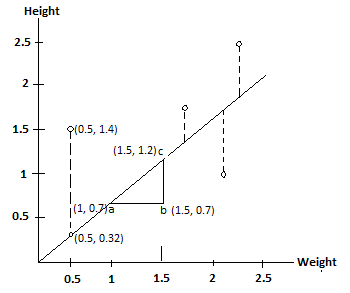
Linear Regression and Gradient Descent

## Linear regression



Equation of a straight line:

🡪 Best fit line

where m=slope and c = y intercept when x = 0

**Slope:** for a unit change on x axis, what is the change on y-axis.

From the figure above, for a unit change from point a to point b (1.5 – 1 = 0.5) the change in height is (1.2 – 0.7 = 0.5).

Therefore, = In this case, it is 1. The steeper the line, the greater the slope and vice-versa.

Cost function (Mean squared error) =

y’ = predicted value for y

n = number of observations

m = slope

Now, let’s start with intercept (c) = 0, and an initial random value of slope (m) = 0.64

For c=0 and m=0.64, we have y = 0.64x

In fig 1, we have one point (0.5, 1.4)

For x = 0.5, predicted value of y would be 0.64 x 0.5 = 0.32

y’ = 0.32

The difference of y – y’ = 1.4 – 0.32 = 1.1

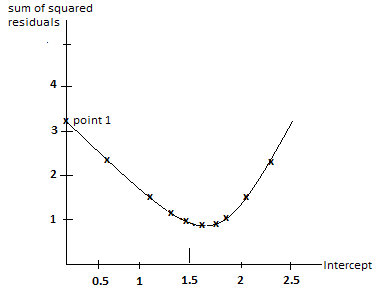
We do that for all the other points. Let’s just move on with the calculations. Our values are 0.4 and 1.3

Squaring the difference helps to offset the values of points where y’ > y (i.e. points that lie below the best fit line). A point below the line and farther, will negate the total sum and give a less value of the sum of squared residuals. Which is not a good value.

So, we have 1.12 + 0.42 + 1.32 = 3.1

## Gradient Descent

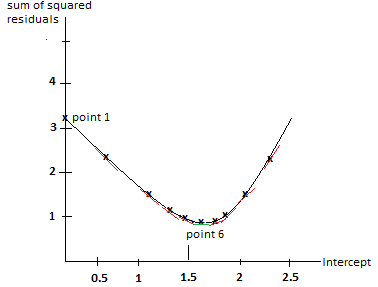
Therefore, for 1 line with respect to all the data points we get 3.1 as the sum of squared residuals. We plot a graph of sum of squared residuals vs intercept.



For, c = 0 we get point 1 on the y-axis. Similarly for different values of intercept we continue to plot the points.

The next intercept value after 0 is determined using step size by the algorithm.

Gradient descent takes long steps for intercept values at first. When the step size starts approaching 0 it takes small values for intercept. Step size starts approaching 0 when slope starts approaching 0. The slope of the line at point 6 is almost 0. This is how a gradient descent is formed.



With three points, the maths did not take long. But with millions of data points, it takes time. So we can use stochastic gradient descent.

## Stochastic Gradient descent

uses a randomly selected subset of the data at every step instead of the full dataset. This reduces the time spent on calculating derivatives of the loss function.