Gradient Descent (GD) Algorithm

Zengchang Qin (Ph.D.)

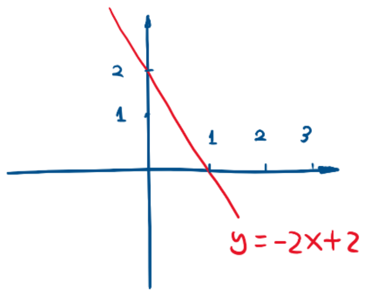
March 26, 2018

Given a database $D = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots (x^{(N)}, y^{(N)})\}$
where $x^{(i)}$ is a vector in n-dimensional space and $y^{(i)}$ is a scalar
i.e.: $x_i \in R^n$, $y_i \in R$

We hope to learn $f(x^{(i)}) \to y^{(i)}$.

This is a typical machine learning problem

1) If our hypothesis of relation function is a linear model.



$y = ax + b$

$y = \theta_1 x + \theta_0$

if we set $x_0 = 1$

$y = \theta_1 x_1 + \theta_0 x_0$

$= \theta^T x$

$\theta^T = (\theta_0, \theta_1)$

$x = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}$

In general form

$y = \theta^T x$ $\quad \begin{cases} \theta \in R^{n+1} \\ x \in R^{n+1} \end{cases}$

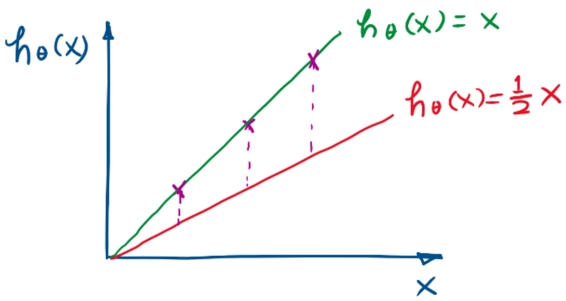
$$y^{(i)} = h_\theta(x^{(i)}) = \theta^T x^{(i)} = \sum_{j=1}^{n} \theta_j x_j^{(i)}$$

Let us consider the simplest case that

$$h_\theta(x) = \theta_1 x_1 + \theta_0, \text{ where } \theta_0 = 0$$

If given the training data

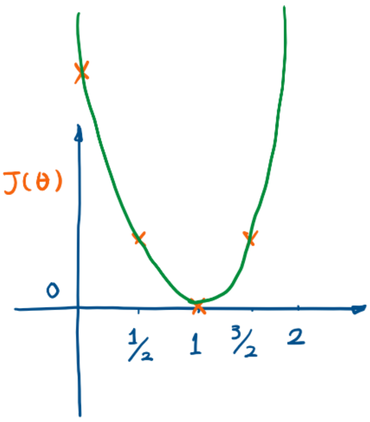
$$\boxed{\begin{matrix} x^{(1)} = 1, & y^{(1)} = 1 \\ x^{(2)} = 2, & y^{(2)} = 2 \\ x^{(3)} = 3, & y^{(3)} = 3 \end{matrix}}$$



$$J(\theta) = \frac{1}{2N} \sum_{i=1}^{N} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$J\left(\tfrac{1}{2}\right) = \frac{1}{2\times 3}\left(\left(-\tfrac{1}{2}\right)^2 + (-1)^2 + \left(-\tfrac{3}{2}\right)^2\right)$$
$$= \frac{1}{6}\left(\frac{1}{4} + \frac{4}{4} + \frac{9}{4}\right) = \frac{14}{24} = \frac{7}{12}$$

$$J(0) = \frac{1}{2\times 3}\left((-1)^2 + (-2)^2 + (-3)^2\right)$$
$$= \frac{1}{6} \times 14 = \frac{7}{3}$$

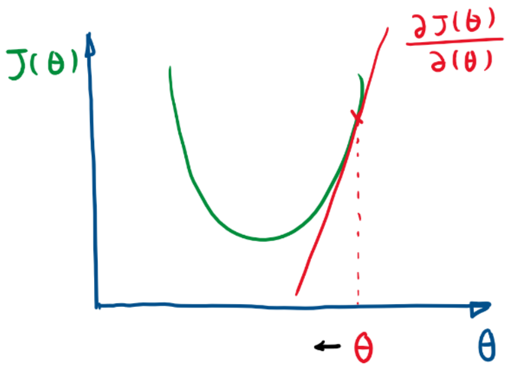


2) How to minimize Cost function $J(\theta)$?

$$J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

If $h_\theta(x) = \theta x$

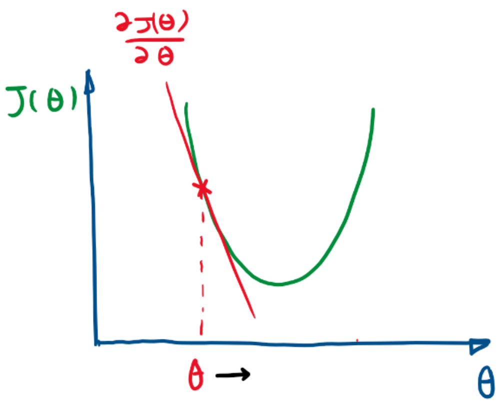
$$\frac{\partial J(\theta)}{\partial \theta} = \frac{\partial}{\partial \theta}\left(\frac{1}{2N}\sum_{i=1}^{N}\left(\theta x^{(i)} - y^{(i)}\right)^2\right)$$
$$= \frac{1}{N}\sum_{i=1}^{N}\left(\theta x^{(i)} - y^{(i)}\right)x^{(i)}$$



$$\theta \leftarrow \theta - \alpha\frac{\partial J}{\partial(\theta)}$$

if $\frac{\partial J}{\partial \theta} > 0$ (positive)

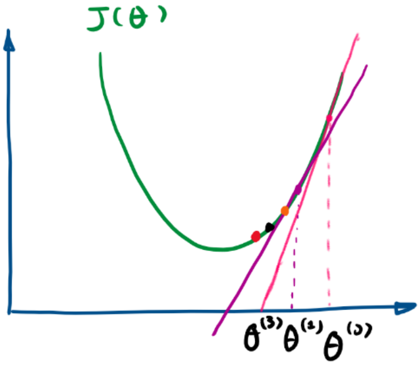
$\theta \leftarrow \theta - \alpha$ (positive number)



if $\frac{\partial J(\theta)}{\partial \theta} < 0$ (negative)

$\theta \leftarrow \theta - \alpha$ (negative number)

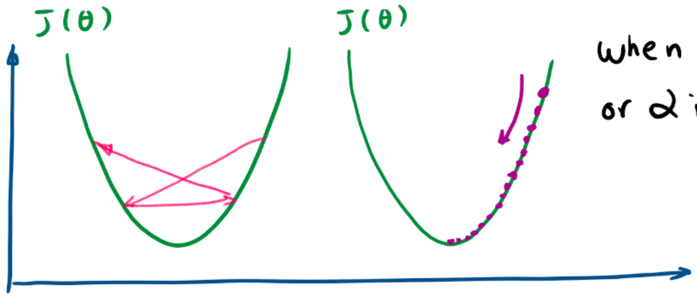
How Gradient Change ?



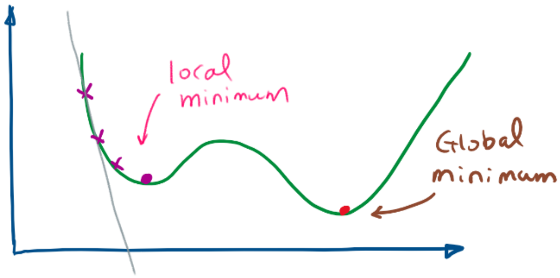
$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \alpha \frac{dJ(\theta)}{d\theta}$$

Gradient is diminishing in the process of iterative updating.

α can be fixed !



when α is too big.
or α is too small.



Gradient descent does not have the graunteе to converge to the global minimum. Be aware of the local minimum.

3) Gadient descent :

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

E.g. for the above case of $h_\theta(x) = \theta X$

$$\theta \leftarrow \theta - \alpha \frac{dJ(\theta)}{d\theta}$$

If $h_\theta(x) = \theta_1 X + \theta_0$, GD is updated Simultaneously by :

$$\boxed{\begin{aligned} \theta_0 &\leftarrow \theta_0 - \alpha \frac{J(\theta_0, \theta_1)}{\partial \theta_0} \\ \theta_1 &\leftarrow \theta_1 - \alpha \frac{J(\theta_0, \theta_1)}{\partial \theta_1} \end{aligned}}$$

$(\alpha > 0)$ is the learning rate.

$$\text{where } \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0} = \frac{\partial}{\partial \theta_0}\left(\frac{1}{2N} \Sigma_{i=1}^{N} (h_\theta(x^{(i)} - y^{(i)})^2 \right)$$

$$= \frac{1}{N} \Sigma_{i=1}^{N} h_\theta(x^{(i)} - y^{(i)})$$

$$\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1} = \frac{1}{N} \Sigma_{i=1}^{N} h_\theta(x^{(i)} - y^{(i)}) x^{(i)}$$

4) For a more general case of multivariate inear regression where $h_\theta(x) = \theta^T x$

The gradient descent algorithm becomes :

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

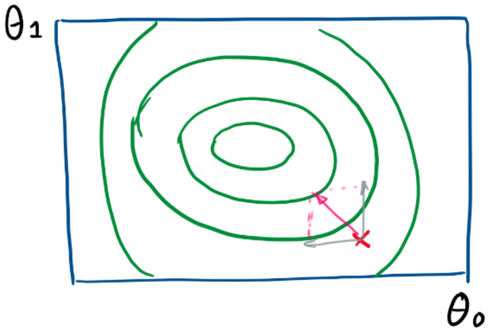
$$= \theta_j - \alpha \frac{1}{N} \sum_{i=1}^{N} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

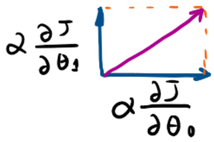
$$\theta_0 \leftarrow \theta_0 - \alpha \frac{1}{N} \sum_{i=1}^{N} (h_\theta(x^{(i)} - y^{(i)}) x_0^{(i)} \qquad x_0 = 1$$

$$= \theta_0 - \alpha \frac{1}{N} \sum_{i=1}^{N} (h_\theta(x^{(i)} - y^{(i)})$$

A batch of data $(x_i : i = 1 \cdots N)$ is used for updating θ_j $(j = 1 \cdots n)$

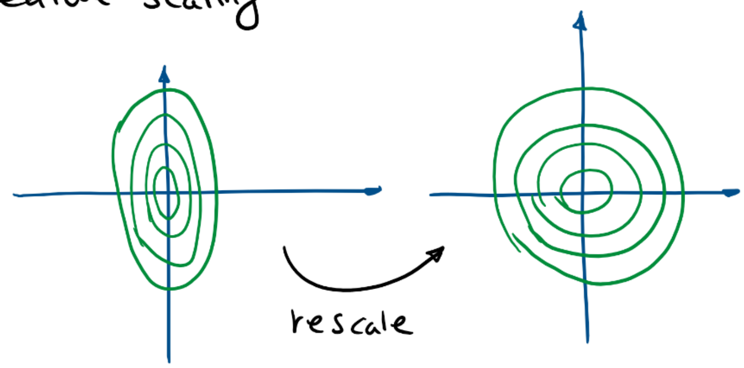
5) Contour with two variables (gradient is the direction, α is actually the step forward)





It is possible to be trapped into a local optima.
We can try different initialization

6) Feature Scaling



① $x_i \leftarrow \dfrac{x_i}{\max(x_i) - \min(x_i)}$

② $x_i \leftarrow \dfrac{x_i - \bar{x}}{\max(x_i) - \min(x_i)}$, where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$

7) A linear Combination of features tells the importance of each feature towards the target.

$$y = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 \cdots \theta_n x_n$$

Weight

$\theta_j > 0$, positive influence

$\theta_j < 0$. negative influence .