chain Rule)

계산그래프의 역전파



역전파 계산 순서는 신호 E에 노드(f)의 국소적 미분을 곱한 후 엣지(edge)를 통해 다음 노드로 전달하는 것이다.
여기서 국소적 미분은 순전파 때의 y=f(x)에 대한 미분을 구하는 것이고, 이것은 x에 대한 y의 미분을 구한다는 의미이다.

연쇄 법칙
: 합성함수(여러 함수로 구성된 함수)의 미분은 합성 함수를 구성하는 각 함수의 미분의 곱으로 나타낼 수 있다.

$$t = x + y$$

$$z = t^2$$

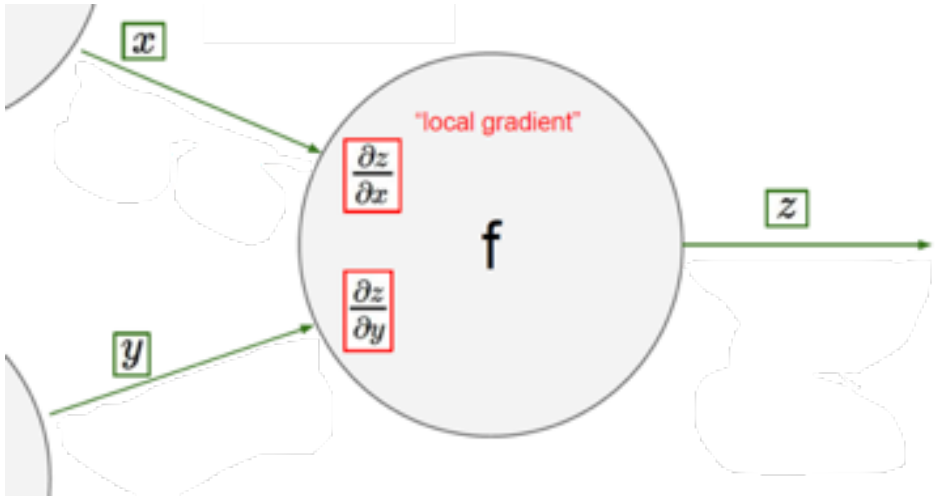
$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t}\frac{\partial t}{\partial x}$$

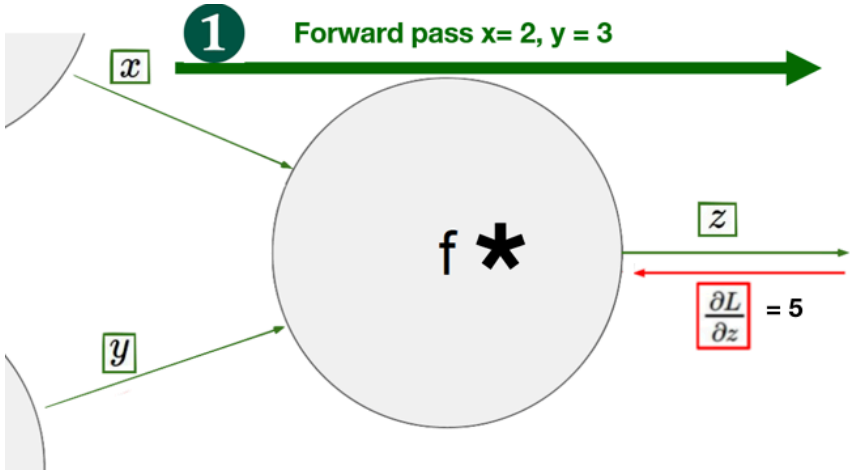
$$\frac{\partial z}{\partial t} = 2t$$

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

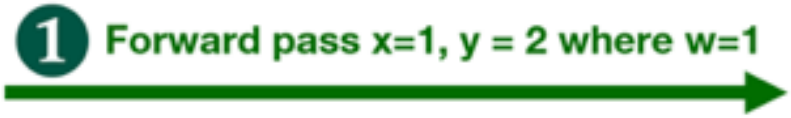
$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t}\frac{\partial t}{\partial x} = 2t \cdot 1 = 2(x + y)$$

연쇄법칙과 계산그래프

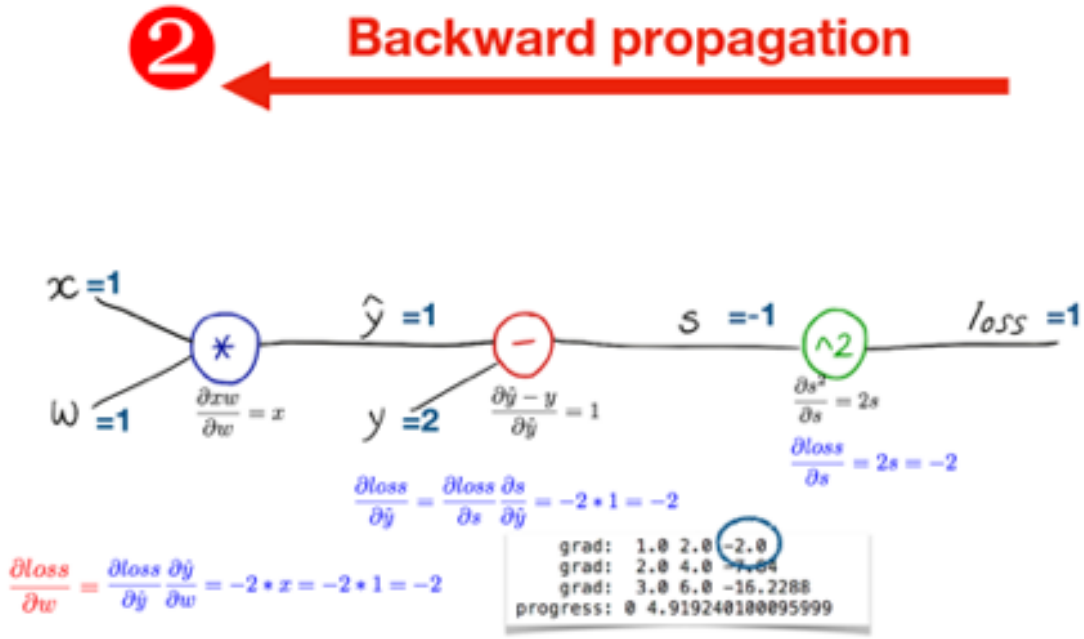

x
y
"local gradient"
f
z


1
Forward pass x= 2, y = 3
x
y
f *
z
= 5

4-3) 역전파


1
Forward pass x=1, y = 2 where w=1


x
w
*
ŷ
y
−
s
^2
loss



5) 이진분류예측(Logistic regression)

이진분류

: 둘 중 하나를 결정하는 문제

이진분류예측 예제)

1시간, 2시간 공부하면 불합격이고, 3시간 공부하면 합격일 때 4시간 공부했을 때는 합격일까 불합격일까?

Hours (x)	Points	fail/pass
1	2	0
2	4	0
3	6	1
4	?	?

이러한 그래프는 S자 형태로 표현된다. x와 y의 관계를 표현하기 위해서는 Wx+b와 같은 직선함수가 아니라

f(Wx+b)와 같은 S자 형태로 표현할 수 있는 함수가 필요하다. 이러한 함수가 **시그모이드 함수**이다.

5-1) 시그모이드 함수(Sigmoid function)

$$H(x) = sigmoid(Wx + b) = \frac{1}{1 + e^{-(Wx+b)}} = \sigma(Wx + b)$$

Meet Cross Entropy Loss

$$\hat{y} = x * w + b$$

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

$$\hat{y} = \sigma(x * w + b)$$

$$loss = \frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - y_n)^2$$

$$loss = -\frac{1}{N}\sum_{n=1}^{N} y_n \log \hat{y}_n + (1 - y_n)\log(1 - \hat{y}_n)$$

Hours (x)	Points	fail/pass
1	2	0
2	4	0
3	6	1
4	?	?