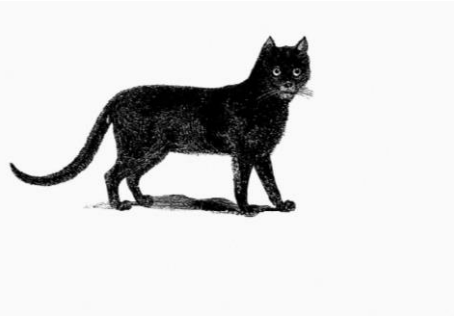
CNU 데이터 분석 교육



9th lecture

"Linear Analysis-3"

2021 - 11 - 14

수학과 최명수

지난시간?

1. Loss function
   - The key to determining the performance of a model.
   - SMnP, OD, log cosh

오늘은 무엇을?

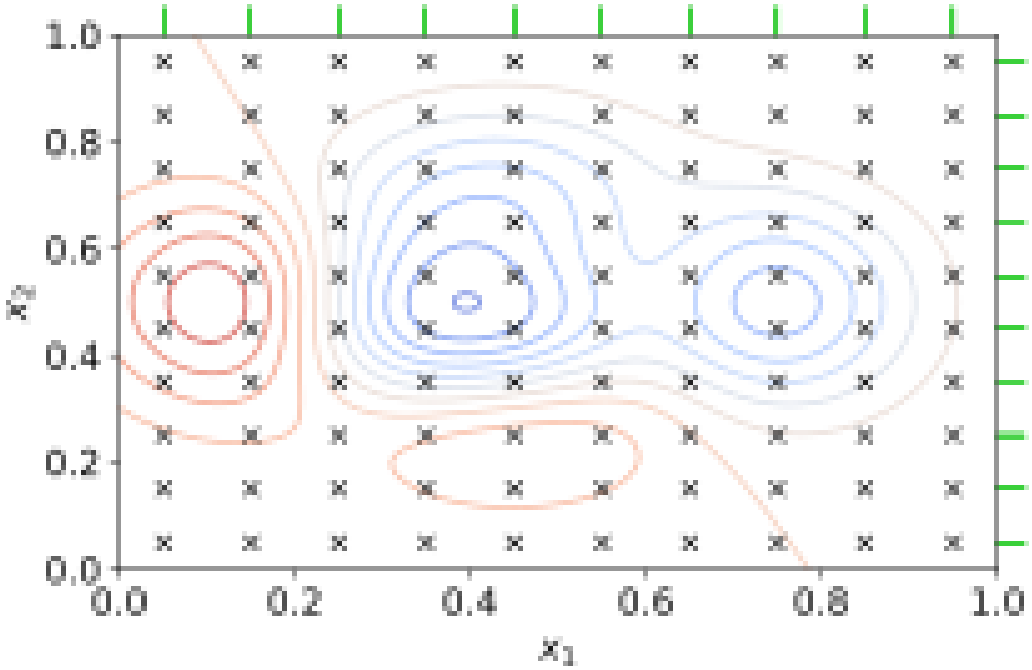
1. Loss function을 최소화 하는 parameter를 찾는 방법?

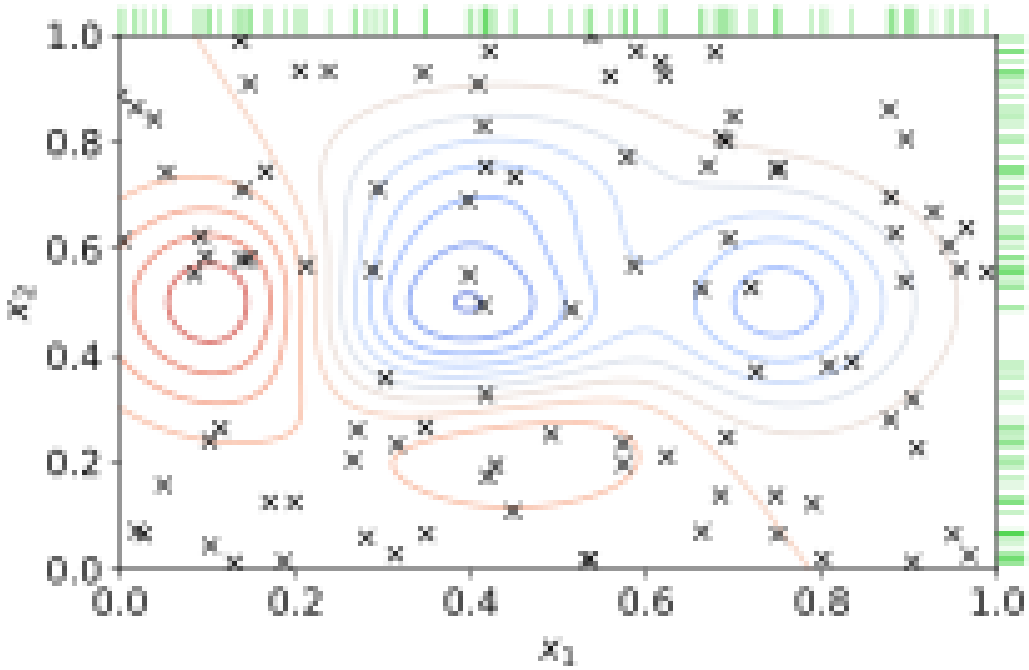
   - Gradient descent method
   - Calculus analysis

2. Linear regression model의 성능을 평가할 수 있는 방법?

Grid search

꼭 격자여야 하는가?



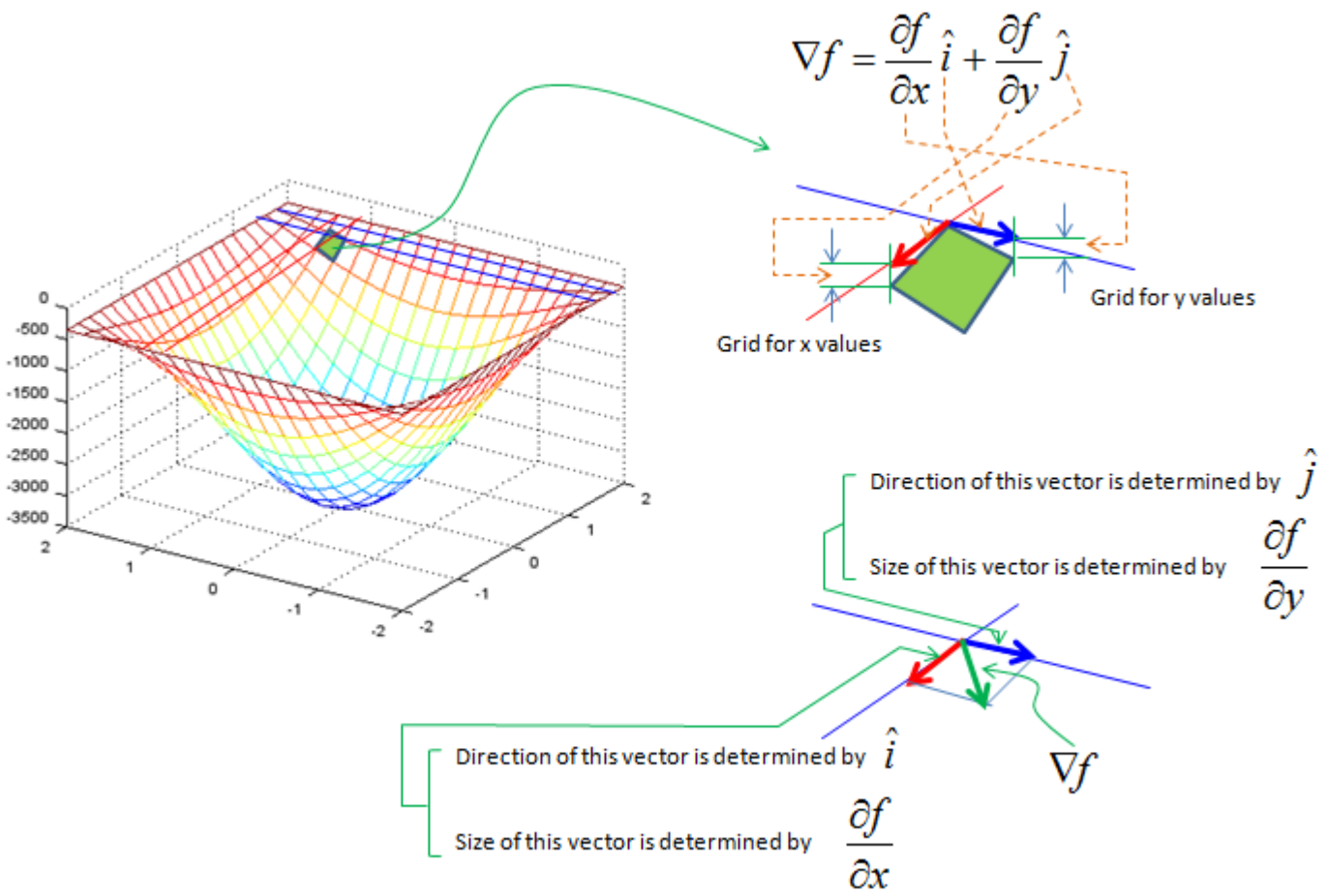


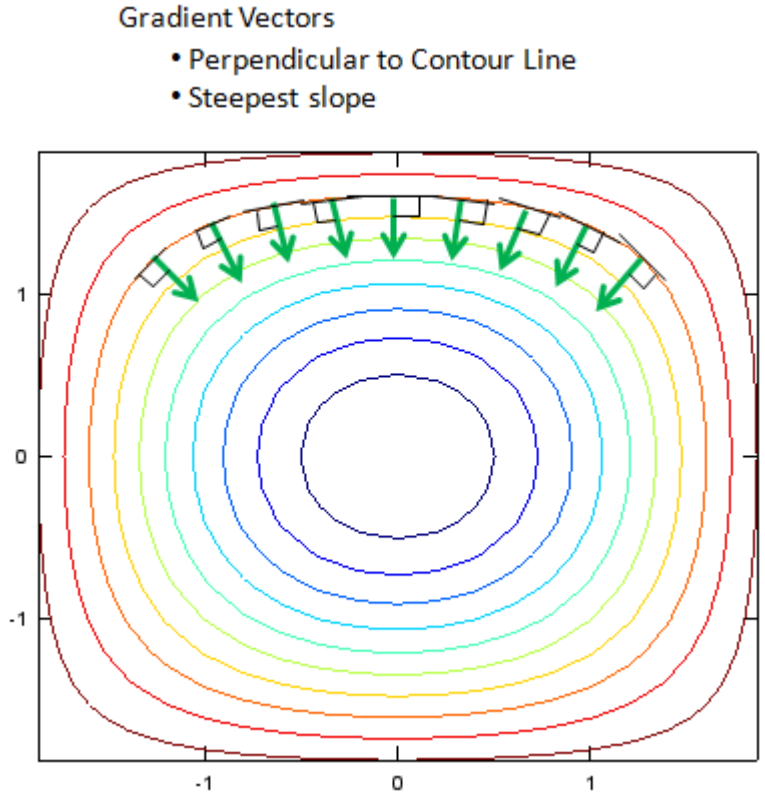
https://en.wikipedia.org/wiki/Hyperparameter_optimization

Gradient descent

★Gradient : 해당 점에서 함숫값이 가장 크게 증가하는 방향

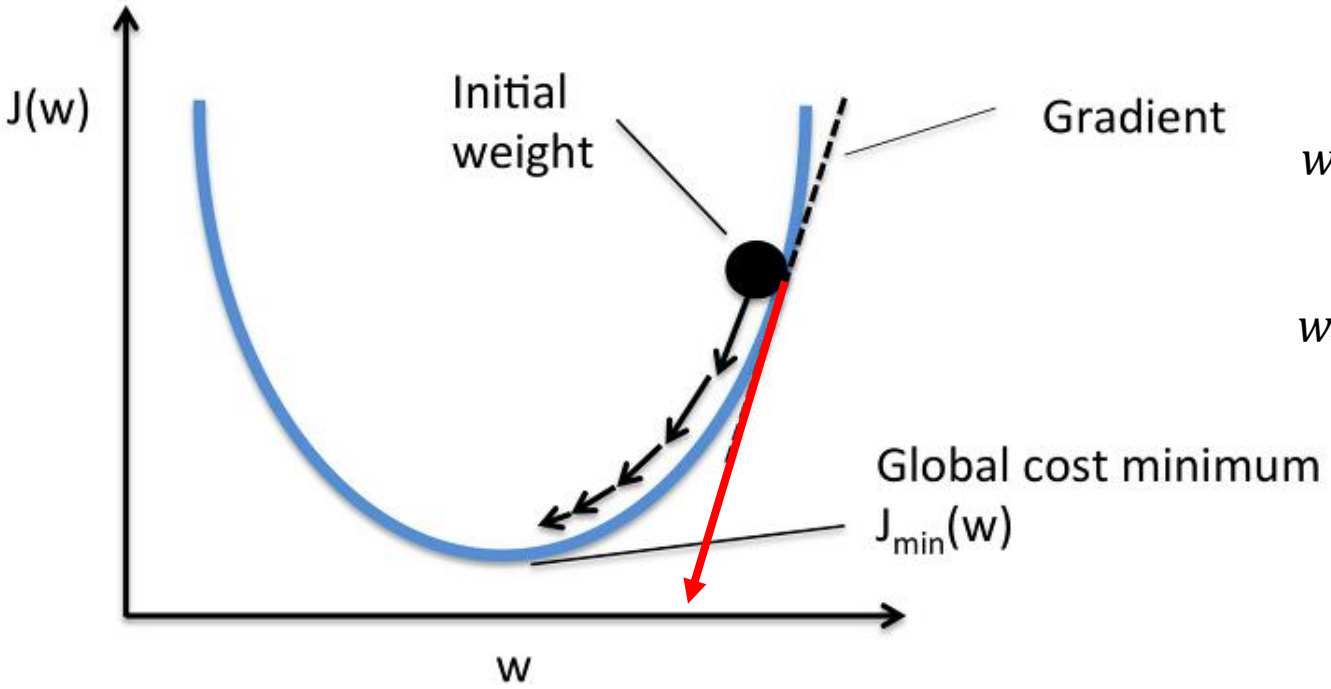
$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)$$





http://www.sharetechnote.com/html/Calculus_Gradient.html

Gradient descent



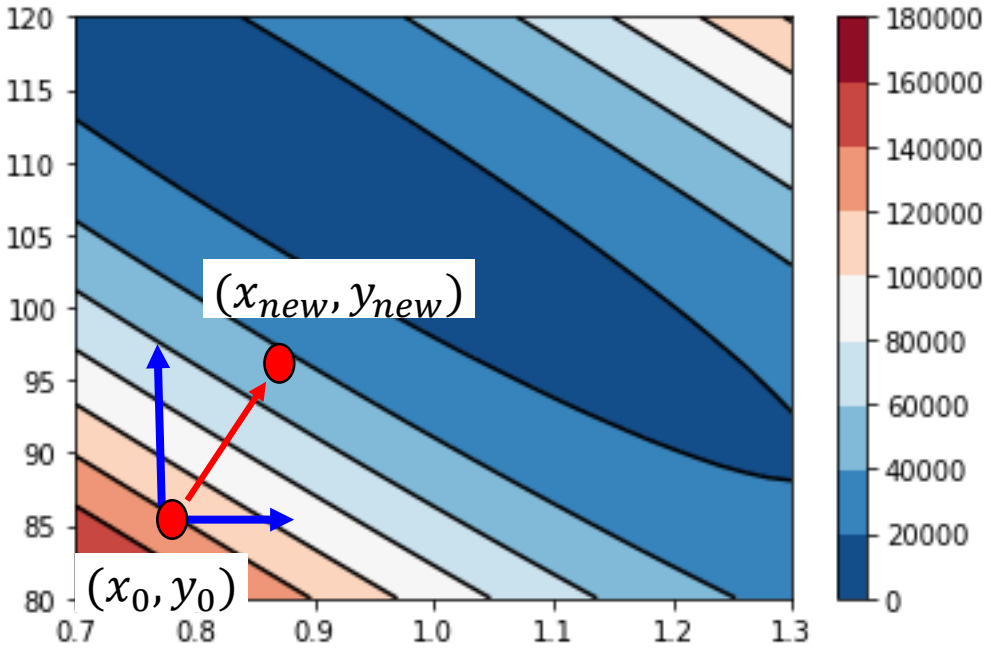
https://en.wikipedia.org/wiki/Gradient_descent

$$w_{new} = w_0 - J'(w_0)$$

$$w_{new} = w_0 - \alpha J'(w_0)$$ 이동하는 정도를 조절해준다!

이 α를 "learning late"라 한다.

Gradient descent



$$x_{new} = x_0 - \alpha \frac{\partial}{\partial x} J(x_0, y_0)$$

$$y_{new} = y_0 - \alpha \frac{\partial}{\partial y} J(x_0, y_0)$$

Gradient descent

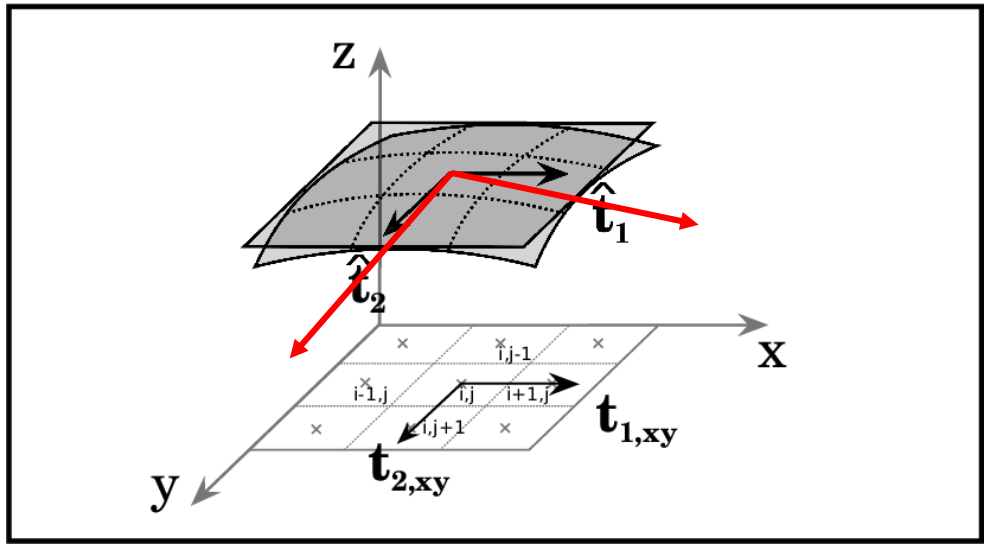
For n-th parameter $(\theta_1, \theta_2, \dots \theta_n)$, update by

$$\theta_i = \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta_1, \theta_2, \dots \theta_n)$$

Where α is "learning late",

$J(\theta_1, \theta_2, \dots \theta_n)$ is "loss function"

(or "object function")



https://www.researchgate.net/figure/A-stencil-used-for-computing-surface-gradient-in-the-column-containing-the-cell-C-i-j_fig1_314152762

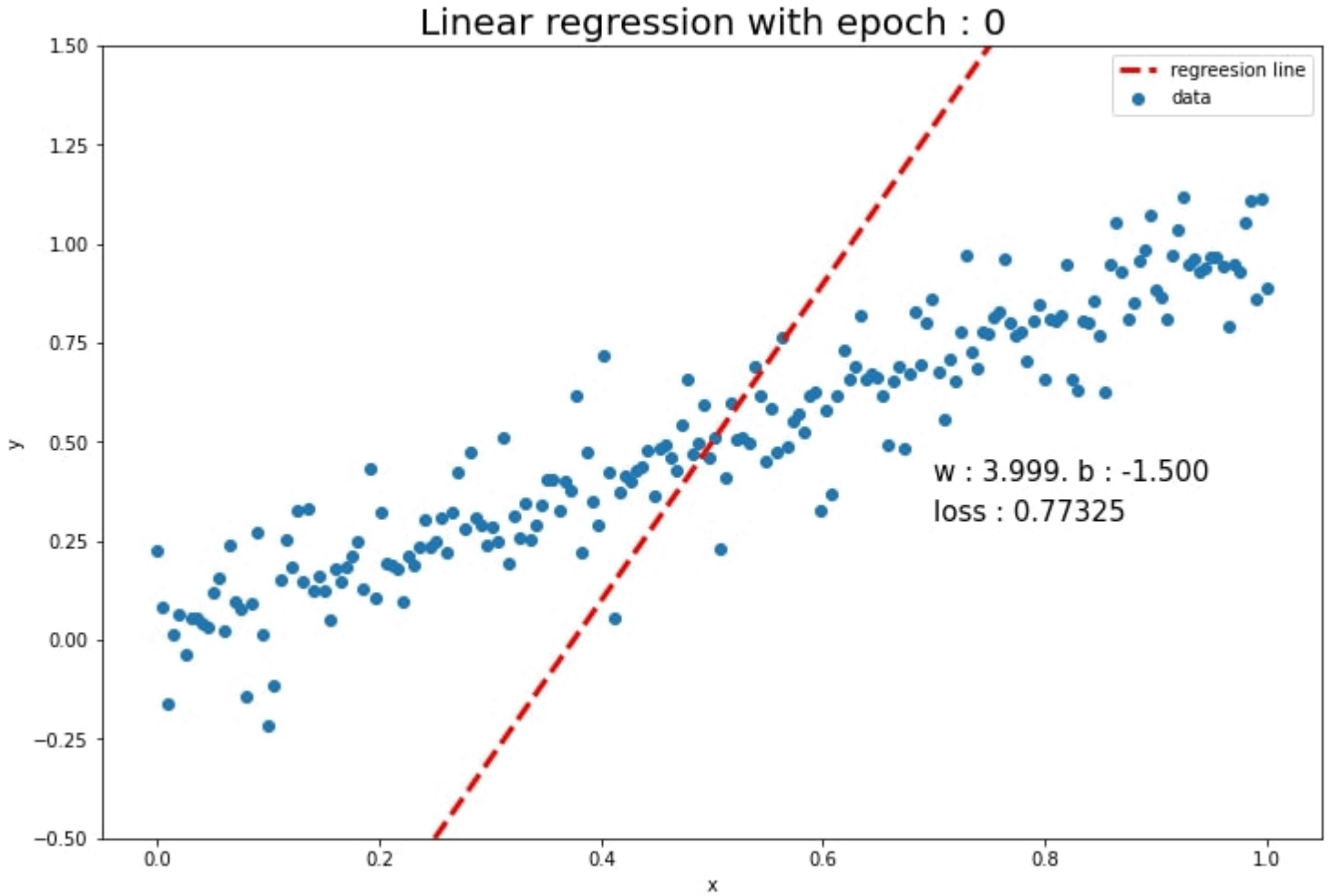
Gradient descent

Our loss function is $J(w,b) = \sum (wx + b - y)^2$

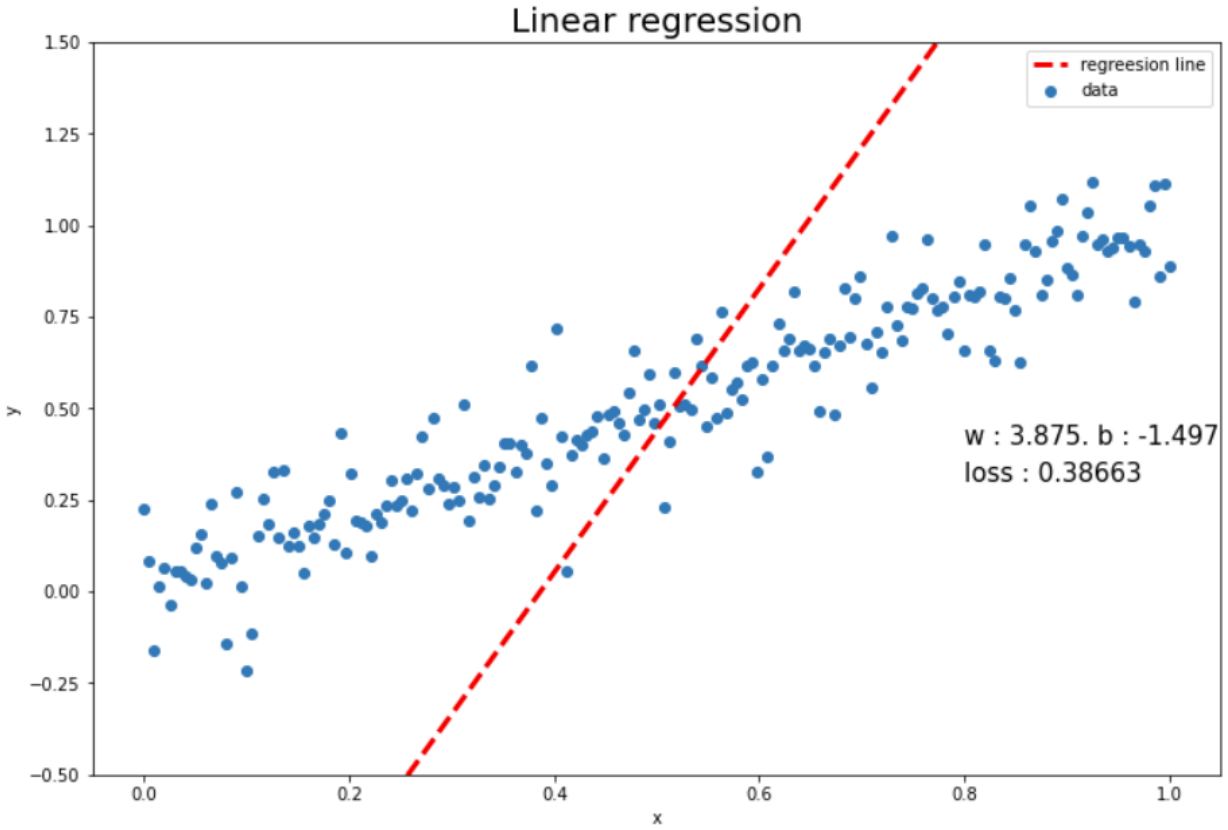
For gradient descent, $w = w - \alpha \frac{\partial}{\partial w} J(w,b) = w - \alpha \sum 2(wx + b - y)x$

$b = b - \alpha \frac{\partial}{\partial b} J(w,b) = b - \alpha \sum 2(wx + b - y)$

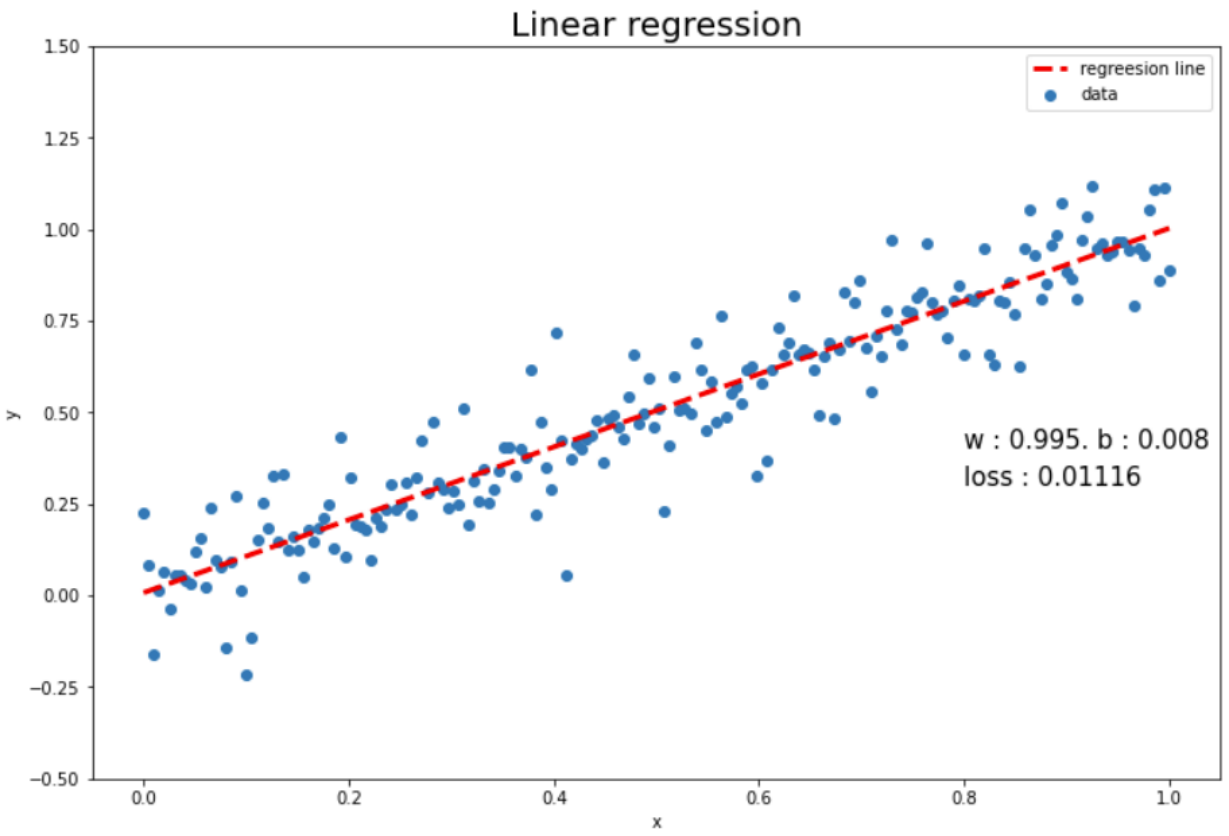
Gradient descent



Gradient descent

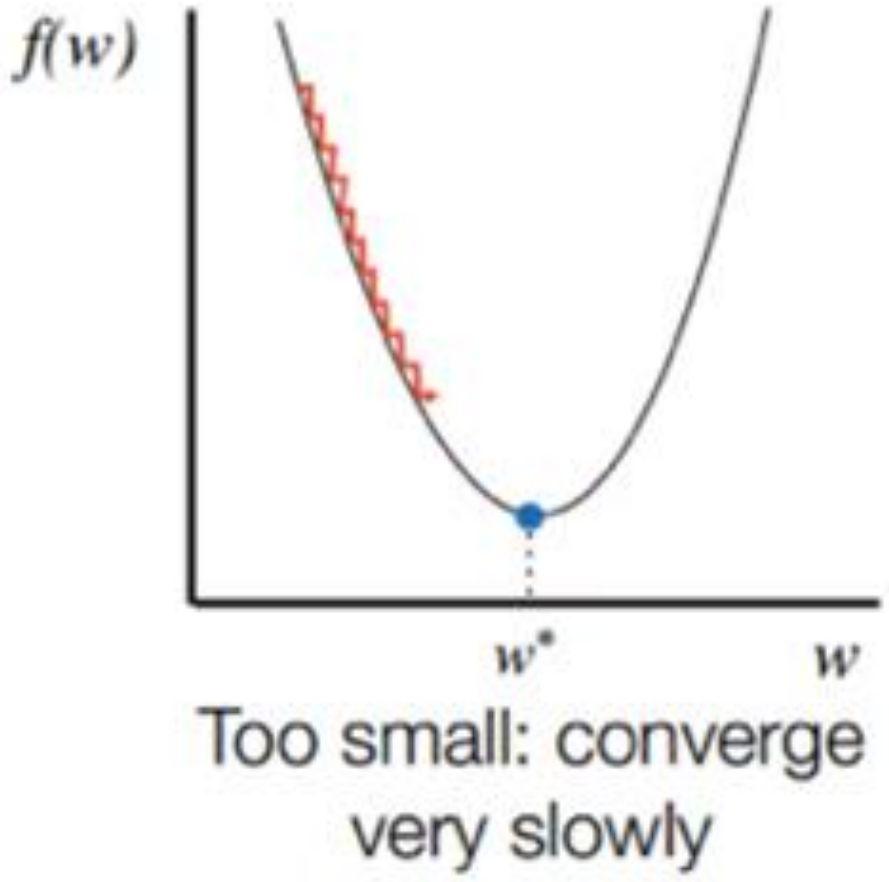


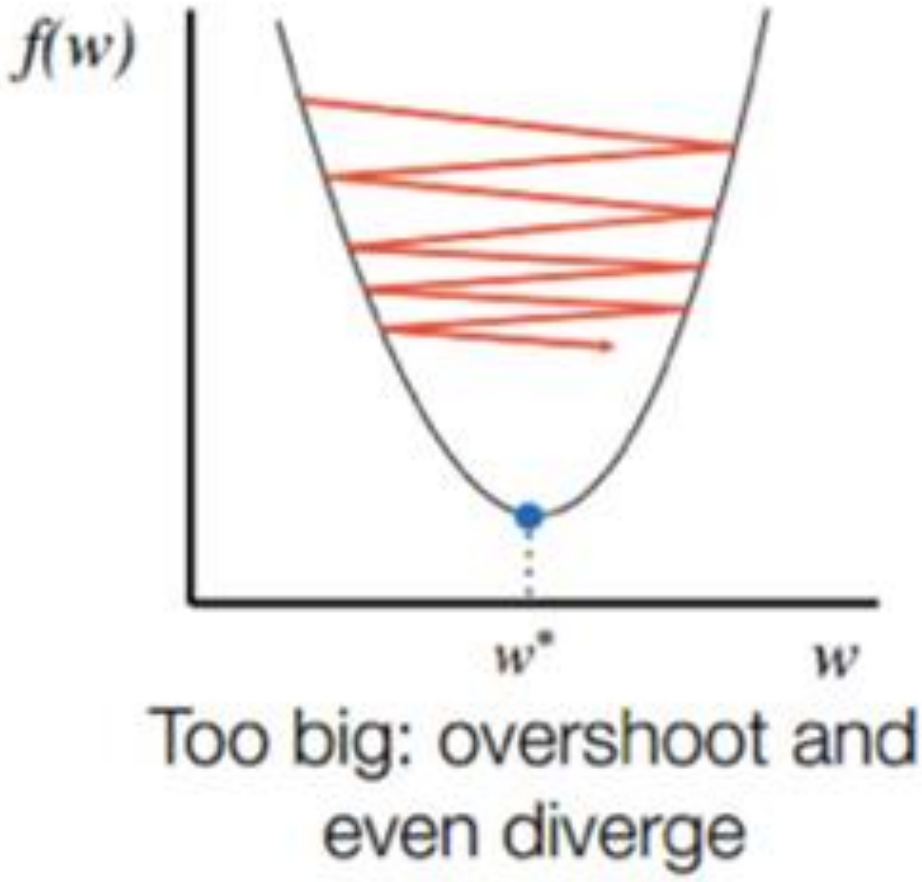
Epoch = 20000 이후



Gradient descent 단점

Learning late를 잘 조절해야 한다.

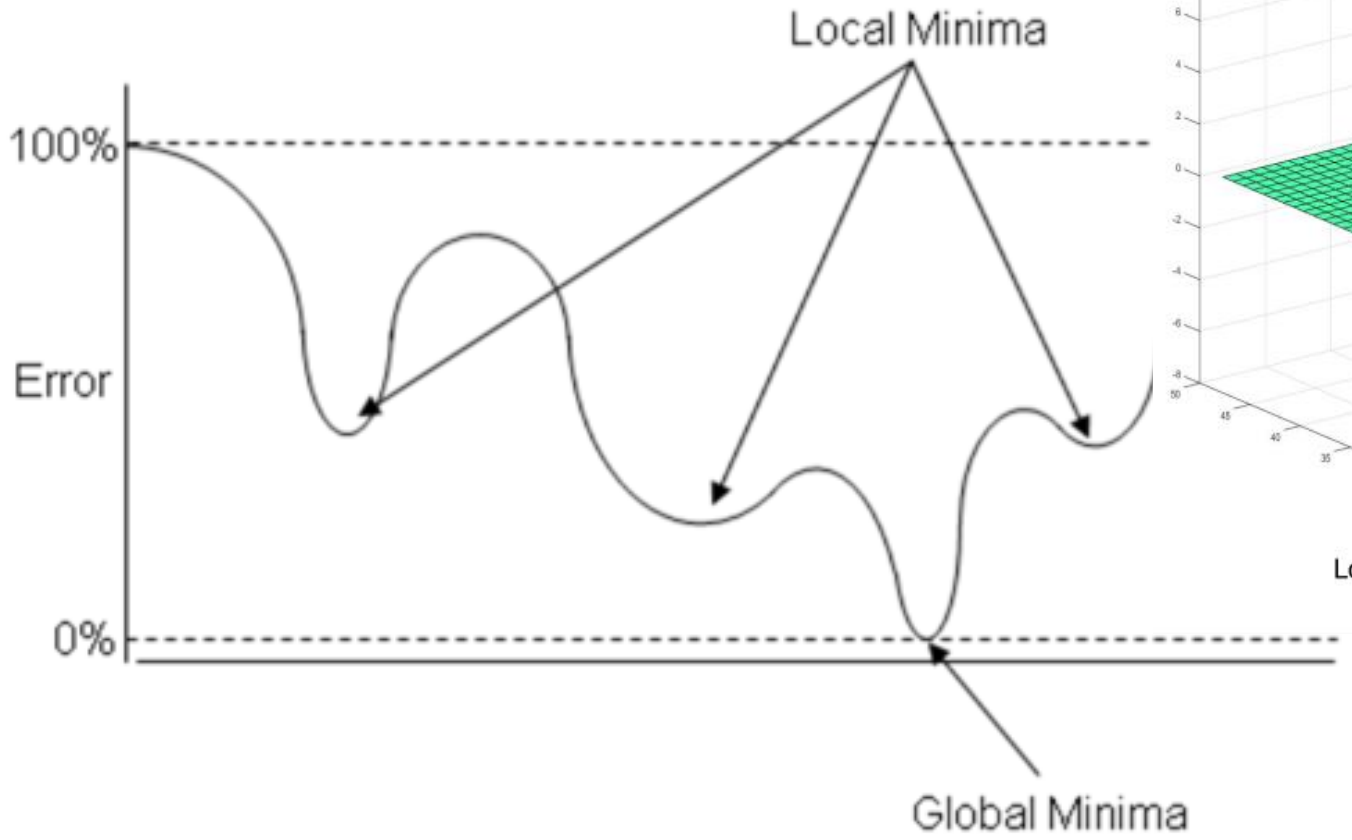


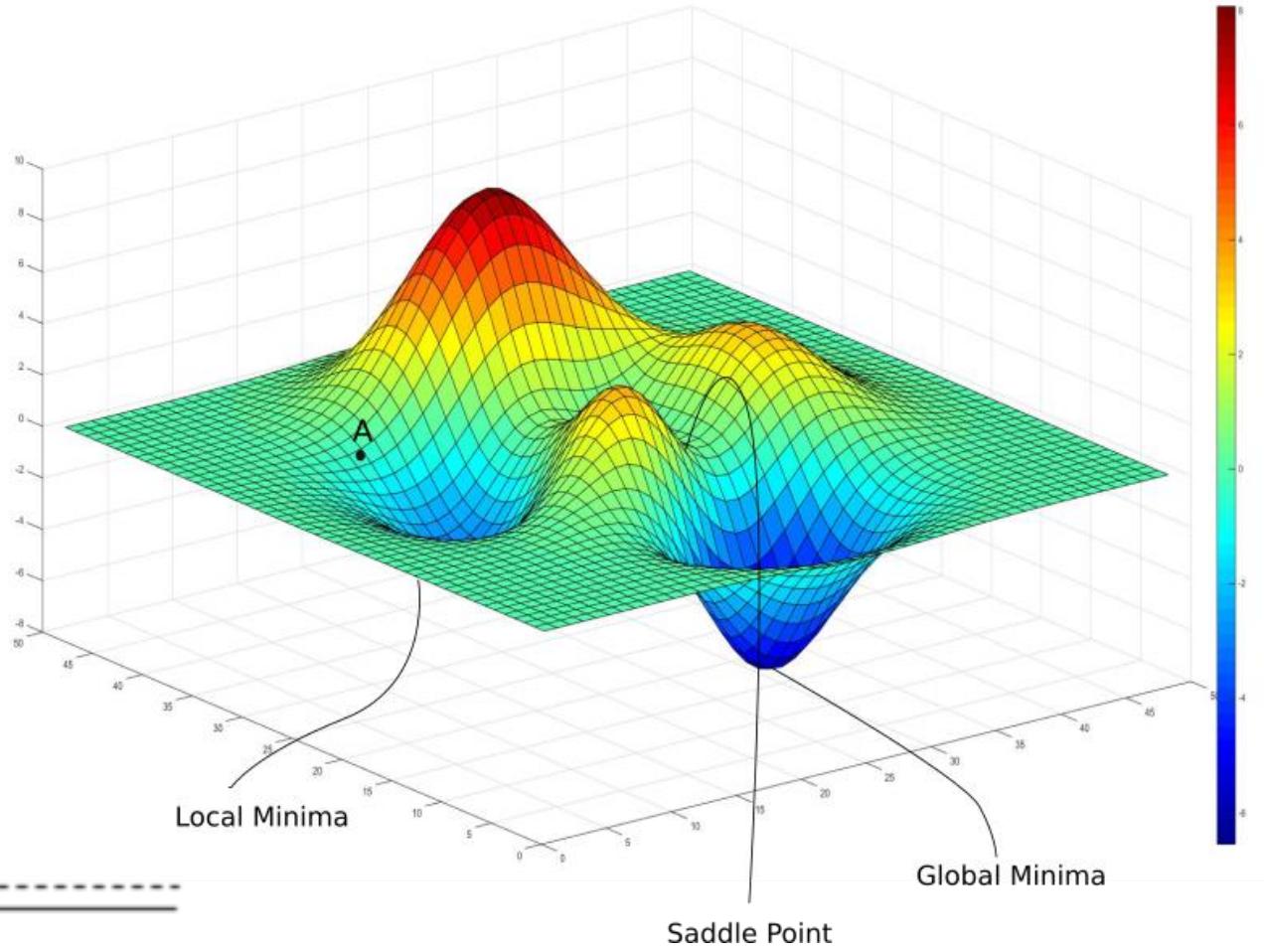


https://m.blog.naver.com/jevida/221855713144

Gradient descent 단점

지역최솟값(Local Minima)에 빠지면 억까당함

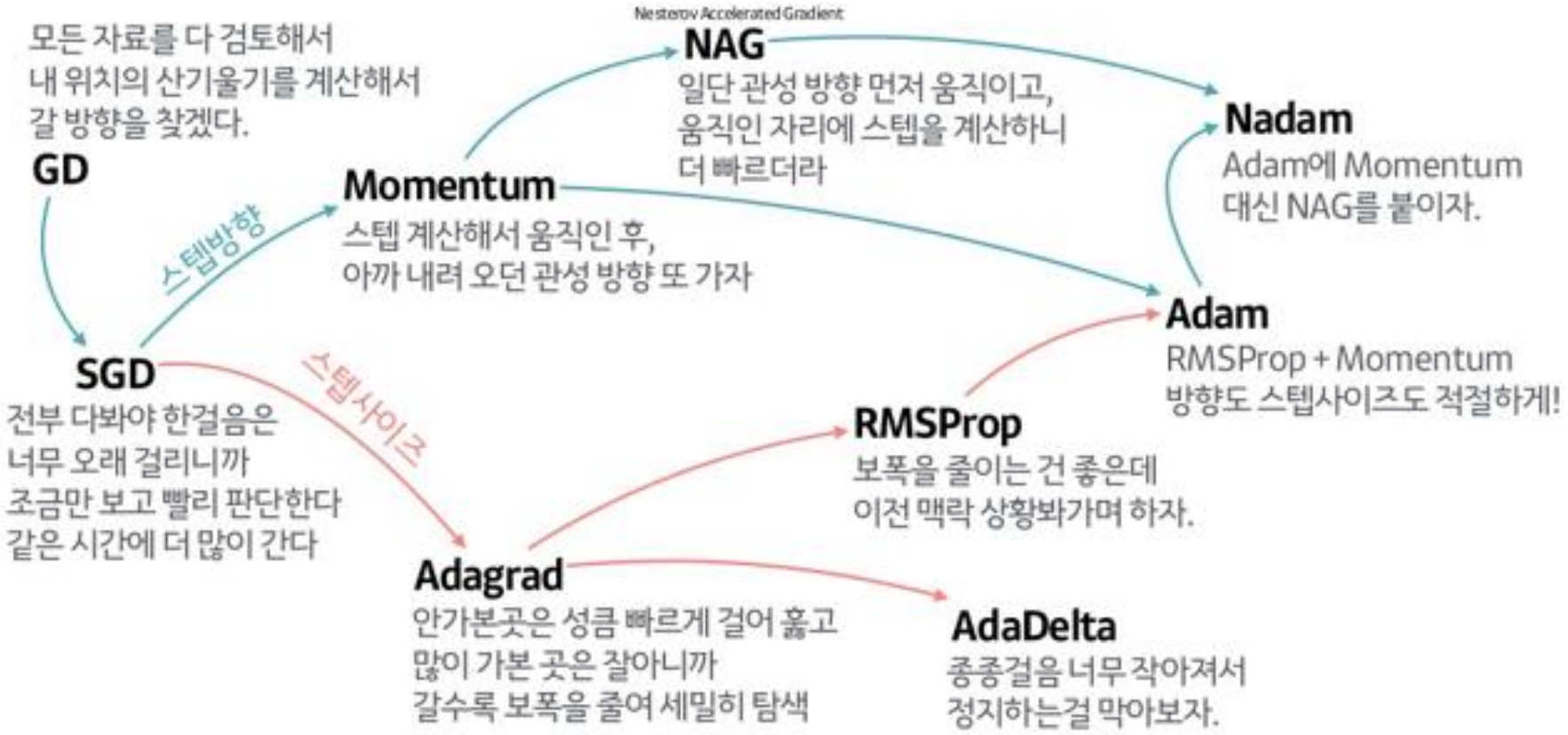




https://m.blog.naver.com/jevida/221855713144

Gradient descent 단점

지역최솟값(Local Minima)를 벗어나게 하는 다양한 알고리즘이 존재.



https://truman.tistory.com/164

Linear Regression coefficient

수학적인 관점에서 $\text{loss} = \left(\text{y}_{\text{pred}} - \text{y}\right)^2 = (wx + b\ - y)^2$ 인데,
각각 $w.b$ 에 대한 편미분이 0인점을 계산해보면 되지 않을까?

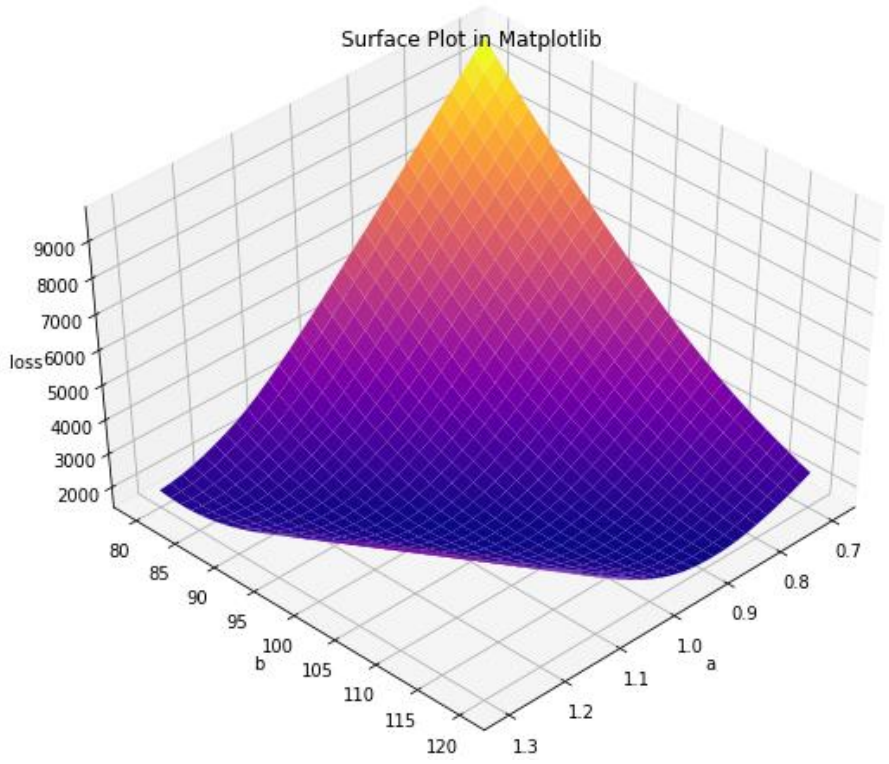
1학년 미분적분학...

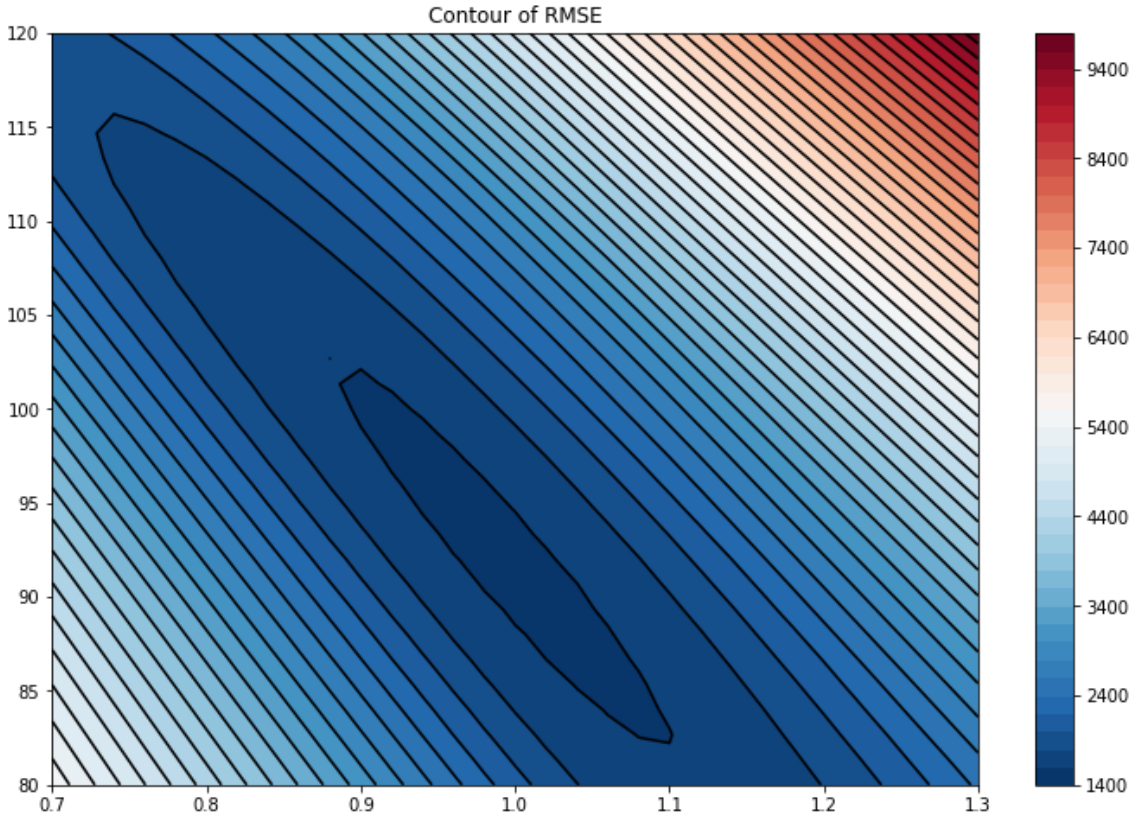
$$\nabla_p f = (y, x) = (0, 0) \qquad H_f = \begin{vmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{vmatrix} = f_{xx}f_{yy} - f_{xy}f_{yx}$$

1) $P\ is\ Local\ Maximum, when\ H_f(P) > 0, f_{xx}(P) > 0$
2) $P\ is\ Local\ Minimum, when\ H_f(P) > 0, f_{xx}(P) < 0$
3) $P\ is\ Saddle\ Point, when\ H_f(P) < 0$
4) $Impossible, when\ H_f(P) = 0$

Linear Regression coefficient

$z = (x - y)^2$ 꼴의 함수는 볼록(convex) 함수이고, 연속이기에 전역최솟값(Global minimizer)가 존재한다.





Linear Regression coefficient

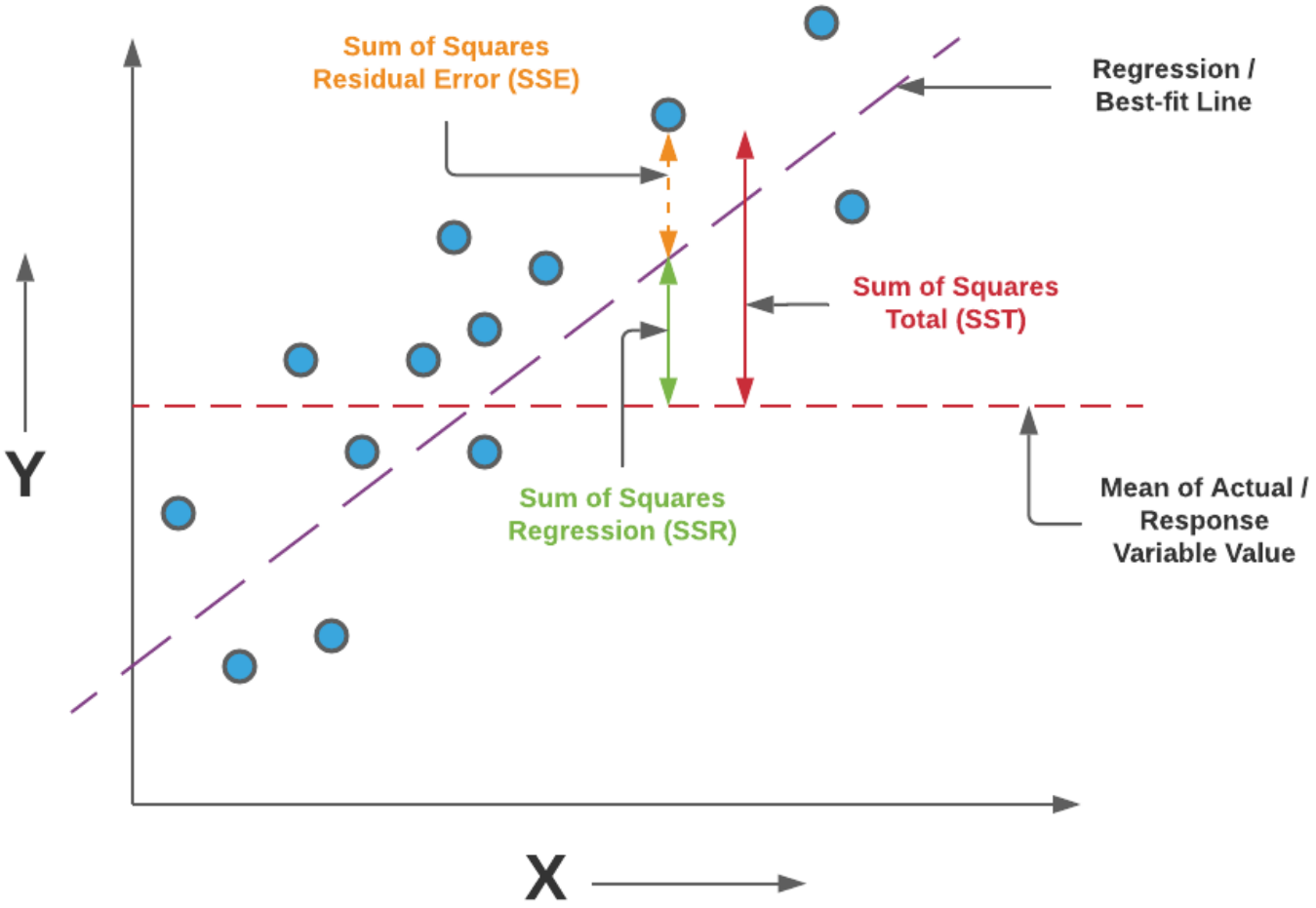
$$y = wx + b$$

각각 편미분해서 0이 되는 점을 계산한 것!

$$w = \frac{Cov(X, Y)}{Var(X)}$$

$$b = E(Y) - wE(X)$$

R2 Score



$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

https://vitalflux.com/mean-square-error-r-squared-which-one-to-use/

R2 Score

Property 1.

$$SST = SSE + SSR$$

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$(a-b)^2 = (a-c)^2 + (c-b)^2$??????? 이게 된다고?

$$1 = \frac{SSR + SSE}{SST} \quad \Rightarrow \quad \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

R2 Score

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

Linear Regression을 평가할 수 있는 척도!

너의 의미는??

전체중에 설명가능한 아이가 얼마나 있니???

$$0 \leq R^2 \leq 1$$

즉, 데이터의 개수와 상관없이 Linear Regression 모델의 성능 평가 가능!

문제 1

Loss function이 RMSE이고, optimizer가 Gradient descent method인 Linear Regression 함수를 아래 조건에 따라 만들어라.

1. 함수이름 : LR_GD(X, Y, w, b, num_epoch, learning_rate)
2. Output : w, b, loss, loss_list
3. Loss가 0.00005 보다 작으면 정지 후 몇 번째 epoch에서 정지했는지 출력.
4. Epoch가 500의 배수일때마다 epoch, w, b, loss를 print 하게 설정.

문제 1

정답

```
def LR_GD(X, Y, w, b, num_epoch, learning_rate):
  # w_list = []
  # b_list = []
  loss_list = []

  for epoch in range(num_epoch):
    Y_hat = w*X + b

    loss = ((Y_hat - X) ** 2).mean()

    if loss < 0.0005:
      break

    w = w - learning_rate * ((Y_hat - Y)*X).mean()
    b = b - learning_rate * (Y_hat - Y).mean()

    loss_list.append(loss)

    if epoch % 500 == 0:
      print("{} epoch  w = {:1.3f}, b = {:1.3f}, loss = {:1.5f}".format(epoch, w, b, loss))

  print("stop at :{}".format(epoch))

  return w, b, loss, loss_list
```

정리 요약

1. Gradient descent method
   - learning rate를 잘 정하는 것이 중요하다.
   - local minima에 빠지면 노답이다.

2. RMSE를 이용한 loss function은 Global minimizer가 존재한다.
   - 따라서 수학적 계산을 통해 가장 좋은 parameter를 해석적으로 찾을 수 있다.

3. 그런데 대부분의 모델들의 Global minimizer를 해석적으로 찾을 수 없으니 Gradient descent method 같은 수치해석방법을 이용하여 접근한다.

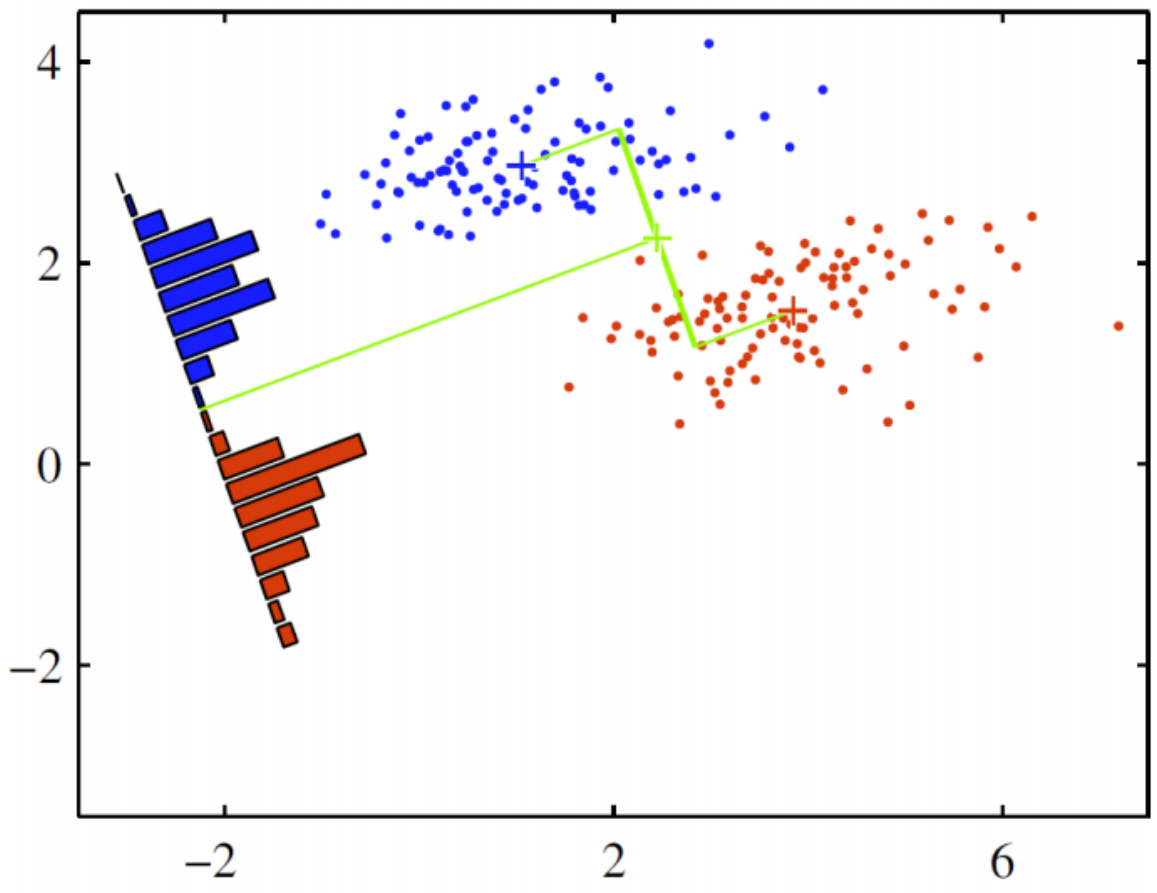
정리 요약

4. R2 Score는 RMSE를 이용한 Linear regression 모델에서만 설명 가능.
   (아무 regression에서 막 사용하면 안됨!)

   (근데 현업에서는 그냥 무시하고 쓰는 경우도 있더라…)

공지

1. 다음시간에는 선형판별분석(Linear Discriminant Analysis)에 대해 배울 것



끝

담에봅시당