모두의 딥러닝 (Deep learning)

Nambeom Kim (nbumkim@gmail.com)

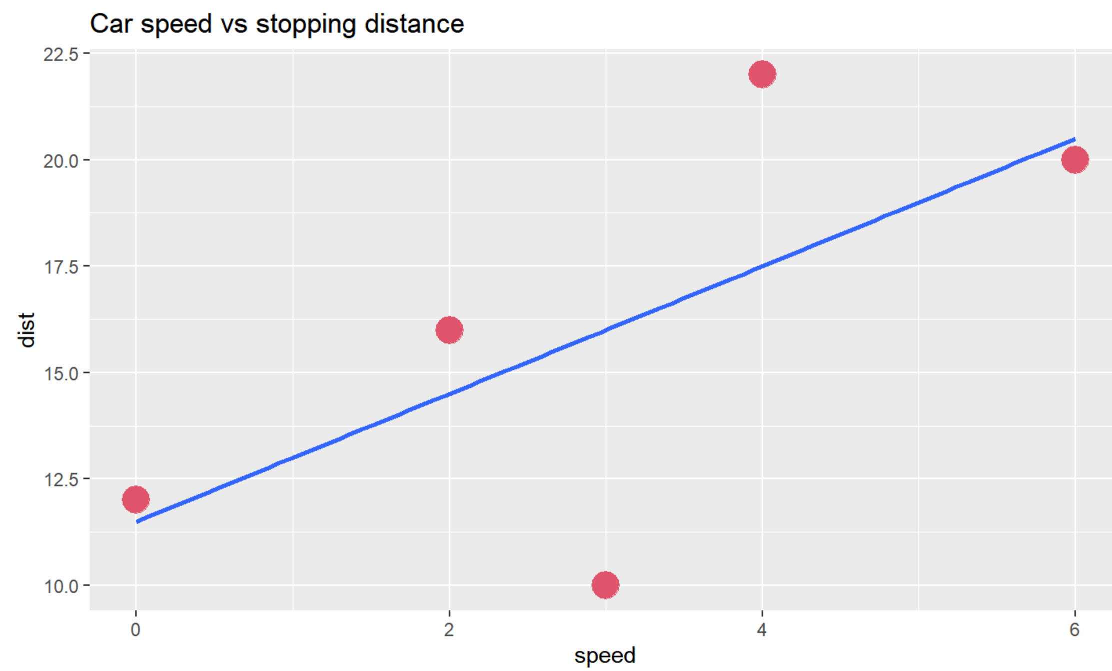
Gradient descent (GD)

The concept of gradient descent (GD) algorithm

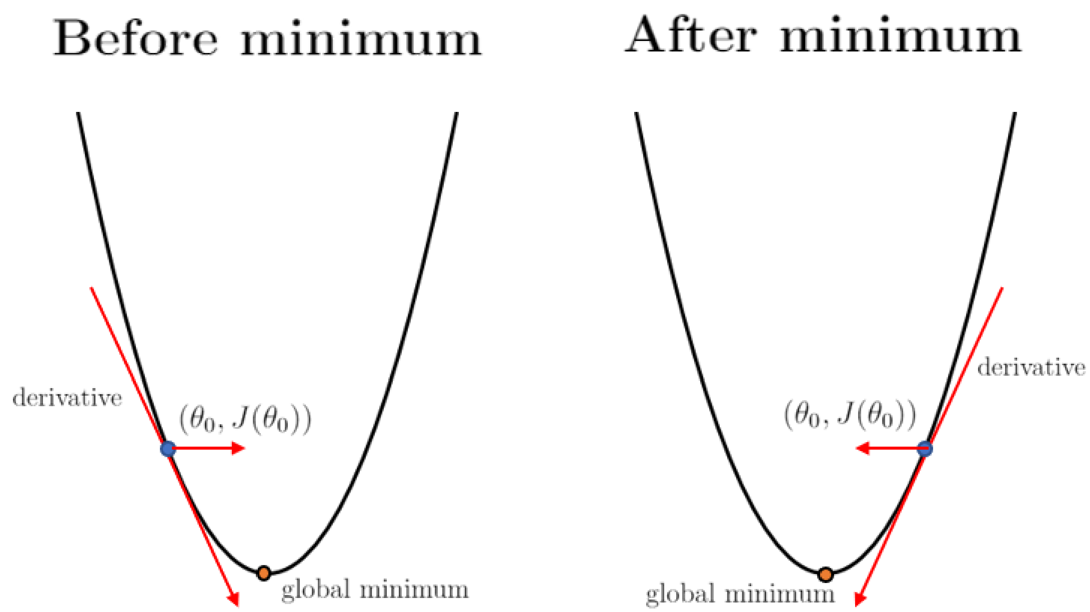
```
# A tibble: 5 x 2
  speed  dist
  <dbl> <dbl>
1     0    12
2     2    16
3     3    10
4     6    20
5     4    22
```

```
(Intercept)       speed
       11.5         1.5
```

```
[1] "Sum of squard error =  59"
```



Gradient descent (GD) algorithm

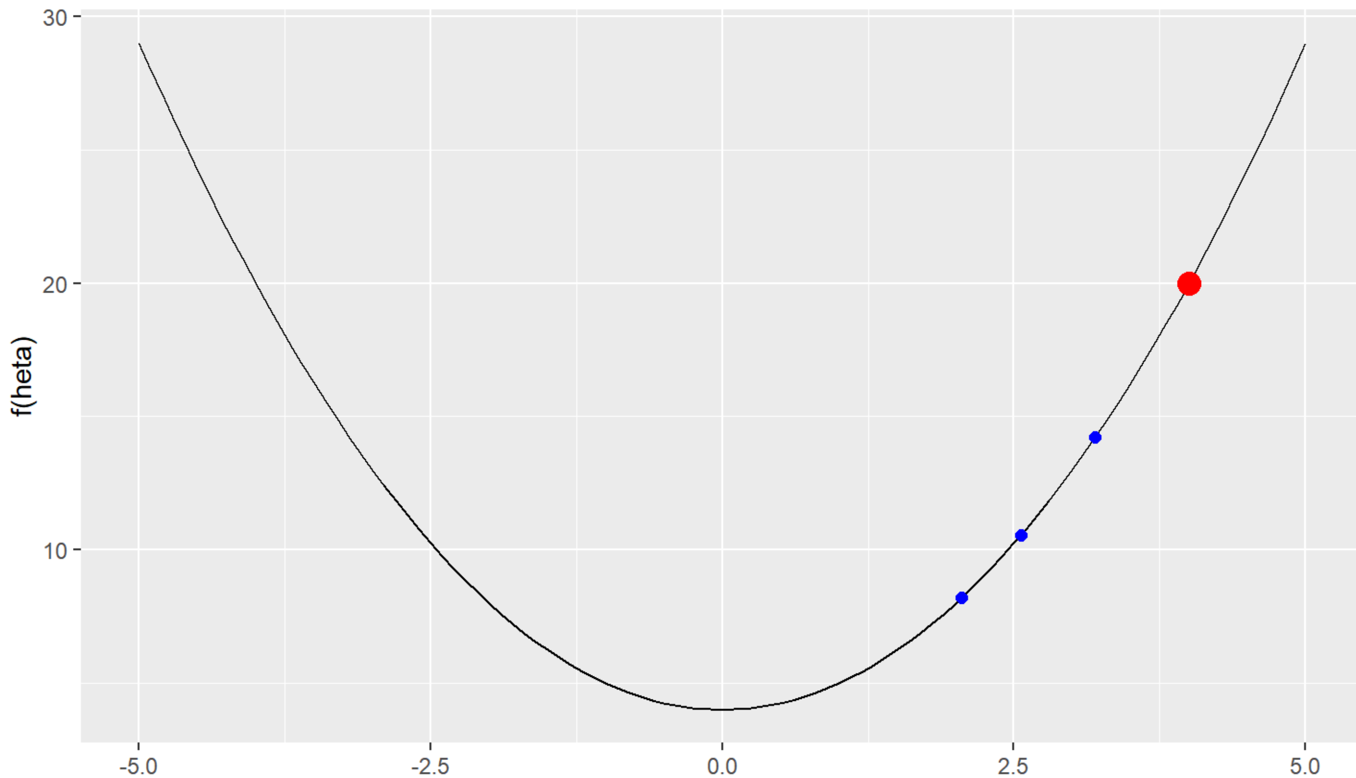


Since the derivative is negative,
if we subtract the derivative from θ_0,
it will increase and go closer the minimum.

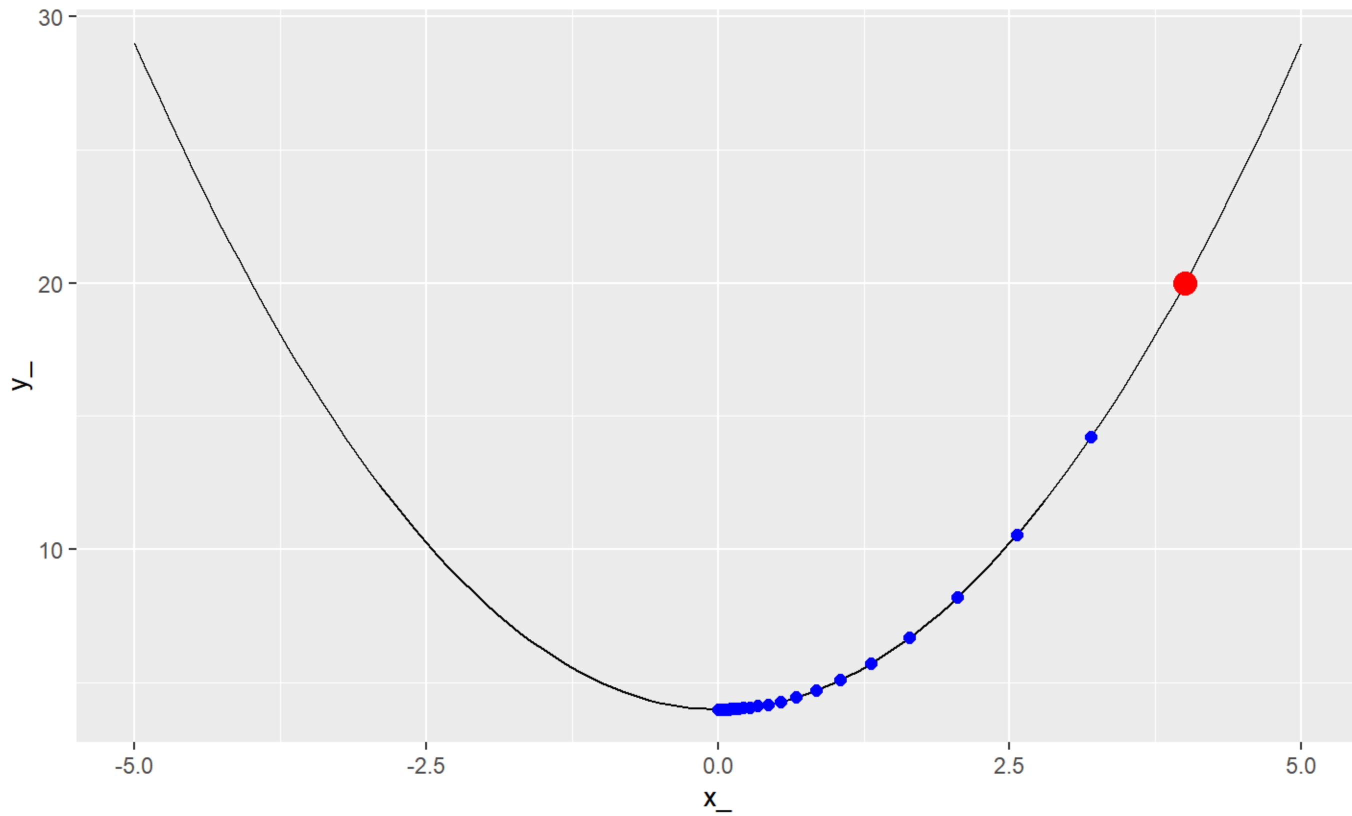
Since the derivative is positive,
if we subtract the derivative from θ_0,
it will decrease and go closer the minimum.

https://blog.goodaudience.com/gradient-descent-for-linear-regression-explained-7c60bc414bdd

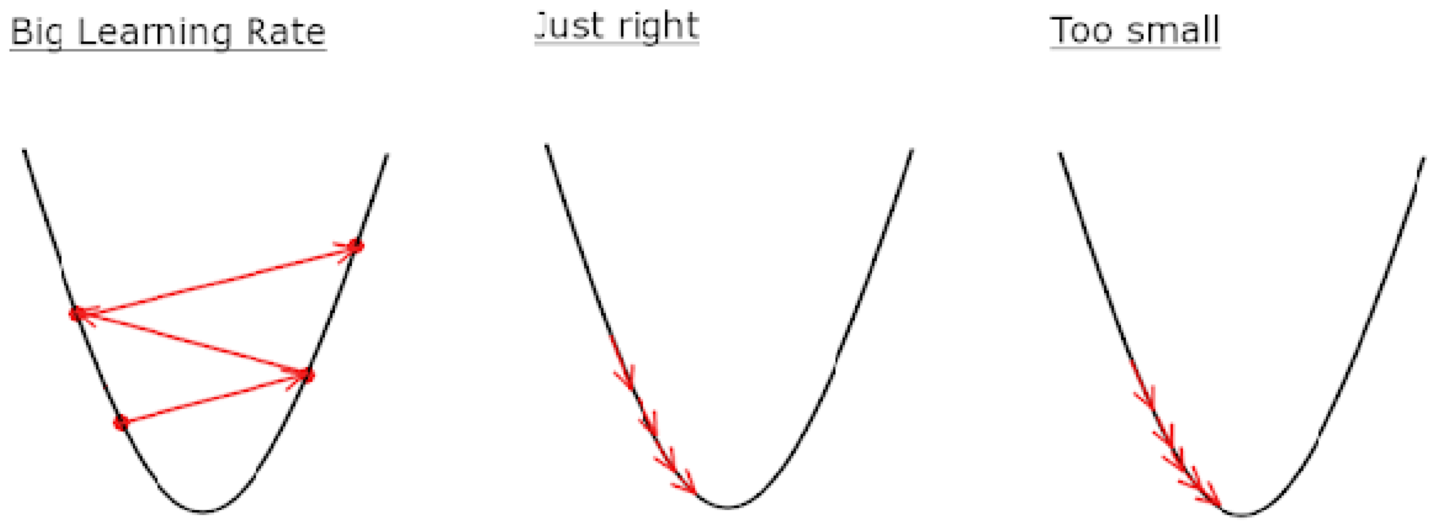
Gradient descent plot (iteration = 3)



Gradient descent plot (iteration = 30)

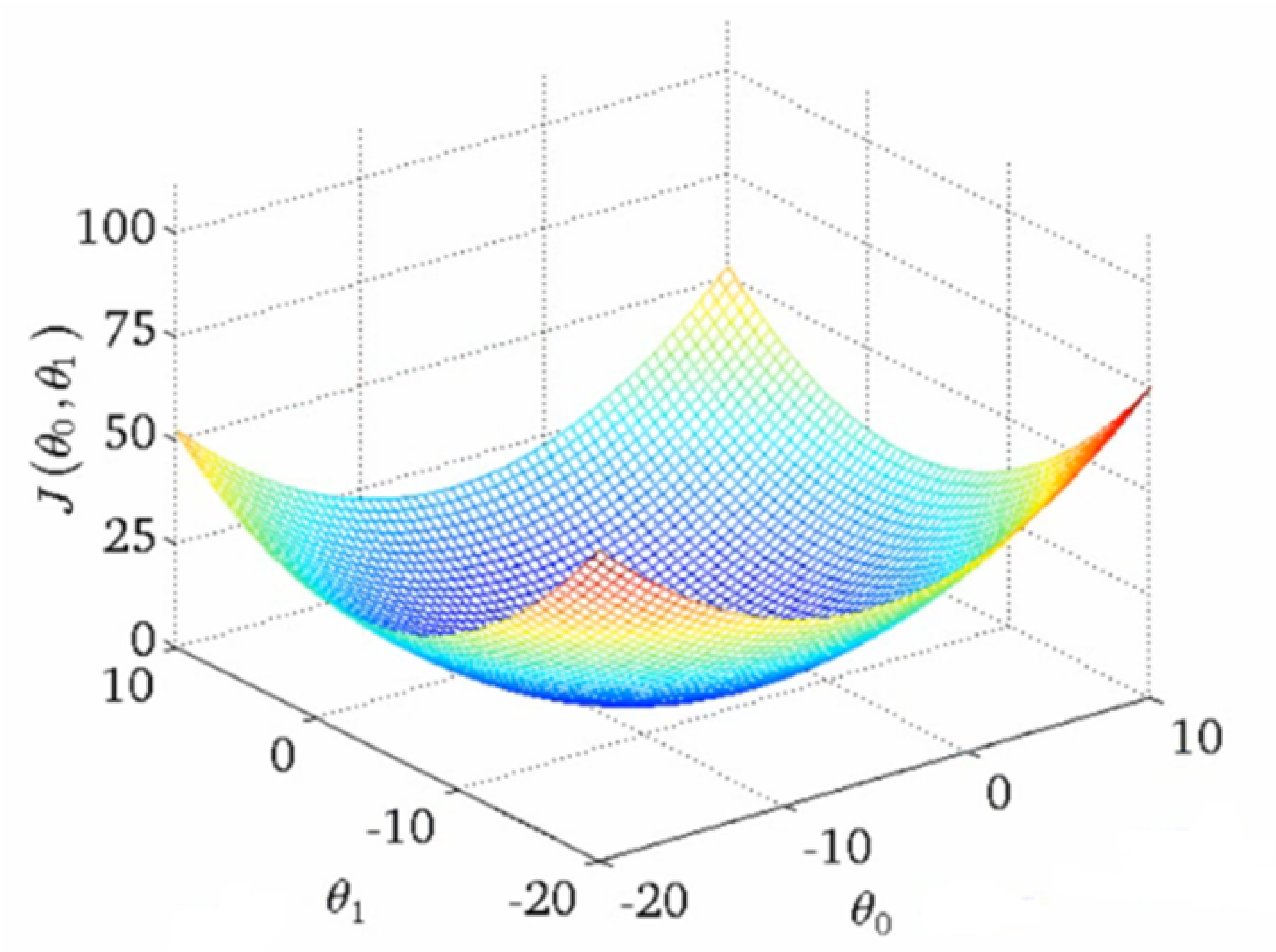


Step size (Learning rate) problem



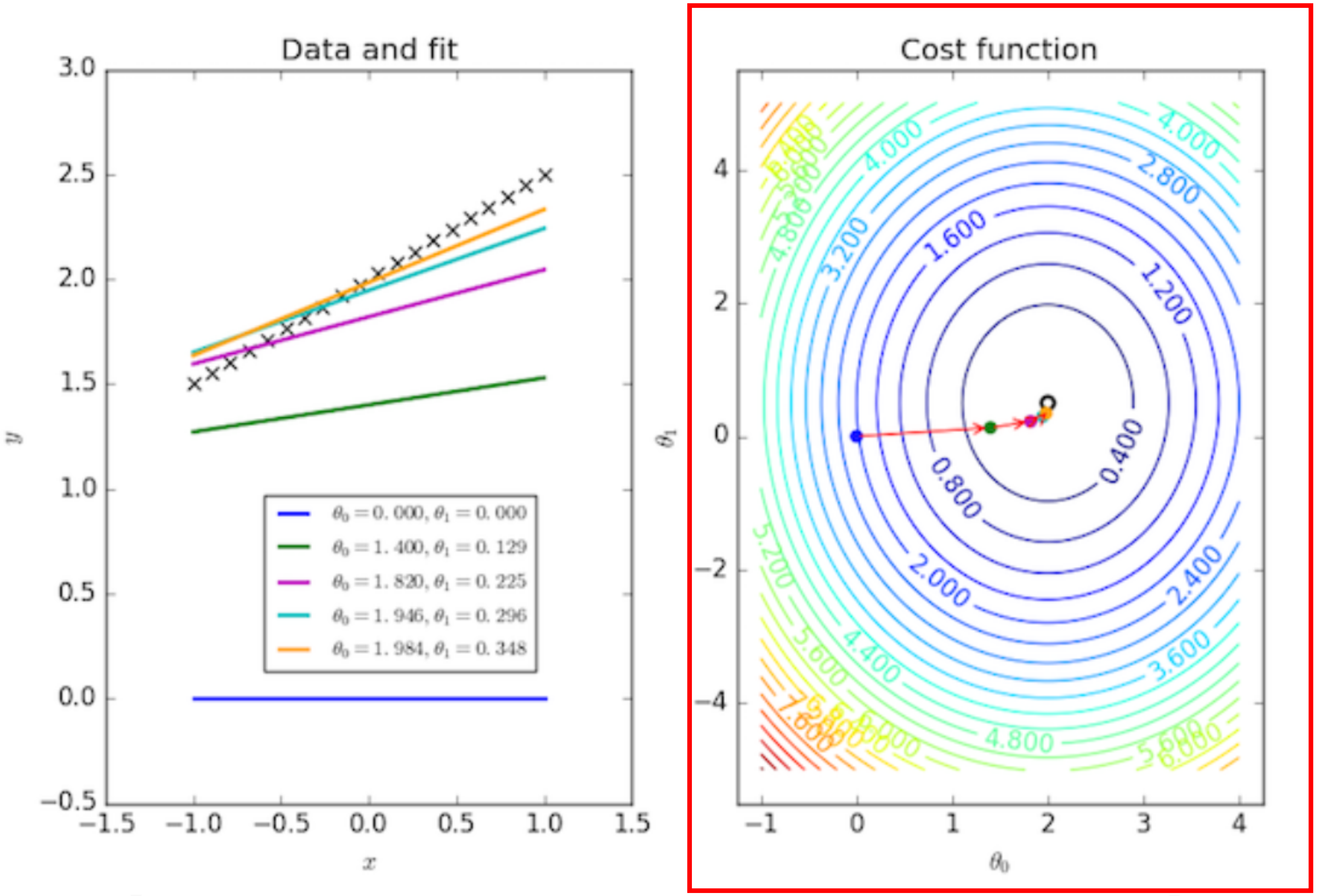
https://medium.com/analytics-vidhya/gradient-descent-why-and-how-e369950ae7d3

High dimensional GD



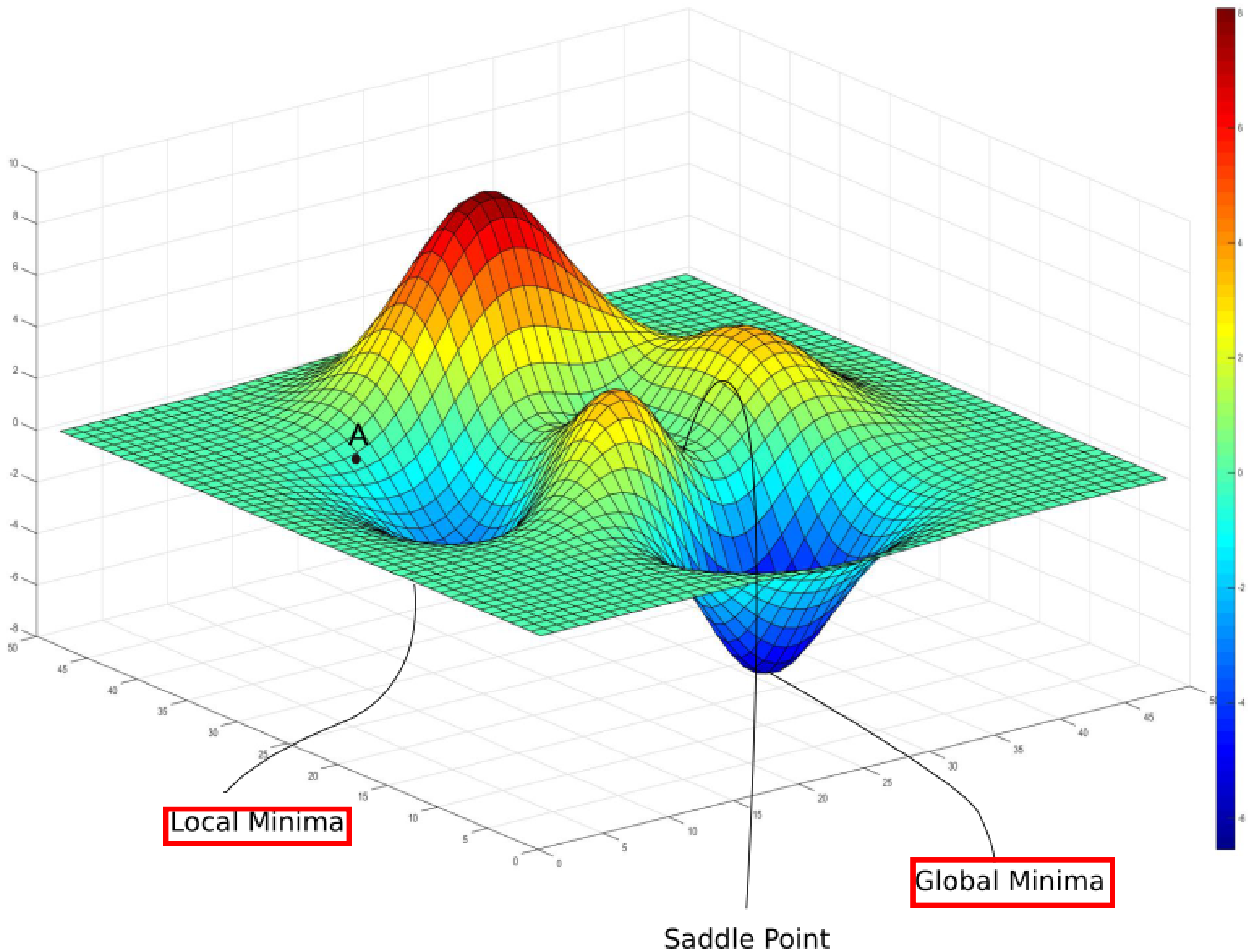
https://medium.com/analytics-vidhya/gradient-descent-why-and-how-e369950ae7d3

Optimizing parameter trajectory of GD algorithm



https://scipython.com/blog/visualizing-the-gradient-descent-method/

Local and global minina



https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/

Derivatives of cost function of logistic regression

Let's see the cases of cost function of logistic regression
This equation does not have a closed-form solution

$$\boxed{J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y_i log(h_\theta(x_i)) + (1-y_i)log(1-h_\theta(x_i))}$$

$where,\ h_\theta(x_i) = \frac{1}{1+e^{-\theta x}},\ \ y \in 0, 1$

Gradient descent (GD) algorithm of sigmoid function

- Objective (cost) function =
  $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2$
  $= \frac{1}{2m}\sum_{i=1}^{m}(y_i - h_\theta(x_i))^2$

- Parameter update :
  수렴한다.
  Repeat until convergence {

$$\boxed{\theta_j^{(n+1)} = \theta_j^{(n)} - \gamma \frac{\partial}{\partial \theta_j} J(\theta^{(n)})}$$

}