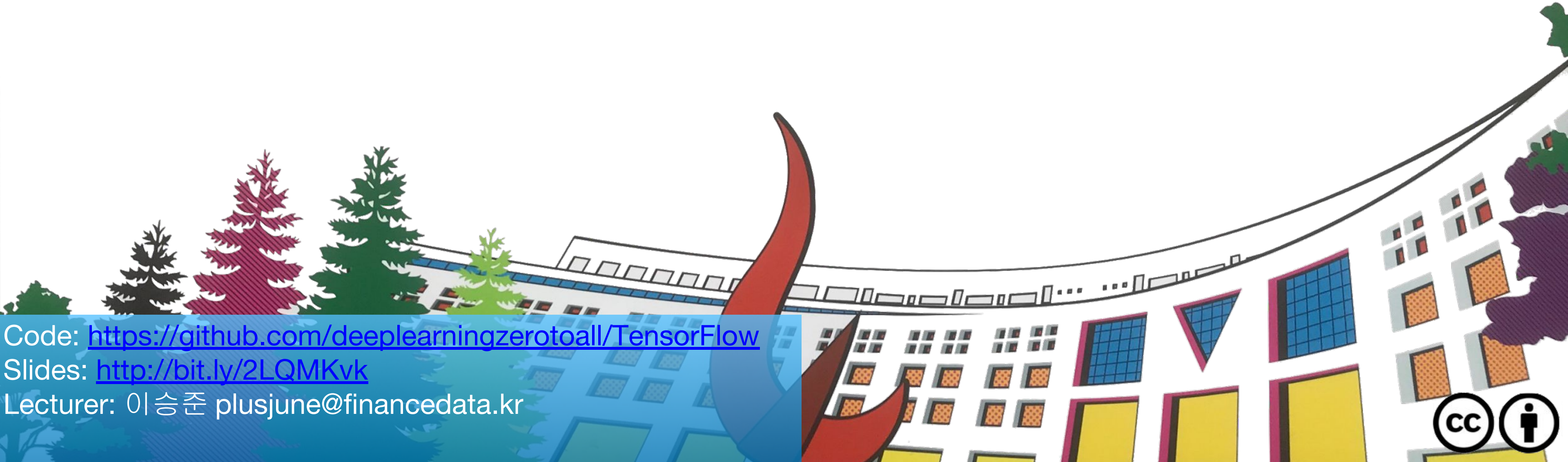
ML/DL for Everyone Season2

with TensorFlow

03 - How to minimize cost



Code: https://github.com/deeplearningzerotoall/TensorFlow
Slides: http://bit.ly/2LQMKvk
Lecturer: 이승준 plusjune@financedata.kr

Hypothesis and Cost

Hypothesis $$H(x) = Wx + b$$

Cost $$cost(W, b) = \frac{1}{m} \sum_{i=1}^{m} \left(H(x_i) - y_i\right)^2$$

Simplified hypothesis

Hypothesis $$H(x) = Wx$$

Cost $$cost(W) = \frac{1}{m} \sum_{i=1}^{m} (Wx_i - y_i)^2$$

What cost(W) looks like?

$$cost(W) = \frac{1}{m} \sum_{i=1}^{m} (Wx_i - y_i)^2$$

- W = 0, cost(W) = ?

x	y
1	1
2	2
3	3

What cost(W) looks like?

$$cost(W) = \frac{1}{m} \sum_{i=1}^{m} (Wx_i - y_i)^2$$

x	y
1	1
2	2
3	3

- W = 0, cost(W) = 4.67

$$\frac{1}{3}((0*1-1)^2 + (0*2-2)^2 + (0*3-3)^2))$$

What cost(W) looks like?

$$cost(W) = \frac{1}{m} \sum_{i=1}^{m} (Wx_i - y_i)^2$$

x	y
1	1
2	2
3	3

- W = 0, cost(W) = 4.67

  $$\frac{1}{3}((0*1-1)^2 + (0*2-2)^2 + (0*3-3)^2))$$

- W = 1, cost(W) = 0

  $$\frac{1}{3}((1*1-1)^2 + (1*2-2)^2 + (1*3-3)^2))$$

- W = 2, cost(W) = 4.67

  $$\frac{1}{3}((2*1-1)^2 + (2*2-2)^2 + (2*3-3)^2))$$

- W = 3, cost(W) = 18.67

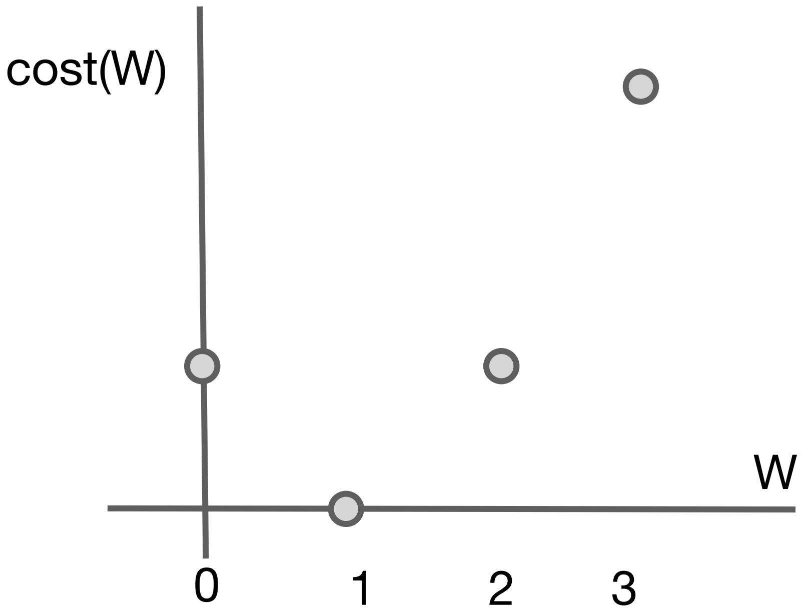
  $$\frac{1}{3}((3*1-1)^2 + (3*2-2)^2 + (3*3-3)^2))$$

What cost(W) looks like?

- $W = 0$, $\text{cost}(W) = 4.67$
- $W = 1$, $\text{cost}(W) = 0$
- $W = 2$, $\text{cost}(W) = 4.67$
- $W = 3$, $\text{cost}(W) = 18.67$

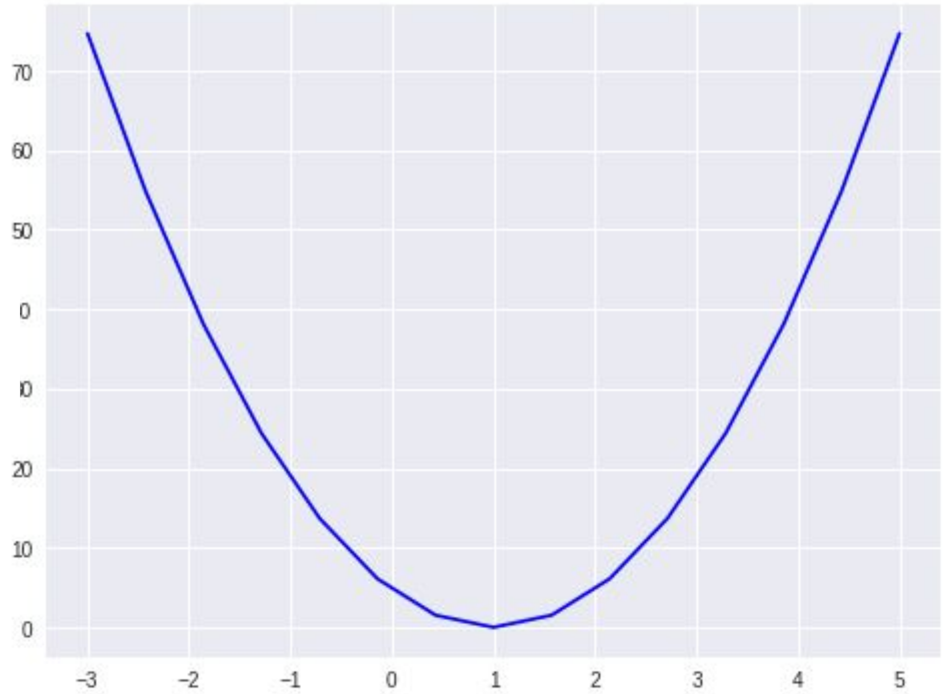
What cost(W) looks like?

- W = 0, cost(W) = 4.67
- W = 1, cost(W) = 0
- W = 2, cost(W) = 4.67
- W = 3, cost(W) = 18.67



What cost(W) looks like?

$$cost(W) = \frac{1}{m} \sum_{i=1}^{m} (Wx_i - y_i)^2$$

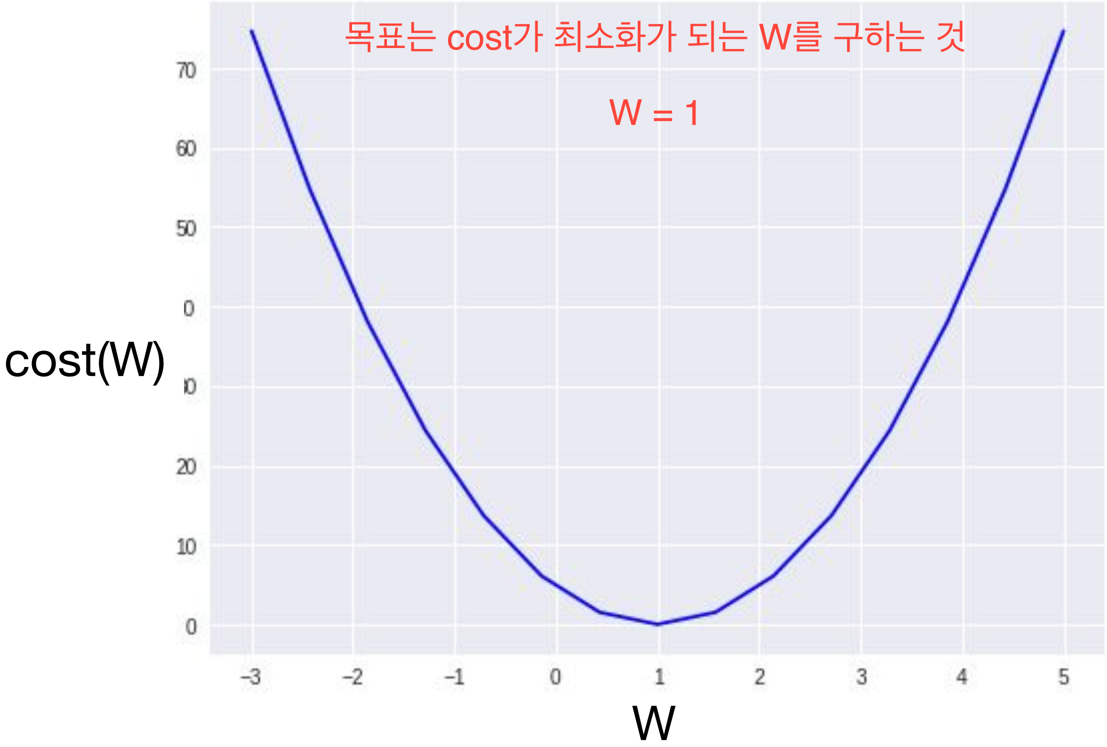


cost(W)

W

How to minimize cost?

$$cost(W) = \frac{1}{m}\sum_{i=1}^{m}(Wx_i - y_i)^2$$



컴퓨터가 최저점을 찾는 방법으로 널리 알려진 알고리즘

경사를 따라 내려가면서 최저점을 찾도록 설계된 알고리즘

Gradient descent algorithm

경사 하강법 / 경사 하강 알고리즘

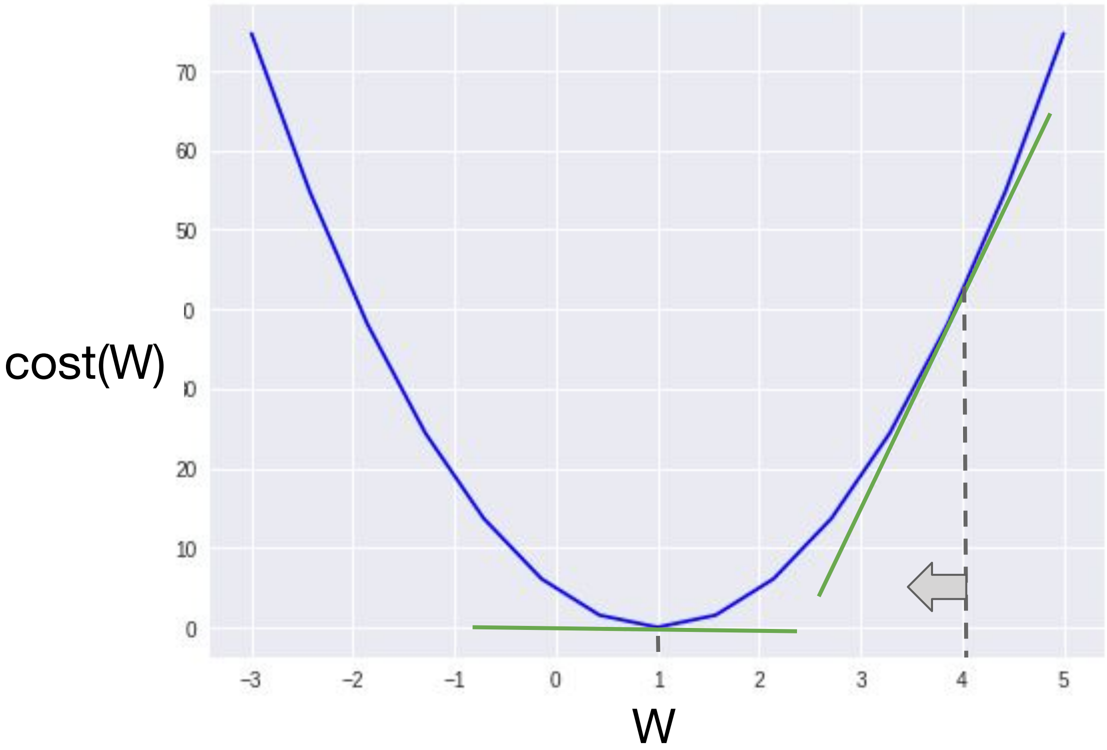
- Minimize cost function
- Gradient descent is used many minimization problems
- For a given cost function, cost (W, b), it will find W, b to minimize cost
- It can be applied to more general function: *cost (w1, w2, ...)*

변수가 여러 개 일 때도 사용 가능

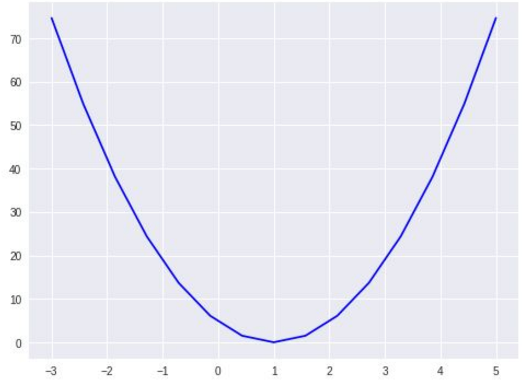
엔지니어링 문제는 최적화 문제이고, 손실을 최소화하거나 이득을 최대화하는 것

How it works?

How would you find the lowest point?



How it works?



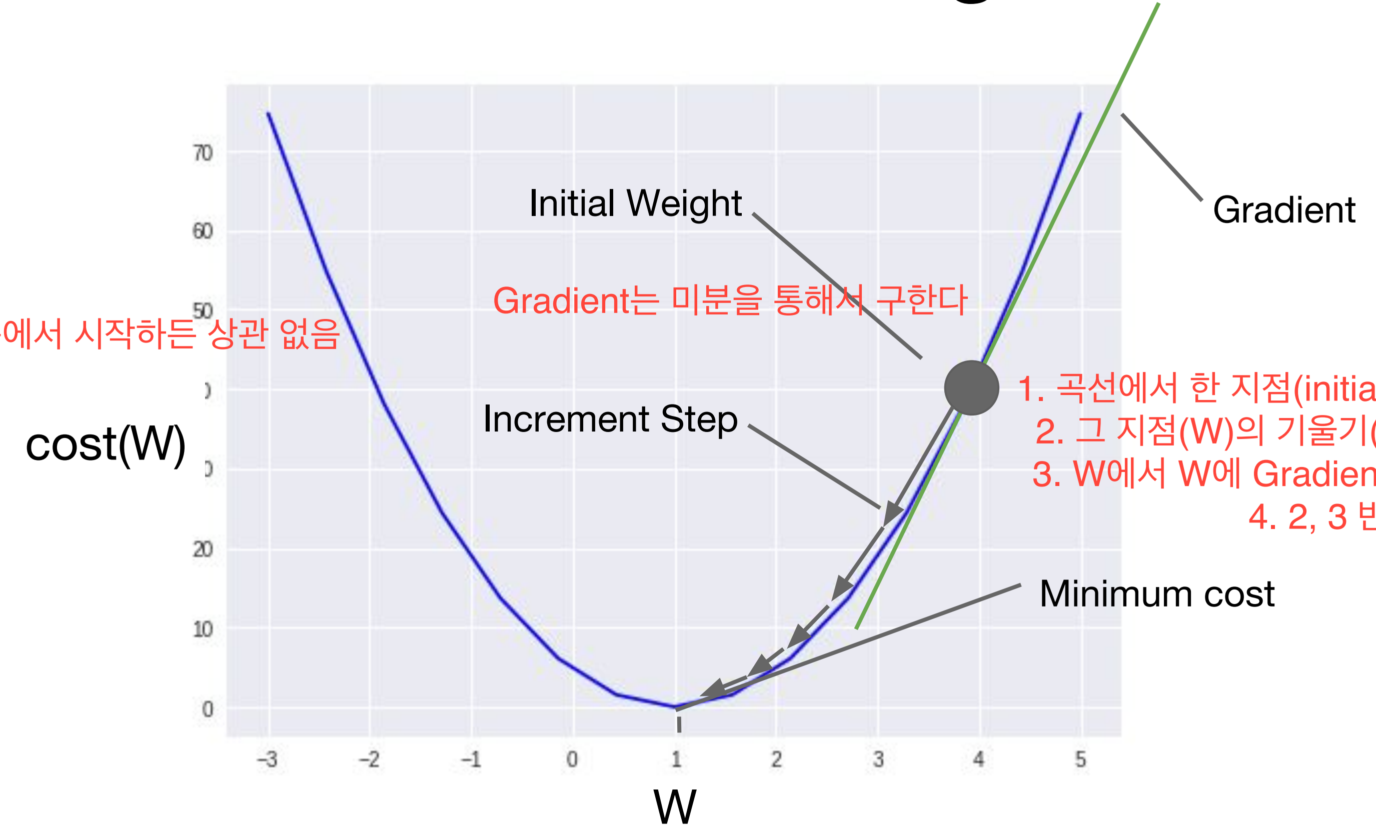
- Start with initial guesses
  - Start at 0,0 (or any other value)
  - Keeping changing W and b a little bit to try and reduce <u>cost(W, b)</u>
- Each time you change the parameters, you select the gradient which reduces cost(W, b) the most possible

  기울기 값을 구해서 cost가 최소화되는 방향으로 업데이트를 해나간다
- Repeat
- Do so until you converge to a local minimum

  최소점에 도달했다고 판단될 때까지
- Has an interesting property
  - Where you start can determine which minimum you end up

http://www.holehouse.org/mlclass/01_02_Introduction_regression_analysis_and_gr.html

Gradient descent algorithm



Gradient는 미분을 통해서 구한다

W를 왼쪽 오른쪽에서 시작하든 상관 없음

1. 곡선에서 한 지점(initial Weight)을 정하기
2. 그 지점(W)의 기울기(Gradient) 구하기
3. W에서 W에 Gradient를 곱한 값 빼주기
4. 2, 3 반복

최저점에 다가갈 수록 기울기 작아진다

Formal definition

전체 개수 m으로 나눴는데, 이게 2m, 4m이든 cost의 특성에는 영향을 주지 않는다

$$cost(W,b) = \frac{1}{m} \sum_{i=1}^{m} \left(H(x_i) - y_i\right)^2$$

$$\Downarrow$$

$$cost(W,b) = \frac{1}{2m} \sum_{i=1}^{m} \left(H(x_i) - y_i\right)^2$$

Formal definition

알파값에 따라 W값이 얼마나 빠르게 변할지 결정된다

알파값은 작은 상수로 learning rate - 우리가 구한 값을 얼마나 반영해서 W에서 뺄지 결정하는 배수같은 것

$$W := W - \alpha \frac{\partial}{\partial W} \frac{1}{2m} \sum_{i=1}^{m} (W(x_i) - y_i)^2$$

W에 대해서만 미분하겠다는 편미분

$$W := W - \alpha \frac{1}{2m} \sum_{i=1}^{m} 2(W(x_i) - y_i)x_i$$

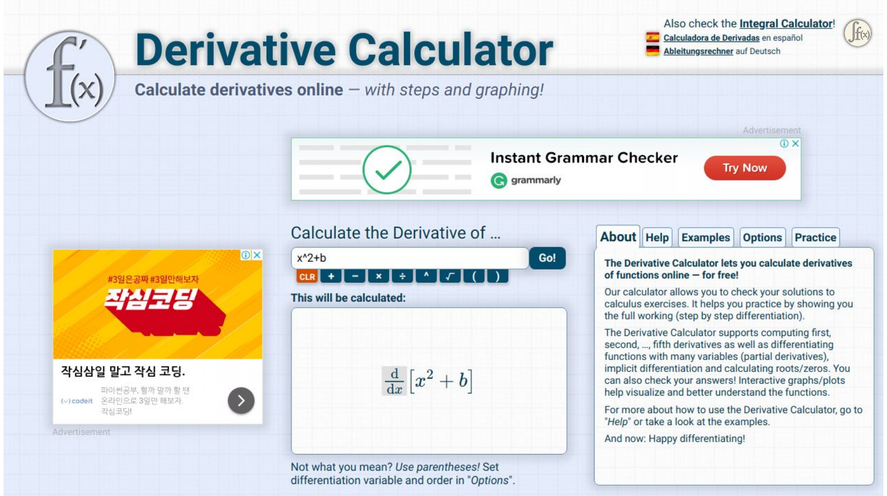
$$W := W - \alpha \frac{1}{m} \sum_{i=1}^{m} (W(x_i) - y_i)x_i$$

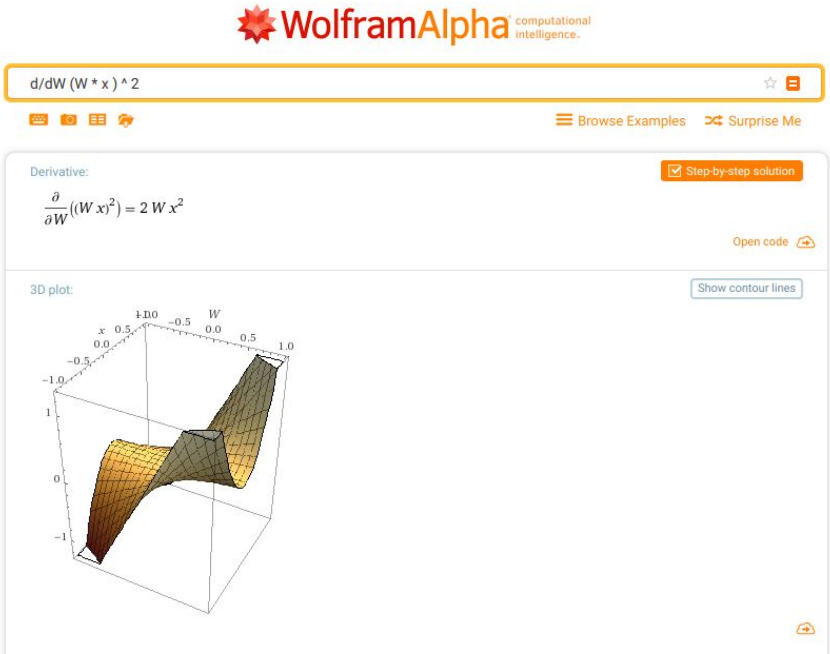
Formal definition

$$cost(W,b) = \frac{1}{2m}\sum_{i=1}^{m}\left(H(x_i) - y_i\right)^2$$

$$W := W - \alpha\frac{\partial}{\partial W}cost(W)$$

:= 는 '정의된다'는 의미


Derivative Calculator
Calculate derivatives online — with steps and graphing!
Also check the Integral Calculator!
Calculadora de Derivadas en español
Ableitungsrechner auf Deutsch
Advertisement
Instant Grammar Checker
grammarly
Try Now
#3일은공짜 #3일만해보자
작심코딩
작심삼일 말고 작심 코딩.
Advertisement
Calculate the Derivative of ...
x^2+b
Go!
CLR
This will be calculated:
$\frac{d}{dx}\left[x^2 + b\right]$
Not what you mean? Use parentheses! Set differentiation variable and order in "Options".
About
Help
Examples
Options
Practice
The Derivative Calculator lets you calculate derivatives of functions online — for free!
Our calculator allows you to check your solutions to calculus exercises. It helps you practice by showing you the full working (step by step differentiation).
The Derivative Calculator supports computing first, second, ..., fifth derivatives as well as differentiating functions with many variables (partial derivatives), implicit differentiation and calculating roots/zeros. You can also check your answers! Interactive graphs/plots help visualize and better understand the functions.
For more about how to use the Derivative Calculator, go to "Help" or take a look at the examples.
And now: Happy differentiating!


WolframAlpha computational intelligence.
d/dW (W * x) ^ 2
Browse Examples
Surprise Me
Derivative:
Step-by-step solution
$\frac{\partial}{\partial W}((W\,x)^2) = 2\,W\,x^2$
Open code
3D plot:
Show contour lines
x
W

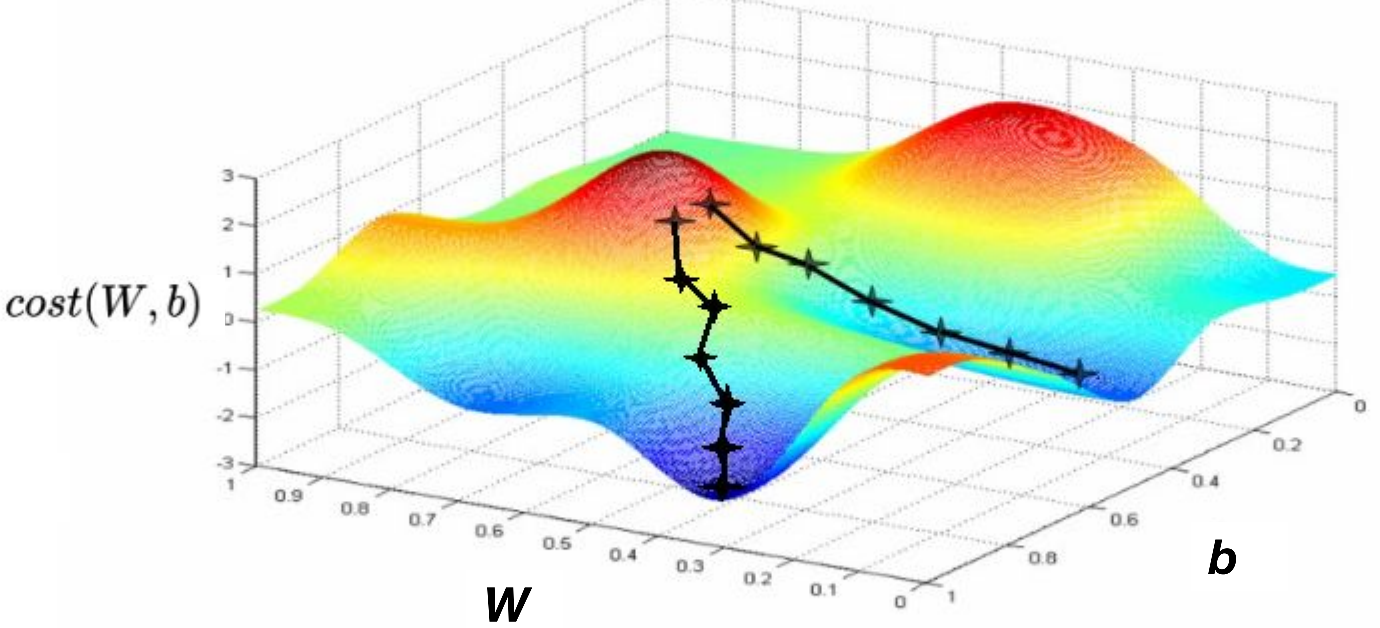
Gradient descent algorithm

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^{m} (W(x_i) - y_i) x_i$$

적절한 알파값을 지정하는 것도 과제

Convex function

어디서 시작하는지에 따라서 전체에서의 최저점을 찾지 못할 수도 있다



주변에서 가장 낮은 지점, 즉 기울기가 0인 지점을 local minimum
여기서는 지금 여러 개 있다
이런 상황에서는 gradient descent는 이런 상황에서 쓸 수 없다.

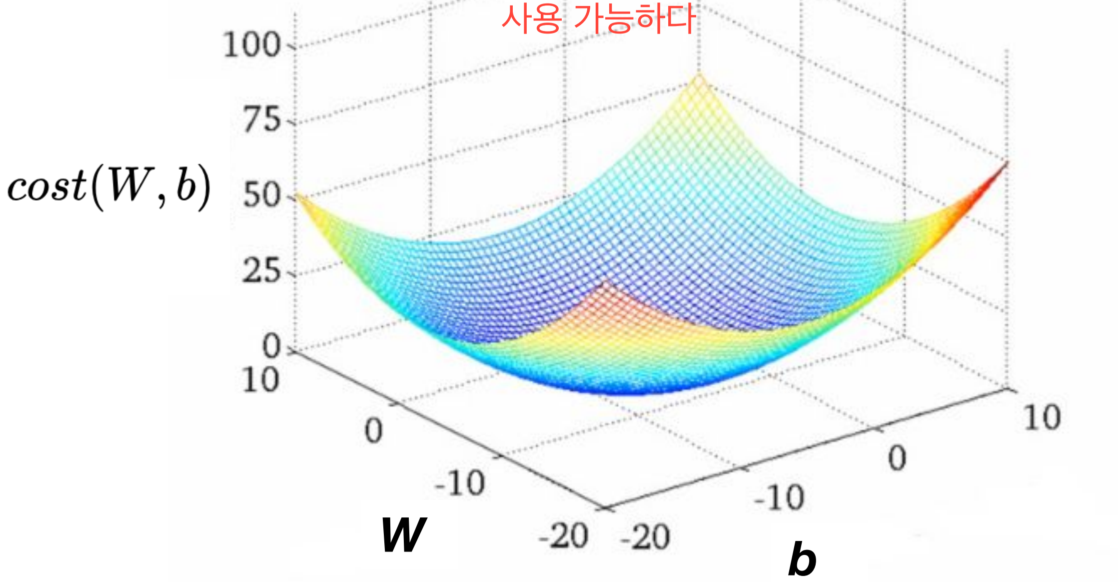
http://www.holehouse.org/mlclass/

Convex function

global minimum와 local minimum이 일치하는 함수

$$cost(W,b) = \frac{1}{m} \sum_{i=1}^{m} (H(x_i) - y_i)^2$$

cost function이 항상 convex function인 것은 아니지만,
convex function이라면 gradient descent를 사용해도 언제나 global minimum에 도착하는 것을 보장할 수 있기 때문에
사용 가능하다



What’s Next?

- Multi-Variable Linear regression