What is Gradient Descent?

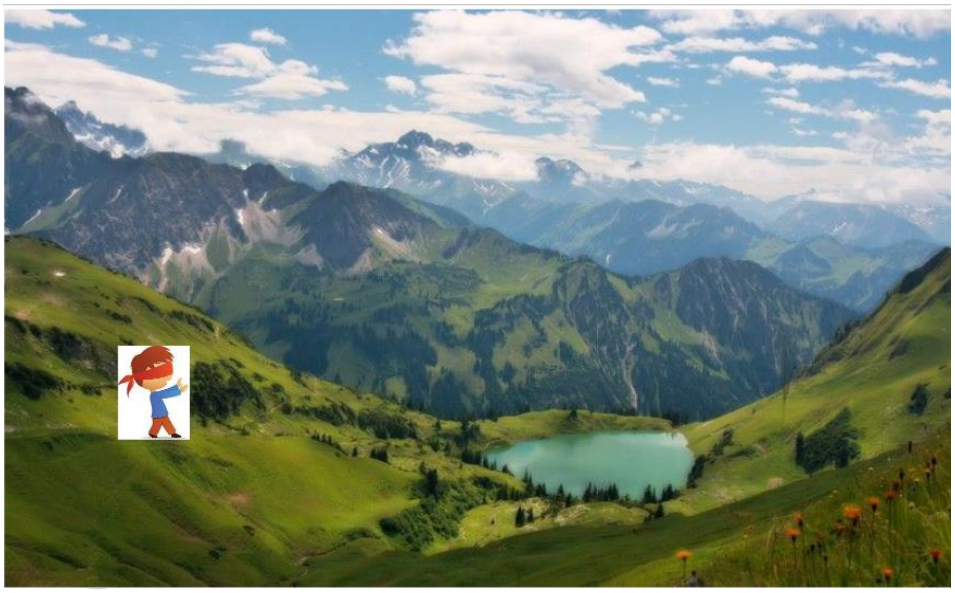
Optimization is a big part of machine learning. Almost every machine learning algorithm has an optimization algorithm at its core.

Gradient descent is an optimization algorithm used to find the values of parameters (coefficients) of a function (f) that minimizes a cost function (cost).\

Gradient descent is best used when the parameters cannot be calculated analytically or when you need an optimized way to calculate those parameters.

Intuition for Gradient Descent

Suppose you are at the top of a mountain, and your goal is to reach a lake which is at the lowest point of the mountain. A twist is that you are blindfolded, and you have zero visibility to see where you are headed.



The best way is to check the ground near you and observe where the land tends to descend. This will give an idea in what direction you should take your first step. If you follow the descending path, it is very likely you would reach the lake.

Any position on the mountain is the cost value evaluated by the current values of the coefficients (values of coefficients corresponding given by that point on bowl).

The lake is the point where the cost will be minimum. The goal is to find this point.

Procedure:

Let the cost function be, f:

The procedure starts by assuming the initial values of coefficients. These could be 0.0 or some random value.

Coefficient, m = 0

Intercept, b = 0

The cost in evaluated by using these coefficients.

Cost = f (m, b)

Now derivative of cost is calculated with respect to the coefficients i.e. slope of the cost function is calculated. We need to know the slope to determine the direction to move the coefficient values to get lower cost at next iteration. The selection will be done based on slope at that point, whether it is positive or negative, just as in the case when you were on the mountain and your decision is based on the ground near you.

Delta = derivative(cost), i.e.

Now to update the coefficient values, a learning rate parameter must be specified that controls how much the coefficients can change on each iteration

New-coefficients = coefficient – (learning-rate*derivative)

The cost function, evaluated using the new coefficients will have a better(lower)/ worse(higher) cost depending upon the value of learning rate.

Repeating this process enough times will give the coefficients that will result in minimum cost. The process can be repeated either a specified number of times or till that point where change in cost is negligible.

Example, in case of Linear Regression with only one feature, cost function is given as:

$$\text{Error}_{(m,b)} = \frac{1}{N}\sum_{i=1}^{N}(y_i - (mx_i + b))^2$$

Now the first step is to assume the value of coefficients,

m = 0

b = 0

Next step is to calculate the cost using these values and find partial derivatives,

$$\frac{\partial}{\partial m} = \frac{2}{N}\sum_{i=1}^{N} - x_i(y_i - (mx_i + b))$$

$$\frac{\partial}{\partial b} = \frac{2}{N}\sum_{i=1}^{N} - (y_i - (mx_i + b))$$

Now with the help of these derivatives and learning rate calculate the new coefficients,

new_m = m – (learning rate * derivative wrt m)

new_b = b – (learning rate * derivative wrt b)

Repeat this process several times to get the minimum cost.

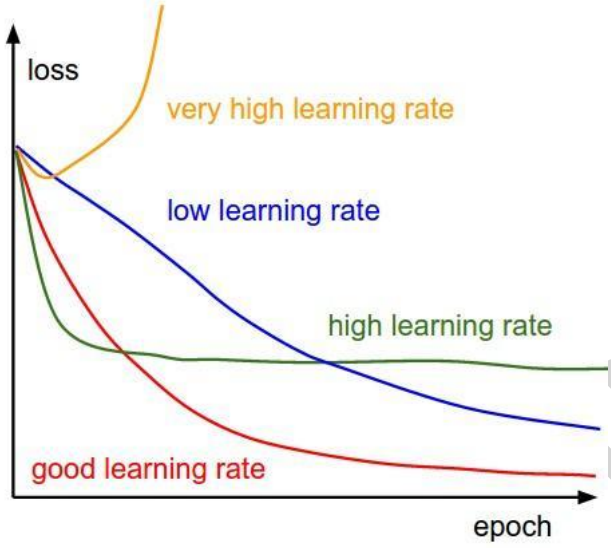
Importance of Learning Rate:

Learning rate controls how much the coefficients can change on each iteration. Therefore, it is very important to decide a good learning rate.

A bad learning rate can make the cost at next iteration higher than the cost at previous iteration, and eventually the cost will become infinity. This process is known as over-shooting.

Generally, learning rates should be as small as possible so that the cost will decrease slowly and eventually reaches its minimum.

The learning rate value is a small real value such as 0.1, 0.001 or 0.0001. Try different values for your problem and see which works best.



Types of grad descent:

1. Batch GD:

    In full batch gradient descent algorithms, we use whole data at once to compute the gradient. This is slow when we have large amount of data.

2. Stochastic DG:

    In stochastic DG, we take a sample of data while computing the gradient. This is used in situations where we have large amount of data.