**Gradient Descent**

**For**

**Machine Learning**

## What is Gradient Descent?

Gradient descent is an optimization algorithm which is mainly used to find the minimum of a function. It’s an iterative process that finds the minima of function in machine learning.

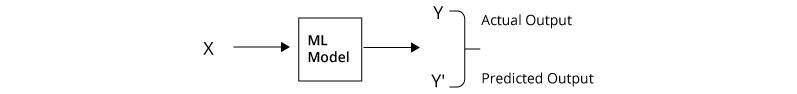
## How does Gradient Descent work?

The primary goal of machine learning algorithms is always to build a model.

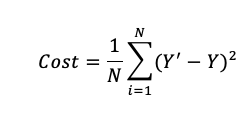
## Cost Function

The objective in the case of gradient descent is to find a line of best fit for some given inputs, or X values, and any number of Y values, or outputs. A cost function is defined as “a function that maps an event or values of one or more variables onto a real number intuitively representing some “cost” associated with the event.”

With a known set of inputs and their corresponding outputs, a machine learning model attempts to make predictions according to the new set of inputs.



Let's say, there are a total of 'N' points in the dataset. So the cost function would be the total squared error

**[](https://lh3.googleusercontent.com/-mUd-RlOHNvs/X4J3PofSXoI/AAAAAAAAAGw/k77MjYRZjboe6I0LsO12csF0RRgYq4oSACLcBGAsYHQ/image.png)**

It is always the primary goal of any Machine Learning Algorithm to minimize the Cost Function. Minimizing cost functions will also result in a lower error between the predicted values and the actual values which also denotes that the algorithm has performed well in learning.

## How do we actually minimize any function?

### Gradient Descent Procedure:

The procedure starts off with initial values for the coefficient or coefficients for the function. These could be 0.0 or a small random value.

coefficient = 0.0

The cost of the coefficients is evaluated by plugging them into the function and calculating the cost.

cost = f(coefficient)

or

cost = evaluate(f(coefficient))

The derivative of the cost is calculated. The derivative is a concept from calculus and refers to the slope of the function at a given point. We need to know the slope so that we know the direction (sign) to move the coefficient values in order to get a lower cost on the next iteration.

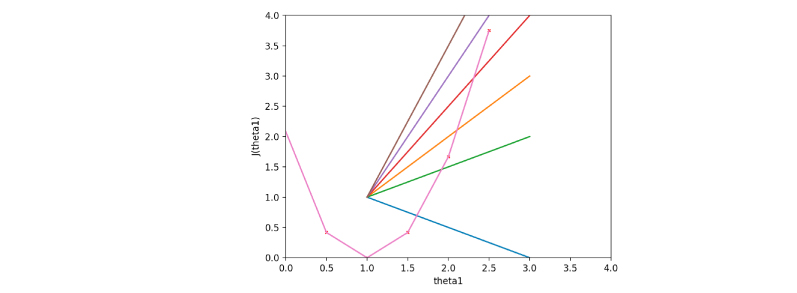
delta = derivative(cost)

Now that we know from the derivative which direction is downhill, we can now update the coefficient values. A [learning rate parameter](https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/) (alpha) must be specified that controls how much the coefficients can change on each update.

coefficient = coefficient – (alpha \* delta)

This process is repeated until the cost of the coefficients (cost) is 0.0 or close enough to zero to be good enough.

After applied mean cost function on each iteration, we will get like this



## Conclusion

When Keep on iterating with the values of m and b, we can see that the best fit line moves towards a position where the error is minimal. We can an animation representing the whole gradient descent.

The point of this article was to demonstrate the concept of gradient descent. We used gradient descent as our optimization strategy for linear regression. by drawing the line of best fit to measure the relationship between student heights and weights. However, it is important to note here that the linear regression example has been chosen for simplicity but can be used with other Machine Learning techniques too.

