Linear Regression

(You cannot understand it without knowing partial derivatives!)

First and foremost, we need to consider vanilla linear regression to understand how it works, and then you can apply the same concept to multivariate linear regression.

But, what is vanilla linear regression? Well, it is just linear regression that involves a single feature per feature vector, i.e., the label is dependent only on one feature.

So, as you might have guessed, we will not be getting a matrix for features, but just a sample
array that contains values of the feature for individual samples.

We will approach this Linear Regression problem with a module-based approach. First, we will
understand what the gradient descent algorithm is, and then we will simply put our Regression
model into that.

So, let's understand gradient descent first!

Gradient Descent

Consider a function,

$$y = x^2$$

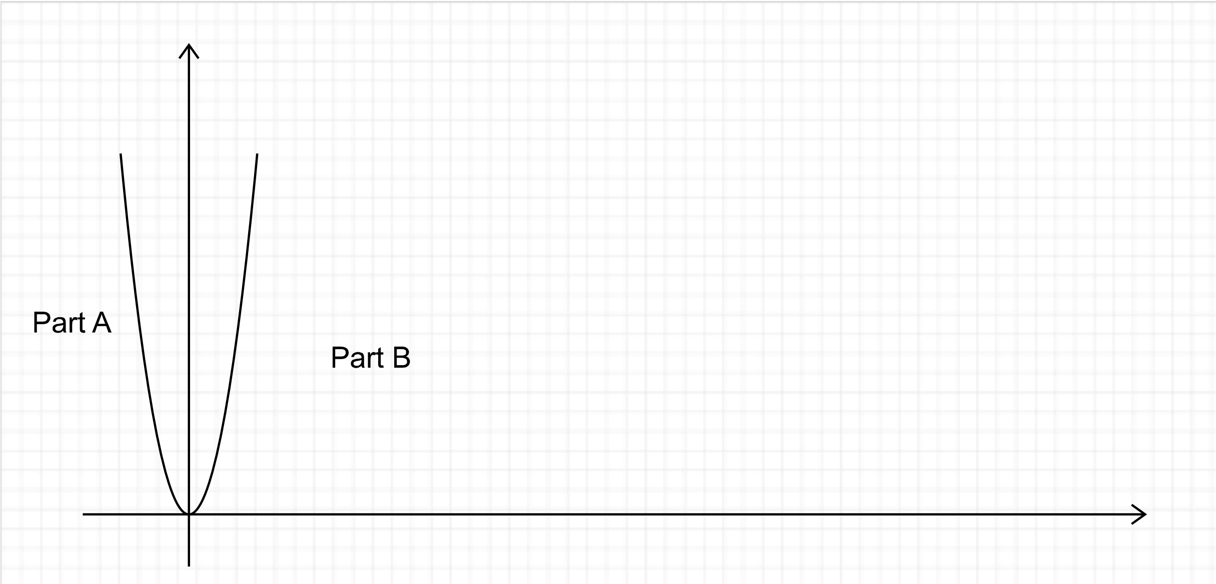
and its derivative with respect to x is,

$$\frac{dy}{dx} = 2x$$

So, this is saying how y is changing with respect to x.

But this is again differentiable, which outputs 2, indicating that the function is convex.

The graph of this function would look like this:



Suppose you don’t know where the function attains its minimum value (I know it is trivial here).
Then what possible options do you have?
What you can do here is just take a random x value (typically we take around zero).
Here, suppose we take the value of x = x1.

Then we put this value into the derivative to calculate the slope, i.e

$$\frac{dy}{dx} = 2x = 2 * x_1$$

Here, the 2x1 value represents the slope. Two conditions are possible:
1.) Positive 2.) Negative
1.) Positive means that the slope is positive, and we are on the increasing part of the function. In order to achieve the global minimum, we must reduce the value of x. That is, we are in part B
and need to reduce x. So we need to calculate the value of x:

$$x_{new} = x_{old} - learningRate \cdot slope$$

(Here, - will remain as it is, thus decreasing the value of the old x).

So here, we get a new x, which we are again going to put into the derivative to calculate the slope and thus calculate the new x value. We will keep doing this until sufficient iterations are met
where we assure that x is closer to the global minimum or we do it until the slope becomes zero,
i.e., no new x values.

2.) Negative means that the slope is negative, and we are on the decreasing part of the function.
In order to achieve the global minimum, we must increase the value of x. That is, we are in part
A and need to increase x. So we need to calculate the value of x:

$$x_{new} = x_{old} - learningRate \cdot slope$$

(Here, - will become + as the slope is negative, thus increasing the value of the old x).

So here, we get a new x, which we are again going to put into the derivative to calculate the slope and thus calculate the new x value (everything else as above).

So how does this help us solve linear regression?

Well, in linear regression, what is important is the Mean Squared Error (MSE). It is given by:

$$MSE = \frac{1}{N} * \sum (y_{pred} - y_{real})^2$$

Here, we calculate the average value of the squared error.
And our target is to minimize this using Gradient Descent!
So, you just need to fit this into the above discussion, see how it fits.
So MSE is the function we need to minimize. As we have talked earlier, as we are considering only one feature, we can represent linear regression with a line that has the equation:

$$y \ = \ mx \ + \ c$$

where m is the slope of the regression line, and c is the y-intercept.
So our ypred is actually the above equation!
So put that in the equation of MSE:

$$MSE \ = \ \frac{1}{N} * \sum ((mx \ + \ c) \ - \ y_{real})^2$$

Now we have two independent variables here: m and c. So here, the need for partial derivatives
comes!
First, keeping c constant and differentiating w.r.t m:

$$\frac{\delta \, MSE}{\delta m} \ = \ \frac{1}{N} * 2 \sum ((mx + c) \ - \ y_{real}) * x$$

Then keeping m constant:

$$\frac{\delta \, MSE}{\delta c} \ = \ \frac{1}{N} * 2 \sum ((mx + c) \ - \ y_{real})$$

So here, in (A), the x multiplied at last is nothing but our feature matrix. So we just now put values of m and c in the above equations and update m and c like:

$$m_{new} \ = \ m_{old} \ - \ learningRate * \frac{\delta \, MSE}{\delta m}$$

$$c_{new} \ = \ c_{old} \ - \ learningRate * \frac{\delta \, MSE}{\delta c}$$

You can find teh **CODE** here : https://github.com/VaradBelwalkar/helpful-tutorials/tree/main/machine_learning/Linear_Regression

If you found the content useful, consider following me on my github: https://github.com/VaradBelwalkar