Gradient Descent:Idiotic Rigorous Analysis

Akash Manna

May 2021

1 Introduction

The very first question that come to mind of a new bee into the Machine Learning is that:what is gradient descent and how it works in machine learning field.

The sole purpose of this blog is to explain each and every aspect of the topic and investigate pros and cons of different types of Gradient Descent(GD)-Stochastic,Batch and Mini-batch GD.

But,First let's first understand what is gradient Descent?
In one sentence:it is a first order iterative optimization algorithm for finding local minima of a differentiable function.
Not impressed??!!
let me put this in a layman's word:it is a machine learning algorithm to find coordinates of global minima of cost function used while doing back-propagation(an parameter update procedure).

So, fasten your seat-belt and dive into Gradient Descent

2 Intuition for GD

Suppose you are stuck at the top of a mountain and you are trying to get down from the top.There is heavy fog such that visibility is extremely low.Now,you have to go down from the hill.How would you proceed?

You would proceed in the following steps:

step:1)you will look for steepest gradient currently available to you using some instrument depending upon your local position,as,you could not see further due to heavy fog.

2)Once you find out the steepest gradient,you would take your first step in the direction of the steepest gradient.(i.e. down-hill).
Now,now you land into an updated down-hill position of the hill.

Taking the descent repeatedly,using step(1) and (2),you would eventually land to the flat land where you would not find any descent any more.

Congratulations!!You have reached your destination.

But,suppose as you can't see further due to heavy fog, you have landed into a hole not the flat-land ,how would you overcome that difficulty?

The exactly same thing happens in machine learning.

The hill is nothing but the cost function itself and 'you' is replaced by 'the algorithm'.Top of the hill is nothing but the initial coordinates of the parameters.The flat land is nothing but the position of global minima.

Now,suppose,you want to reach the plane(i.e. global minima) before sunset.Then you have to take bigger step size.

In this analogy,you represent the algorithm and the path taken represents a set of parameters that the algorithm will explore.The steepness of the hill represents the slope of the error surface at that point.The instrument used to determine the slope of the error surface is 'Differentiation' and the step size is the learning rate.

And when the whole bunch of stuff that have been discussed above are taken by the machine is called Gradient Descent.

But,wait,what about that hole?

ohh,that hole is nothing but a local minima,how to get rid of that ambiguity,I will discuss that while discussing different types of GD.

3 Mathematical Analysis

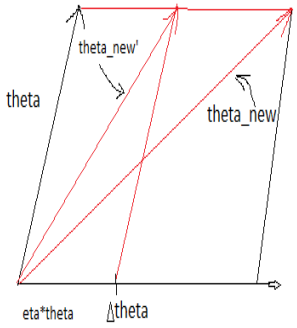
Motto: Finding the most efficient way to navigate over the error surface to reach the minimum value of the error function.

• Let's,there are two randomly initialised vector parameters(i.e. position of the top of the hill)

$$\{\theta = [w, b] \mid \theta \in R^2\}$$

•Then,suppose,there is a small changes in θ such that $\Delta\theta = [\Delta w, \Delta b]$,so that the value of the error function reduces at $\theta \pm \Delta\theta$.(i.e. as soon as you go to $\theta \pm \Delta\theta$,you have gone down slightly.)

Lets say,you are more cautious while taking steps and don't want to take a giant step but want a small step.That is instead of taking $\Delta\theta$ at once,move only by a small amount $\eta \times \Delta\theta$,where,η less than 1.



$$\theta_{new} = \theta \pm \eta \times \Delta\theta$$

where,$0 \leq \eta \leq 1$

$\otimes$**But,how to determine the direction of the updated** θ**?**(i.e either θ should increase or decrease.)

To find the answer of that question:

From Taylor Series expansion of $J(\theta + \eta\Delta\theta)$ about θ:

$$\mathrm{J}(\theta + \eta\Delta\theta) =$$
$$J(\theta) + (\eta\Delta\theta) \times \nabla_\theta J(\theta + \eta\Delta\theta)|_{(\theta+\eta\Delta\theta)=\theta} + \frac{(\eta\Delta\theta)^2}{2} \times \nabla^2_\theta J(\theta + \eta\Delta\theta)|_{(\theta+\eta\Delta\theta)=\theta} + \cdots$$

As,η is very small $\eta^2 \to 0$

$\Rightarrow J(\theta + \eta\Delta\theta)$=$J(\theta) + \eta \cdot \Delta\theta\nabla_\theta J(\theta)$

- Note that,the move $(\eta \cdot \Delta\theta)$ would be favourable if,

$$\mathrm{J}(\theta + \eta \cdot \Delta\theta) - J(\theta) \leq 0$$

(i.e. new loss must be less than the previous loss)

So,

$$\begin{gathered}\mathrm{J}(\theta + \eta \cdot \Delta\theta) - J(\theta) \leq 0 \\ \Rightarrow \eta \cdot \Delta\theta \cdot \nabla_\theta J(\theta) \leq 0 \\ \Rightarrow \Delta\theta \cdot \nabla_\theta J(\theta) \leq 0 (as, \eta \leq 0)\end{gathered}$$

- Let,β be the angle between $\Delta\theta$ and $\nabla_\theta J(\theta)$

$$\begin{gathered}\text{-1} \leq cos(\beta) = \frac{\Delta\theta \cdot \nabla_\theta J(\theta)}{||\Delta\theta|| \cdot ||\nabla_\theta J(\theta)||} \leq 1 \\ \|\Delta\theta|| \cdot ||\nabla_\theta J(\theta)|| = k \\ \Rightarrow -k \leq k \cdot cos(\beta) = \Delta\theta \cdot \nabla_\theta J(\theta) \leq k\end{gathered}$$

To find minimum of $\Delta\theta \cdot \nabla_\theta J(\theta)$ implies

$$\begin{gathered}\mathrm{k} \cdot cos(\beta) = -k \\ \Rightarrow \beta = 180^\circ\end{gathered}$$

$\beta = 180^\circ$ implies that the direction of $\Delta\theta$ must be in the opposite to $\nabla_\theta J(\theta)$

$\boxed{\theta_{new} = \theta - \eta \cdot \Delta\theta}$

4 Algorithm

To implement Gradient Descent,here is some pseudo code for GD

$$\begin{gathered}\mathrm{t}{=}0 \\ \text{max-iter}{=}1000 \\ \text{while t} \leq max - iter\ do \\ \mathrm{w}_{t+1} \leftarrow w_t - \eta\Delta w_t \\ \mathrm{b}_{t+1} \leftarrow b_t - \eta\Delta b_t\end{gathered}$$

5 Types of GD

Depending upon way of calculating the cost function,there are three types of GD

5.1 Stochastic GD

In this type of GD,algorithm takes one training example randomly.

1)Doing forward propagation of that picked example.

2)The cost function is being calculated

. 3) That loss function(=cost function in this case) is being used to update the parameters.

But,there are some cons of Stochastic GD:

1)As,the training examples are being taken randomly,Stochastic GD is noisy update in nature.

2)Computationally expensive:Say we have 10,000 data points and 10 features. The sum of squared residuals consists of as many terms as there are data points, so 10000 terms in our case. We need to compute the derivative of this function with respect to each of the features, so in effect we will be doing 10000 * 10 = 100,000 computations per iteration. It is common to take 1000 iterations, in effect we have 100,000 * 1000 = 100000000 computations to complete the algorithm. That is pretty much an overhead and hence gradient descent is slow on huge data.

5.2 Batch GD

In this type of GD,

1)All the training examples are being forward propagated.

2)Cost function(sum of loss function of each training example) is being calculated.

3)Cost function is being used to update the parameters.

Pros:

1)The updates are less noisy.

2)The magnitude of convergence is high.

Cons:

1)Time taken for the convergence is higher.

2)A,this is less noisy in nature,if the parameter get stucked into a local minima,parameters can't get go out of that local minima.

5.3 Mini-Batch Gradient Descent

Mini-batch gradient descent seeks to find a balance between the robustness of stochastic gradient descent and the efficiency of batch gradient descent. It is the most common implementation of gradient descent used in the field of deep learning.

Pros:

1)Time and space taken for the MBGD is less than that of the Stochastic GD.
2)As,it is more noisy than that of the Batch Gradient Descent,The updates are less likely to get stucked into the local minima.

Cons:

1)Mini-batch requires the configuration of an additional "mini-batch size" hyperparameter for the learning algorithm.

2)Error information must be accumulated across mini-batches of training examples like batch gradient descent.

The hole issue can be overcome by both Stochastic and MBGD method.

6 Conclusion

Every of type of GD has its own pros and cons.So,you have to decide which type of GD is suitable for your data set,wisely.