Getting Started with Machine Learning

Raunak J

So far now we've learned :

Making a Simple Linear Regression model.

Computing its loss.

Finding Goodness-of-Fit.

Since we're finding the loss, is it useful ?

Yes, it is if we can reduce it.

We require Loss Optimizers.

Do we have any ?

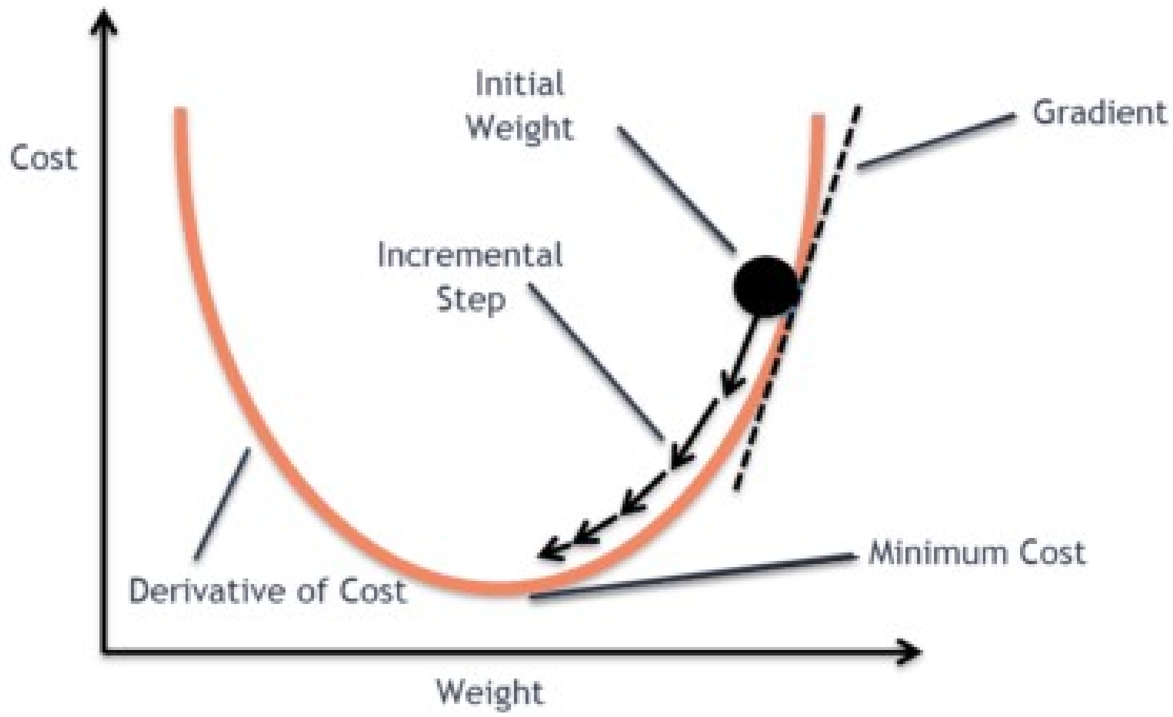
Definitely we do…

In this tutorial we’ll be learning Gradient Descent :

So what is Gradient Descent ?

Gradient descent is a first-order iterative optimization algorithm for finding the local minimum of a function. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient of the function at the current point.

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



Source : https://kraj3.com.np/blog/2019/08/a-mathematical-approach-towards-gradient-descent-algorithm/

So basically the aim will be to converge our loss function.

So as the loss is reduced, the model is able to predict better values.

Which builds model better.

So what is the mathematical expression of Gradient Descent ?

So Gradient Descent for Loss/Cost Function of Linear Regression.

The Loss Function we use for Linear Regression is MSE.

$$Mean\ Squared\ Error\ (MSE) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Gradient Descent procedure :

Take the Partial Derivates of Loss/Cost Function..

Iterate by updating the weights. (Convergence)

Repeat in epochs.

First we'll start by taking derivates of Loss Function (MSE)

So our algorithm on which we'll perform Gradient Descent is Linear Regression.

So we'll be converging β_0 and β_1

So the convergence will be with respect to β_0 and β_1.

$$This\ is\ Mean\ Squared\ Error\ (MSE) = \frac{1}{N}\sum(y_i - \widehat{y_i})^2$$

$$But\ we\ can\ also\ express\ it\ as \frac{1}{N}\sum(y_i - \{\beta_0 + (\beta_1 * x_i)\})^2$$

$$Since\ \widehat{y_i} = \beta_0 + (\beta_1 * x_i)$$

$$\frac{\partial}{\partial \beta_1} = \frac{1}{N}\sum (y_i - \widehat{y_i})^2$$

$$\frac{\partial}{\partial \beta_1} = \frac{1}{N}\sum (y_i - \{\beta_0 + (\beta_1 * x_i)\})^2$$

$$\frac{\partial}{\partial \beta_1} = \frac{1}{N}\sum 2(y_i - \{\beta_0 + (\beta_1 * x_i)\}) * \frac{\partial}{\partial \beta_1}(y_i - \{\beta_0 + (\beta_1 * x_i)\})$$

$$\frac{\partial}{\partial \beta_1} = \frac{2}{N}\sum (y_i - \{\beta_0 + (\beta_1 * x_i)\}) * (-x_i)$$

$$\frac{\partial}{\partial \beta_1} = \frac{-2}{N}\sum (x_i)(y_i - \{\beta_0 + (\beta_1 * x_i)\})$$

$$\frac{\partial}{\partial \beta_0} = \frac{1}{N}\sum(y_i - \widehat{y_i})^2$$

$$\frac{\partial}{\partial \beta_0} = \frac{1}{N}\sum(y_i - \{\beta_0 + (\beta_1 * x_i)\})^2$$

$$\frac{\partial}{\partial \beta_0} = \frac{1}{N}\sum 2(y_i - \{\beta_0 + (\beta_1 * x_i)\}) * \frac{\partial}{\partial \beta_0}(y_i - \{\beta_0 + (\beta_1 * x_i)\})$$

$$\frac{\partial}{\partial \beta_0} = \frac{2}{N}\sum(y_i - \{\beta_0 + (\beta_1 * x_i)\}) * (-1)$$

$$\frac{\partial}{\partial \beta_0} = \frac{-2}{N}\sum(y_i - \{\beta_0 + (\beta_1 * x_i)\})$$

Next step is to update the derivatives which help in converging the weights.

$$\beta_1 = \beta_1 - (\alpha * \frac{\partial}{\partial \beta_1}.[Loss\ Function])$$

$$\beta_0 = \beta_0 - (\alpha * \frac{\partial}{\partial \beta_0}.[Loss\ Function])$$

$$\alpha\ is\ known\ as\ learning\ rate\ with\ which\ the\ function\ takes\ steps\ to\ reach\ global\ minima$$

$$\alpha\ is\ extremely\ small\ value\ as\ 0.0001$$

Using principle of iteration, convergence begins {

$$\frac{\partial}{\partial \beta_1} = \frac{-2}{N} \sum (x_i)(y_i - \{\beta_0 + (\beta_1 * x_i)\})$$

$$\frac{\partial}{\partial \beta_0} = \frac{-2}{N} \sum (y_i - \{\beta_0 + (\beta_1 * x_i)\})$$

$$\beta_1 = \beta_1 - (\alpha * \frac{\partial}{\partial \beta_1} . [Loss\ Function])$$

$$\beta_0 = \beta_0 - (\alpha * \frac{\partial}{\partial \beta_0} . [Loss\ Function])$$

}