Simple Linear Regression

Input: $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$

Output: $H(x) = \theta_0 + \theta_1 x$, where $H(x)$ is our hypothesis function.

Cost:

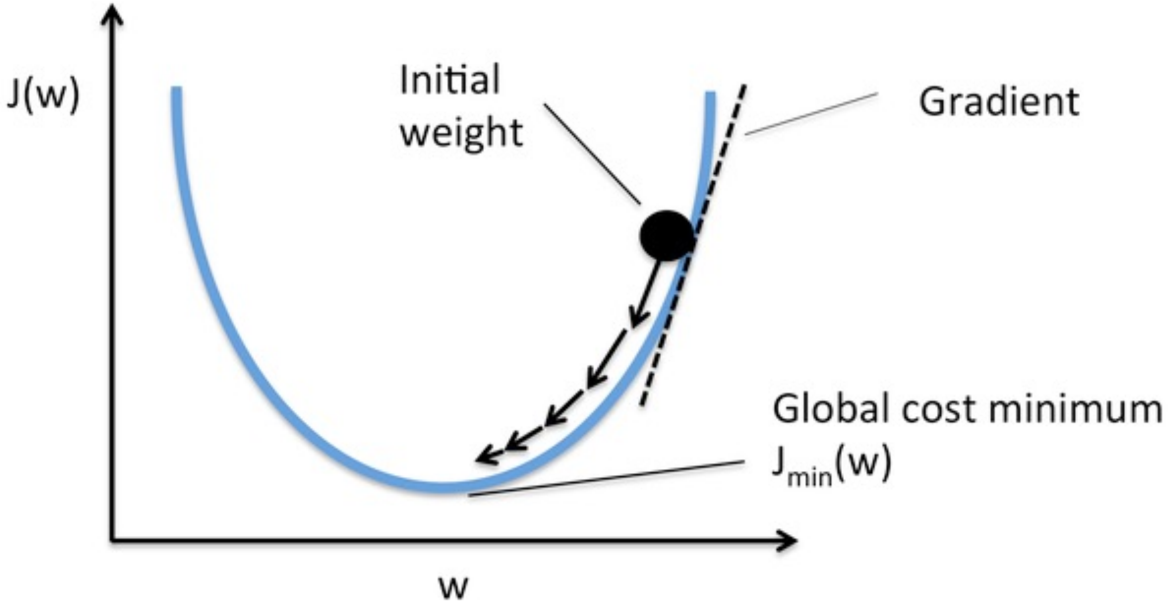
$$MSE = \frac{1}{2} \cdot \sum_{i=1}^{n} \left(y_i - H(x_i)\right)^2$$

To find the best line, we need to minimize our cost function during the learning.

Gradient Descent Method

Gradient Descent is one of the optimizers which helps in calculating the new weights.

In the image below, the curve is our cost function curve, and our aim is the minimize the error such that J_{min} i.e global minima is achieved.



Gradient Descent Method

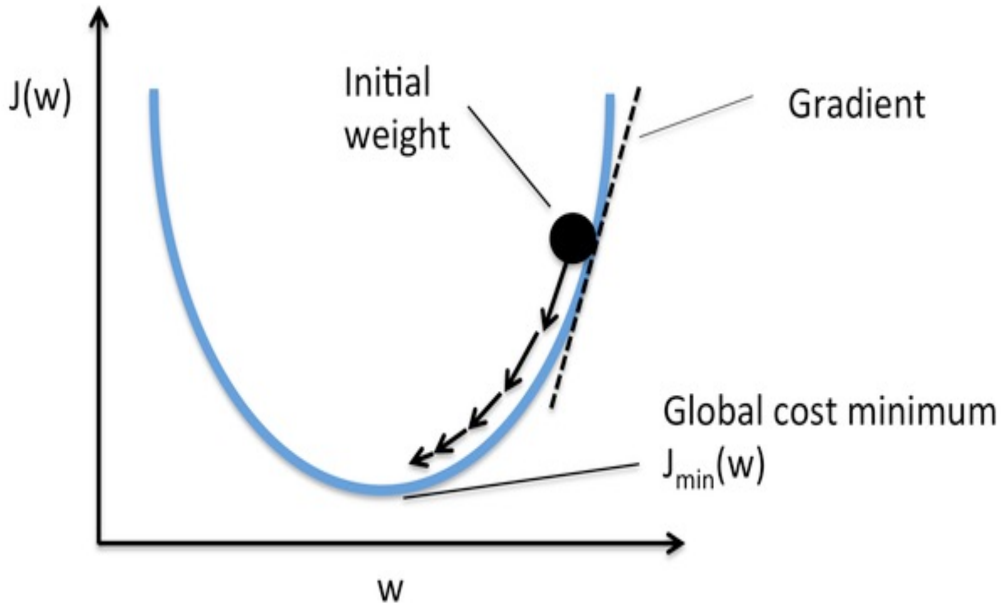
First, **the weights are initialized randomly** i.e random value of the weight,

and intercepts are assigned to the model

while forward propagation

and **the errors are calculated** after

all the computation.



Gradient Descent Method

Then new weights are calculated using the next slide formula,

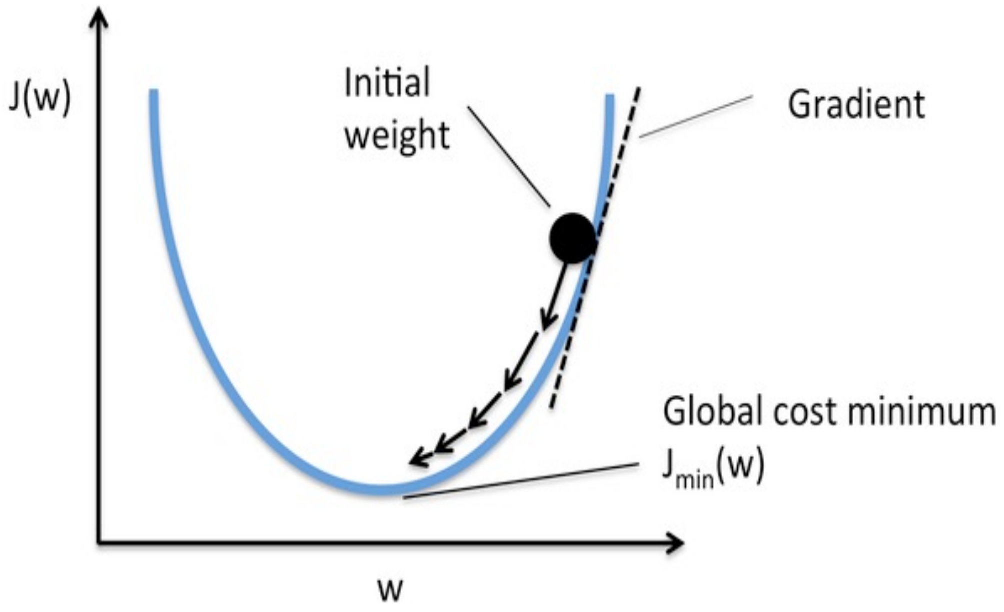
where α **is the learning rate** which is the

parameter also known as step size

to control the speed or steps of

the backpropagation.

It gives additional control



on how fast we want to move on the curve to reach global minima.

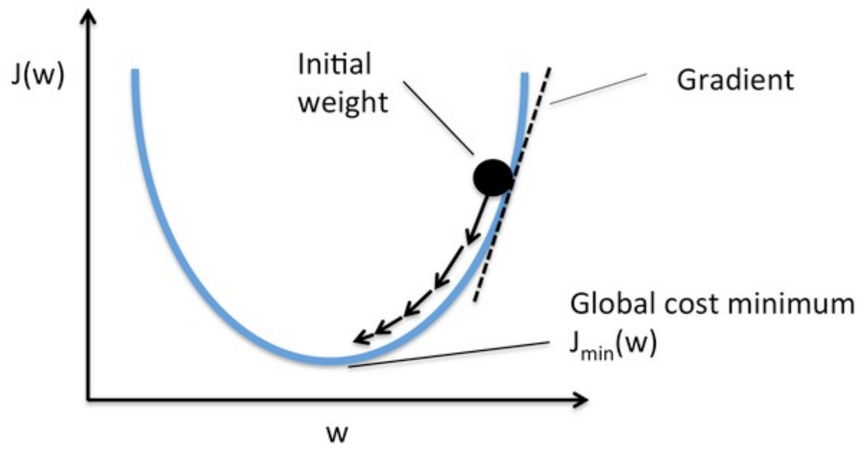
Gradient Descent Method

W_x^* - the new weight

W_x - the old weight

α - learning rate

Each step, calculate:



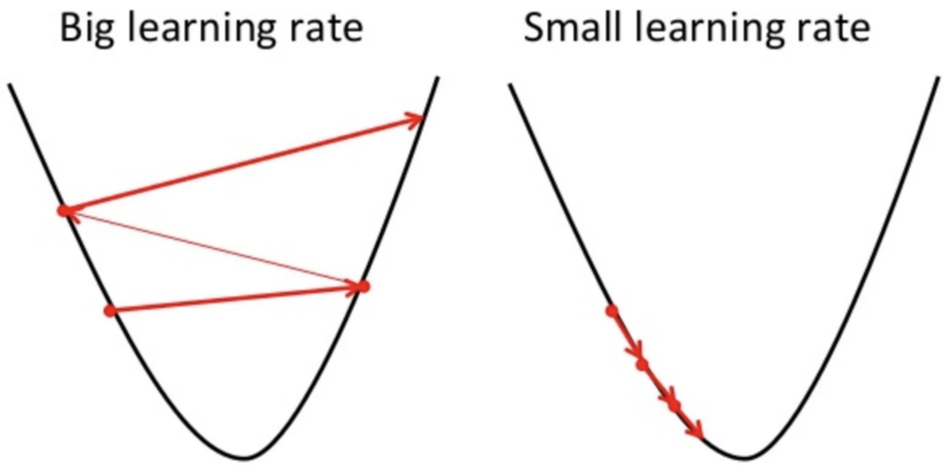
$$W_x^* = W_x - \alpha \cdot \frac{\partial Error}{\partial W_x}$$

This process of calculating the new weights, then errors from the new weights, and then updation of weights **continues till we reach global minima and loss is minimized**.

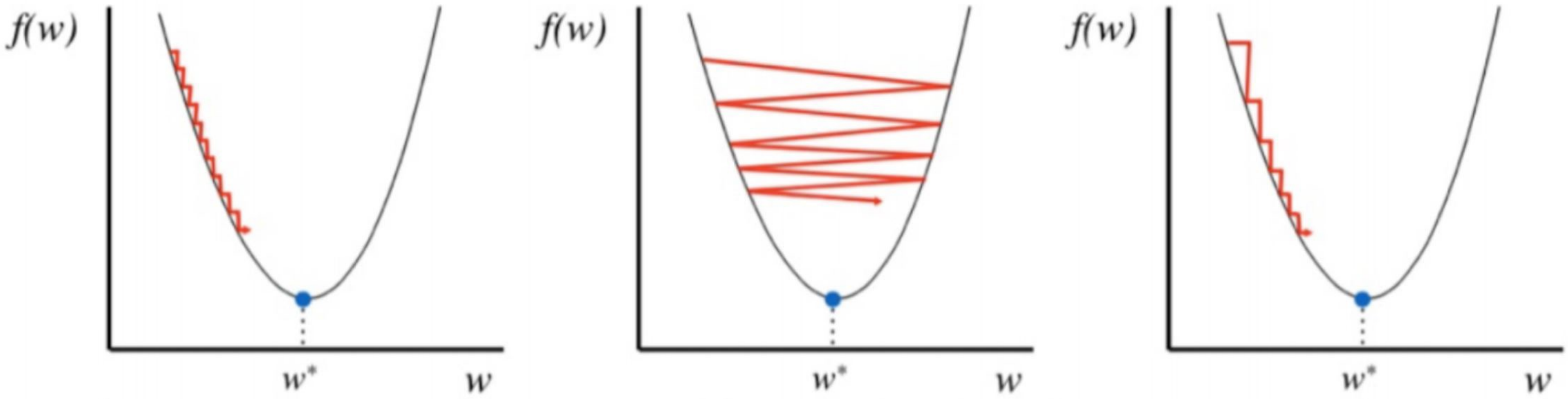
Gradient Descent Method

The learning parameter

- **should not be very small** as it will take time to converge as well as
- **should not be very large** that it doesn't reach global minima at all.



Gradient Descent Method



Too small: converge very slowly

Too big: overshoot and even diverge

Reduce size over time

Simple Linear Regression - Algorithm

1 - Randomly initializing θ_1, θ_0 for the hypothesis function

2 - Compute $MSE = \frac{1}{2} \cdot \sum_{i=1}^{n}(H(x_i) - y_i)^2$ as C

3 - Find the partial derivatives $\frac{\partial C}{\partial \theta_0}$, $\frac{\partial C}{\partial \theta_1}$

4 - Update parameters based on the derivatives and the learning rate:

$$\theta_0 = \theta_0 - \alpha \cdot \frac{\partial C}{\partial \theta_0}$$

$$\theta_1 = \theta_1 - \alpha \cdot \frac{\partial C}{\partial \theta_1}$$

5 - Repeat until the error is minimized.

Simple Linear Regression – Algorithm Analysis

What is the derivation of the i^{th} step?

$$H(x) = \theta_0 + \theta_1 x$$

$$Cost = MSE = \frac{1}{2} \cdot \sum_{i=1}^{n} \left(y_i - H(x_i)\right)^2 = \frac{1}{2} \cdot \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i)^2$$

$$\frac{\partial C}{\partial \theta_0} = \frac{1}{2} \cdot 2 \cdot \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i) \cdot (-1)$$

$$\frac{\partial C}{\partial \theta_1} = \frac{1}{2} \cdot 2 \cdot \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i) \cdot (-x_i)$$