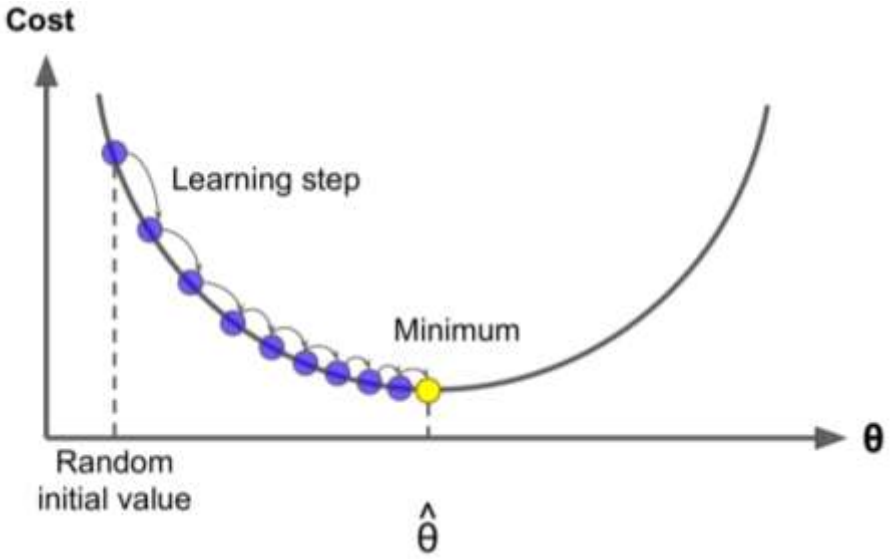
Gradient Descent:

Gradient Descent is a very generic optimization algorithm capable of finding optimal solution to a wide range of problems. The general idea of a Gradient Descent is to tweak parameters iteratively in order to minimize a cost function.

Gradient Descent measures the local gradient of the error function with regards to the parameter vector θ, and it goes in the direction of descending gradient. Once the gradient is zero; you have reached a minimum!

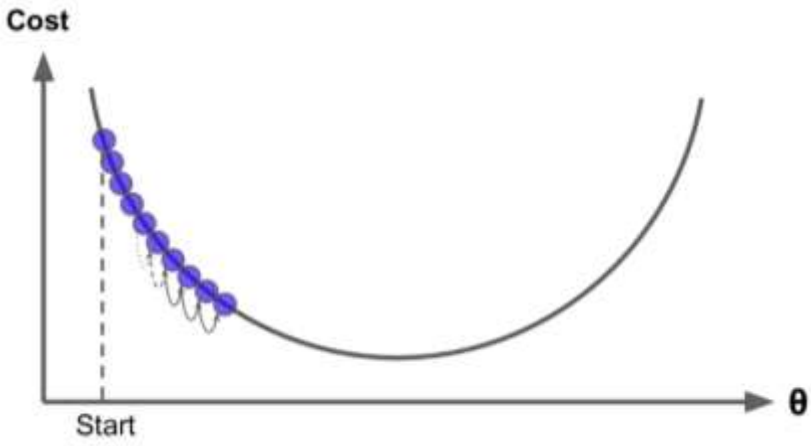
Concretely, you start by θ with random values(this is called random initialization), and then improve it gradually, taking one baby step at a time, each step attempting to decrease the cost function (e.g, the MSE), until the algorithm converges to a minimum.



An important parameter in Gradient Descent is the size of the steps, determined by the learning rate hyperparameter. If the learning rate is too small, then the algorithm will have to go through many iterations to converge, which will take a long time.

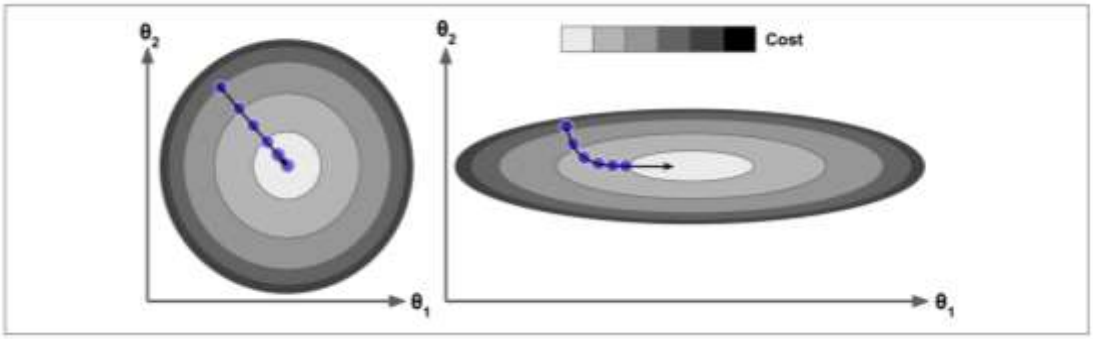
On the other hand, if the learning rate is too high. You must jump across the valley and end up on the other side, possibly even higher up than you were before. This might make the algorithm diverse, with larger and larger values, failing to find a good solution.

Two main challenges with Gradient Descent: if the random initialization starts the algorithm on the left, then it will converge to a local minimum, which is not as good as global minimum. If it starts on the right, then it will take a very long time to cross the plateau, and it you stop too early you will never reach the global minimum.



Fortunately the MSE cost function for a Linear Regression model happens to be a convex function, which means that if you pick any two points on the curve, the line segment joining them never crossed the curve. This implies that there are no local minimum just one global minimum.

In fact, the cost function has the shape of the bowl, but it can be an elongated bowl if the features have very different scales. Figure shows Gradient Descent on a training set where features 1 and 2 have the same scale (on the left), and on a training set where feature 1 has much smaller values than feature 2(on the right).



As you can see, on the left Gradient Descent algorithm goes straight toward the minimum, thereby reaching it quickly, whereas on the right it first goes in a direction almost orthogonal to the direction of the global minimum, and it ends with a long march down an almost flat valley. It will eventually reach the minimum, but it will take a long time.

This diagram also illustrates the fact that training a model means searching for combination of model parameters that minimizes a cost function (over the training set). It is a search in the model's parameter space: the more parameters a model has, the more dimensions this space has, and the harder the search is: