Lecture 3

How to minimize cost

이번 강의는 cost 함수를 어떻게 최소화해서 최종적으로 Linear Regression 학습을 완성하는지 설명

Sung Kim <hunkim+mr@gmail.com>

Acknowledgement

- Andrew Ng's ML class
  - https://class.coursera.org/ml-003/lecture
  - http://www.holehouse.org/mlclass/ (note)
- Convolutional Neural Networks for Visual Recognition.
  - http://cs231n.github.io/
- Tensorflow
  - https://www.tensorflow.org
  - https://github.com/aymericdamien/TensorFlow-Examples

Hypothesis and Cost

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

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

가설값 실제값

Simplified hypothesis

$$H(x) = Wx$$

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

What *cost(W)* looks like?

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

x	Y
1	1
2	2
3	3

- *W*=1, *cost(W)*=?

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=1, cost(W)=0

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

- W=0, cost(W)=4.67

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

- W=2, cost(W)=?

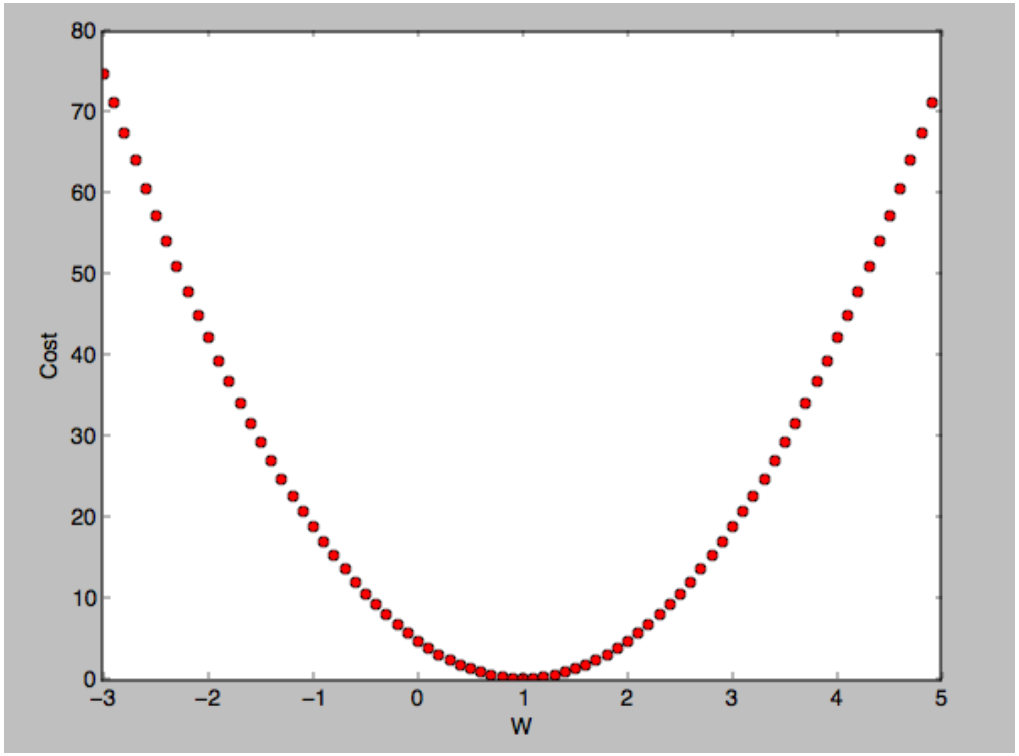
  1/3((2 * 1 - 1)^2 + (2 * 2 - 2)^2 + (2 * 3 - 3)^2)

What *cost(W)* looks like?

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

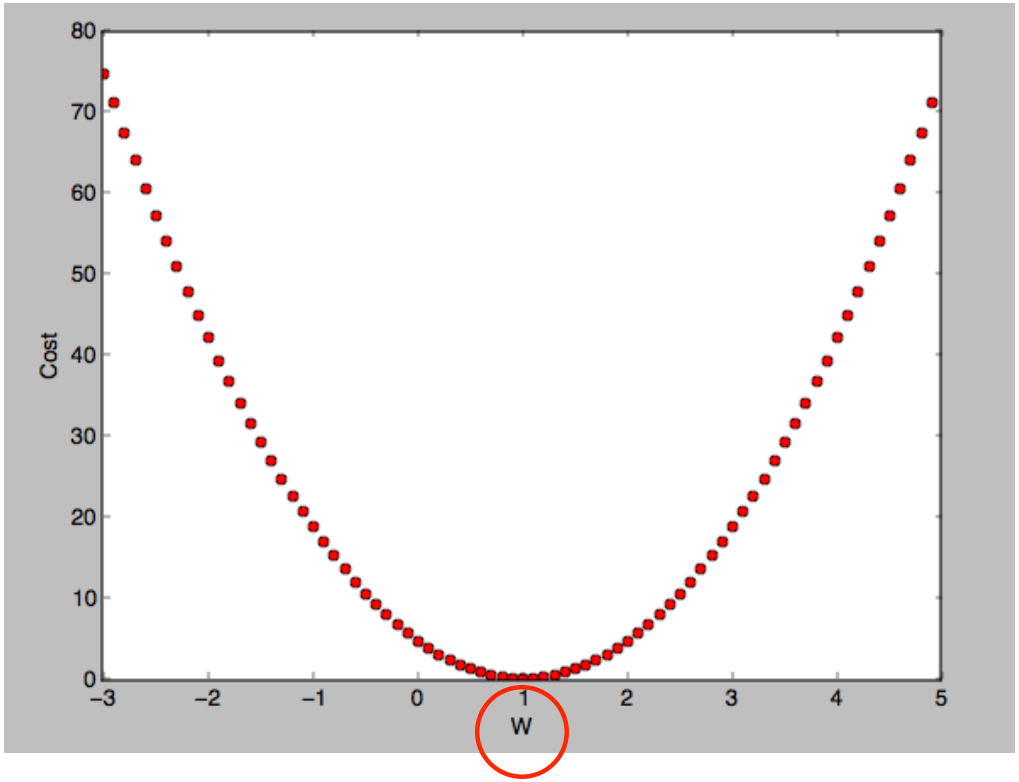
What *cost(W)* looks like?

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



How to minimize cost?

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



cost가 최소화하는 점을 찾아내는 것이 목적

경사 하강법 / 경사 하강 알고리즘

경사를 따라 내려가면서 최저점을 찾도록 설계된 알고리즘

Gradient descent algorithm

경사 내려감(=하강)

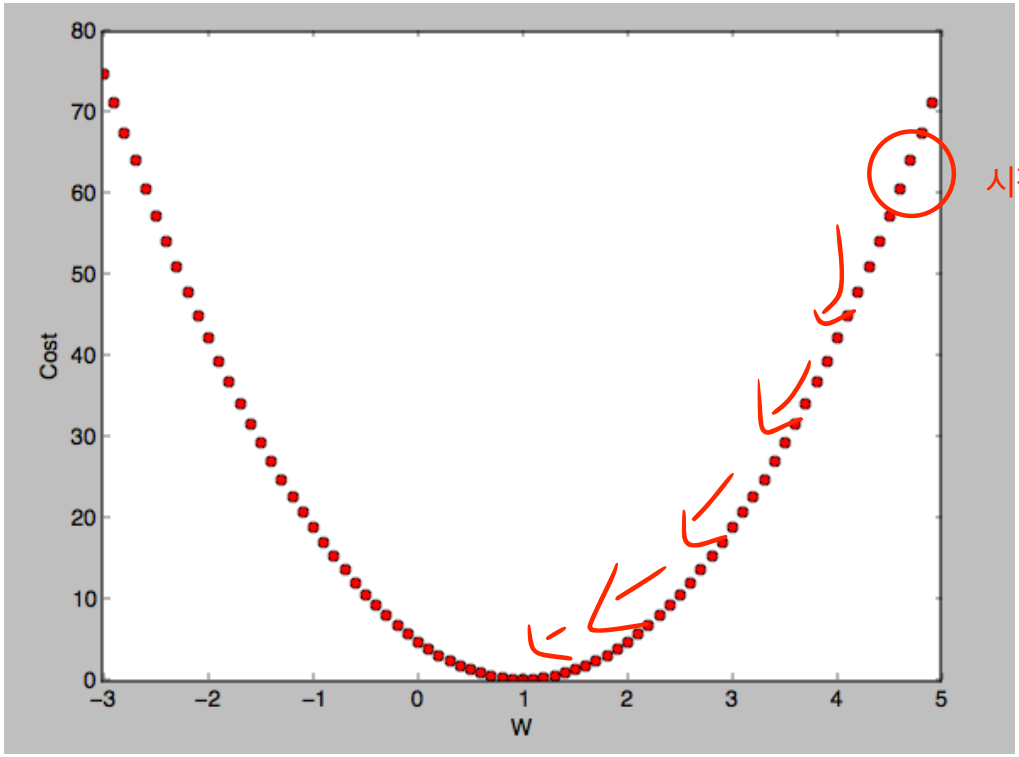
- Minimize cost function
- Gradient descent is used many minimization problems
- For a given cost function, *cost (W, b)*, it will find *W, b* to minimize cost
- It can be applied to more general function: *cost (w1, w2, …)*

변수가 여러 개 일 때도 사용 가능

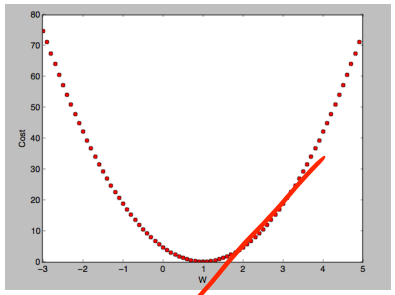
How it works?

How would you find the lowest point?

시작점에서 경사도를 따라서 한 발자국 씩 내려가는 것



How it works?



한 지점에서의 경사도는 미분

- Start with initial guesses
  - Start at 0,0 (or any other value) 어떤 점에서 시작하든 상관 없다
  - Keeping changing W and b a little bit to try and reduce cost(W, b)

W를 조금 변경

- 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

Formal definition

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

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

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

수식을 간단하게 하기 위해서 2m을 사용

Formal definition

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

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

learning rate

Formal definition

알파값에 따라 W값이 얼마나 빠르게 변할지 결정된다

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

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

W에 대해서만 미분하겠다는 편미분

$$W := W - \alpha \frac{1}{2m} \sum_{i=1}^{m} 2(Wx^{(i)} - y^{(i)})x^{(i)}$$

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^{m} (Wx^{(i)} - y^{(i)})x^{(i)}$$

Derivative Calculator

Calculate derivatives online
— with steps and graphing!

Also check the **Integral Calculator**!
Ableitungsrechner auf Deutsch

Hello there!

Was this calculator helpful to you? Then I would highly appreciate **small donations** via PayPal:

… or use **this link for shopping on Amazon**, without affecting your order.

Thank you!

Calculate the Derivative of …

(xa-y)^2 **Go!**

This will be calculated:

$$\frac{\mathrm{d}}{\mathrm{d}x}\left[(xa-y)^2\right]$$

Not what you mean? *Use parentheses!* Set differentiation variable and order in "*Options*".

About | **Help** | **Examples** | **Options**

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. 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!

Recommend this Website

If you like this website, then please support it by

Result

Done! See the result further below.

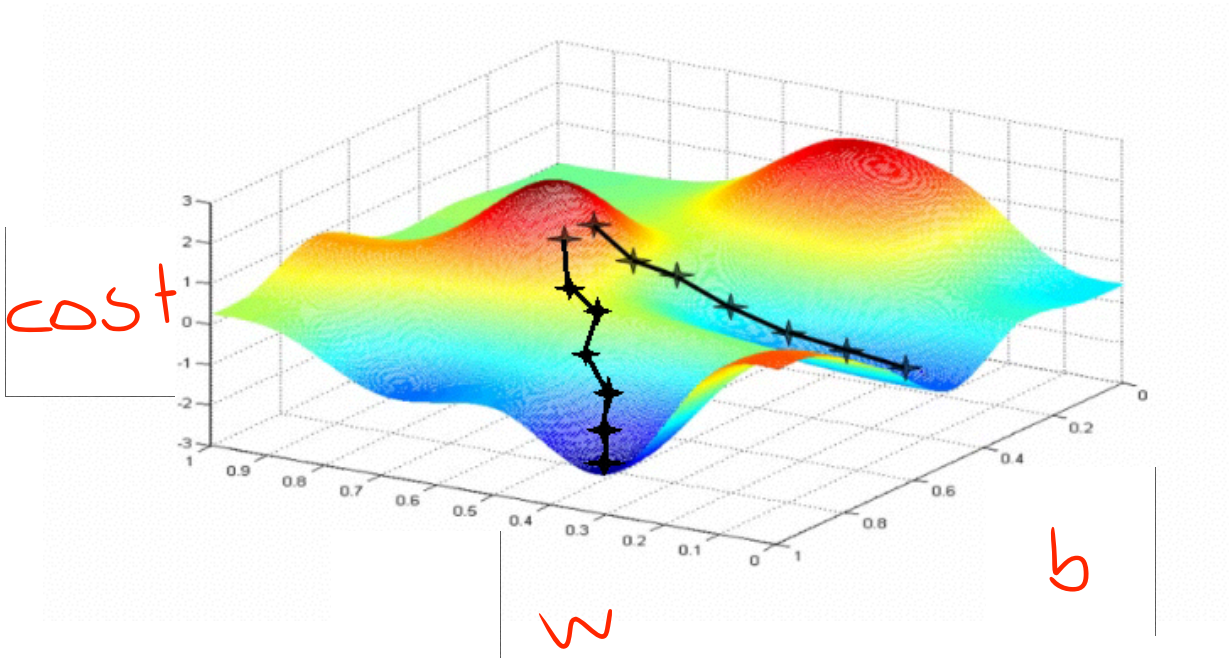
Gradient descent algorithm

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

:= 는 '정의된다'는 의미

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.

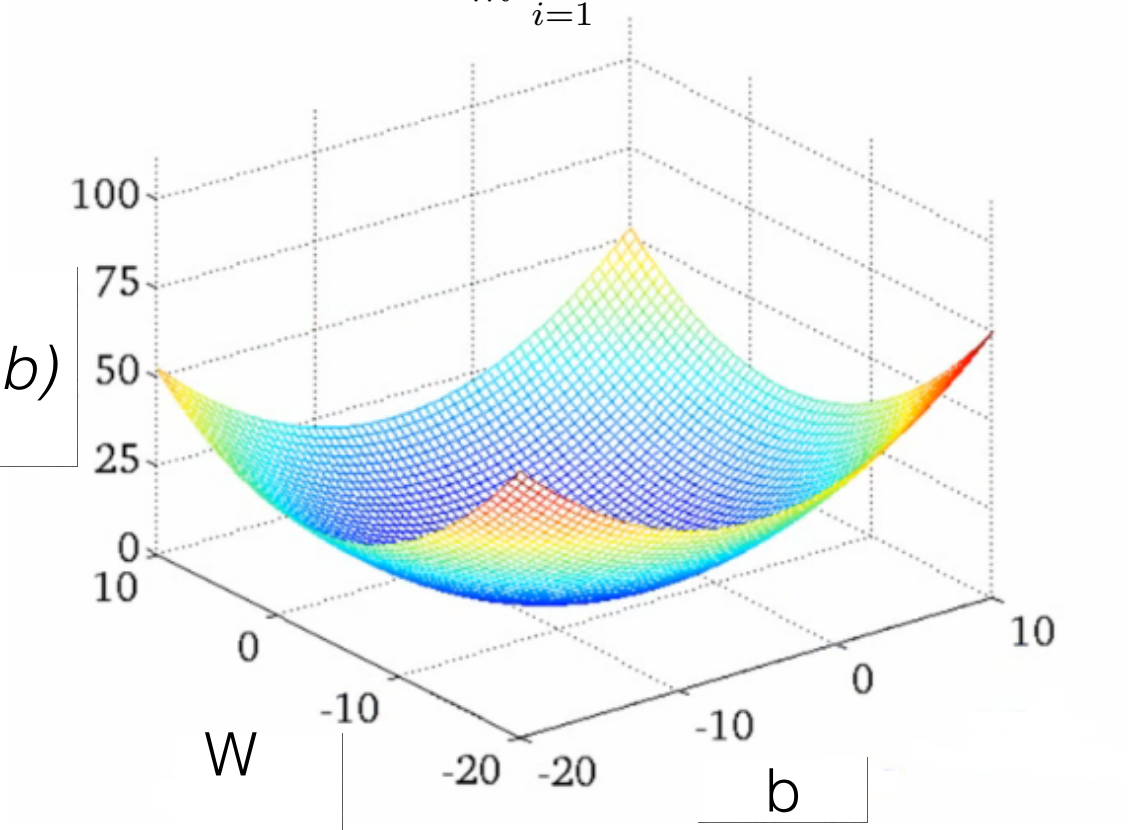
www.holehouse.org/mlclass/

이런 경우는 어느 지점에서 시작하든 gradient decsent 알고리즘이 최저점을 찾는다.

Convex function

알고리즘을 적용하기 전에 cost 함수의 모양이 convex function이 되는지 확인하기

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



www.holehouse.org/mlclass/

Next

Multivariable logistic regression