

# Gradient Descent Algorithm: Methodology

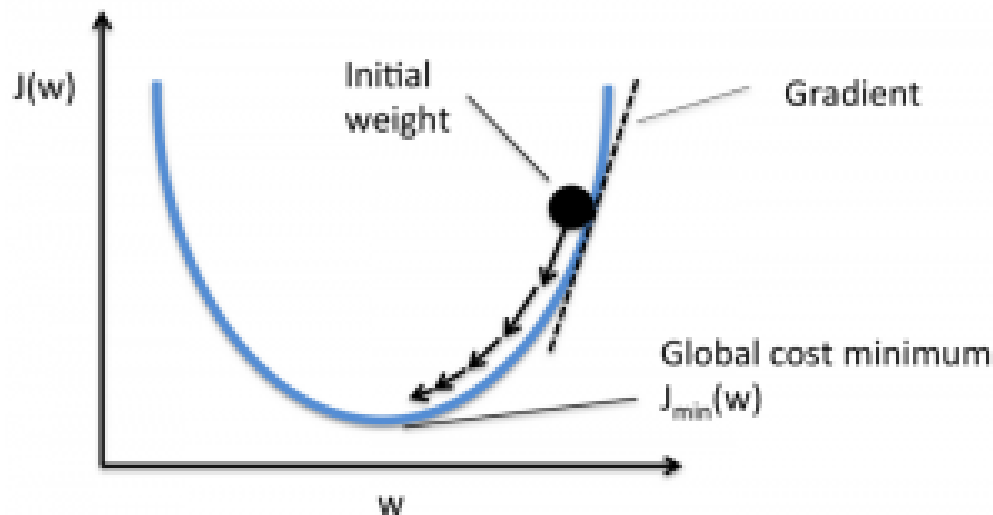
Optimization is an integral part of machine learning. Almost all machine learning algorithms have an optimization function as a crucial segment. As the word suggests, optimization in machine learning is finding the optimal solution to a problem statement.

In this article, you'll read about one of the most widely used optimization algorithms, gradient descent. The **gradient descent algorithm** can be used with any machine learning algorithm and is easy to comprehend and implement. So, what exactly is gradient descent? By the end of this article, you'll have a clearer understanding of the **gradient descent algorithm** and how it can be used to update the model's parameters.

## Gradient Descent

Before going deep into the **gradient descent algorithm**, you should know what cost function is. The cost function is a function used to measure the performance of your model for a given dataset. It finds the difference between your predicted value and expected value, thus quantifying the error margin.

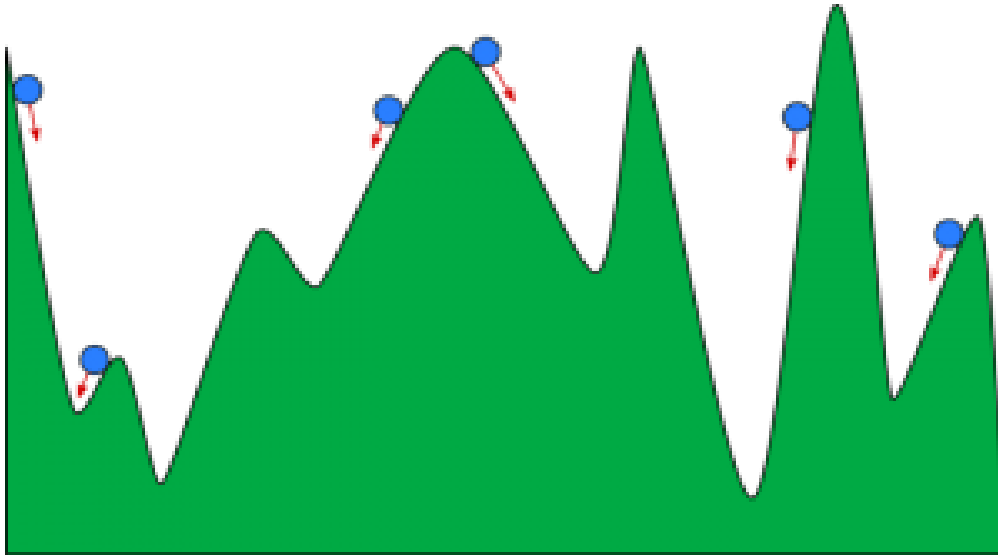
The goal is to reduce the cost function so that the model is accurate. To achieve this goal, you need to find the required parameters during the training of your model. Gradient descent is one such optimization algorithm used to find the coefficients of a function to reduce the cost function. The point at which cost function is minimum is known as global minima.



## The intuition behind the Gradient Descent algorithm

Suppose you have a large bowl similar to something you've your fruit in. This bowl is the plot for the cost function. The bottom of the bowl is the best coefficient for which the cost function is minimum. Different values are used as the coefficients to calculate the cost function. This step is repeated until the best coefficients are found.

You can imagine gradient descent as a ball rolling down a valley. The valley is the plot for the cost function here. You want the ball to reach the bottom of the valley, where the bottom of the valley represents the least cost function. Depending on the start position of the ball, it may rest on many bottoms of the valley. However, these bottoms may not be the lowest points and are known as local minima.



## Gradient Descent Algorithm- Methodology

The calculation of gradient descent begins with the initial values of coefficients for the function being set as 0 or a small random value.  
coefficient = 0 (or a small value)

- The cost function is calculated by putting this value of the coefficient in the function.

Cost function =  $f(\text{coefficient})$

- We know from the concept of calculus that the derivative of a function is the slope of the function. Calculating the slope will help you to figure out the direction to move the coefficient values. The direction should be such that you get a lower cost(error) in the next iteration.

$\text{del} = \text{derivative}(\text{cost function})$

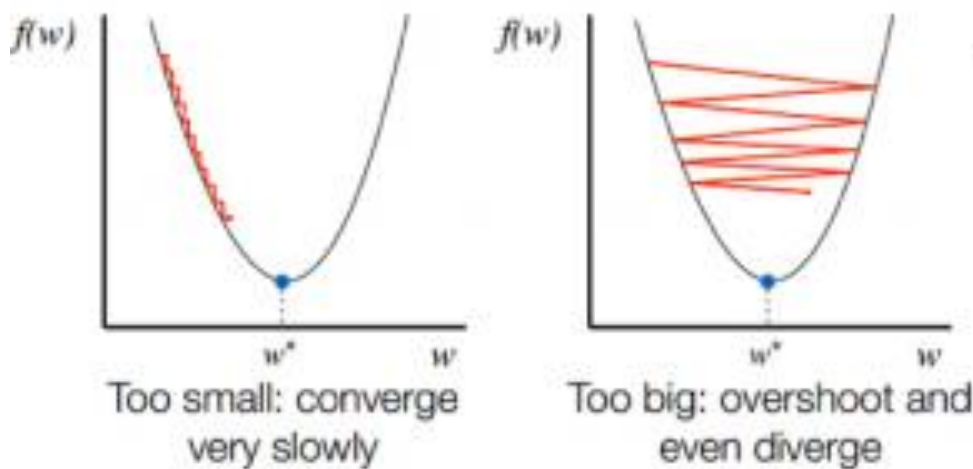
- After knowing the direction of downhill from the slope, you update the coefficient values accordingly. A learning rate (alpha) can be selected to control how much these coefficients will change in each iteration. You need to make sure that this learning rate is not too high nor too low.

$\text{coefficient} = \text{coefficient} - (\text{alpha} * \text{del})$

- This process is repeated until the cost function becomes 0 or very close to 0.

$$f(\text{coefficient}) = 0 \text{ (or close to 0)}$$

The selection of the learning rate is important. Selecting a very high learning rate can overshoot the global minima. On the contrary, a very low learning rate can help you reach the global minima, but the convergence is very slow, taking many iterations.



## Best Practices for Gradient Descent Algorithm

- **Map cost versus time:** Plotting the cost with respect to time helps you visualize whether the cost is decreasing or not after each iteration. If you see the cost to remain unchanged, try updating the learning rate.
- **Learning rate:** The learning rate is very low and is often selected as 0.01 or 0.001. You need to try and see which value works best for you.

## Wrapping up

You get to know the role of gradient descent in optimizing a machine learning algorithm. One important factor to keep in mind is choosing the right learning rate for your **gradient descent algorithm** for optimal prediction.