Gradient Descent

By: Ryan Kawamura

Preface

My name is Ryan Kawamura and I am a third-year student at UCLA studying Data Theory. I am currently taking Math 118 - Mathematical Methods in Data Theory. As the title of the course suggests, we are learning about different mathematical methods commonly used in Data Science. One of the methods we learned about is Gradient Descent. This is one of the concepts that was harder for me to grasp, so I decided to take this opportunity to write a blog post about Gradient Descent to help myself learn more about it and get a better understanding of what is going on in the method. I decided to research and write about different aspects of Gradient Descent such as the history behind it, how it works, what it reveals about data, and examples of its use in Data Science.

Introduction

Gradient descent is an optimization algorithm commonly used in Data Science to train machine learning models and neural networks. It is a step-by-step procedure used to find the minimum of a given function, called the cost function, which allows data scientists to minimize errors between predicted and actual results. The general idea of Gradient Descent is that it is an algorithm that moves in steps towards the direction of steepest descent in order to reach the minimum of the function.

History of Gradient Descent

Gradient Descent was first suggested by Augustin-Louis Cauchy on October 18, 1847. Being a mathematician and physicist, Augustin-Louis Cauchy originally created the idea of gradient descent in an attempt to be able to solve large, dense, and complex calculations and functions in Astronomy. In the 19th century, solving a multivariate system of equations was a very tedious and complicated task. Now, the gradient descent has developed to be used to train machine learning models

Augustin Louis Cauchy (Source: Wikipedia)

How does Gradient Descent Work?

What is a Gradient?

A gradient is a function that takes the partial derivative with respect to the specified variable of the function. In the case of the Gradient Descent, we use the gradient to find the direction of greatest descent by taking the partial derivative of the function with respect to each variable.

$$\underset{\text{Gradient of function f}}{\nabla f(x_0, x_1, x_2, ..., x_n)} = \begin{bmatrix} \frac{\partial f}{\partial x_0}(x_0, x_1, x_2, ..., x_n) \\ \frac{\partial f}{\partial x_1}(x_0, x_1, x_2, ..., x_n) \\ \frac{\partial f}{\partial x_2}(x_0, x_1, x_2, ..., x_n) \\ \dots \\ \frac{\partial f}{\partial x_n}(x_0, x_1, x_2, ..., x_n) \end{bmatrix} \begin{matrix} \text{derivative of } f \text{ with respect to } x_0 \\ \text{derivative of } f \text{ with respect to } x_1 \\ \text{derivative of } f \text{ with respect to } x_2 \\ \\ \text{derivative of } f \text{ with respect to } x_n \end{matrix}$$

Source: Carolina Bento

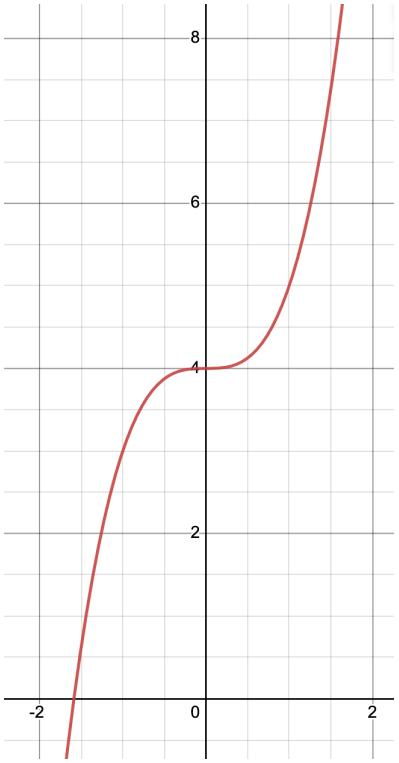
However, finding simply the gradient of a function, points in the direction of ascent, thus we must take the negative gradient to get the direction of steepest descent.

Requirements for Gradient Descent

Before using the method of Gradient Descent, you must first ensure that the function you are analyzing must meet some requirements. The function that we are finding the gradient descent of must be:

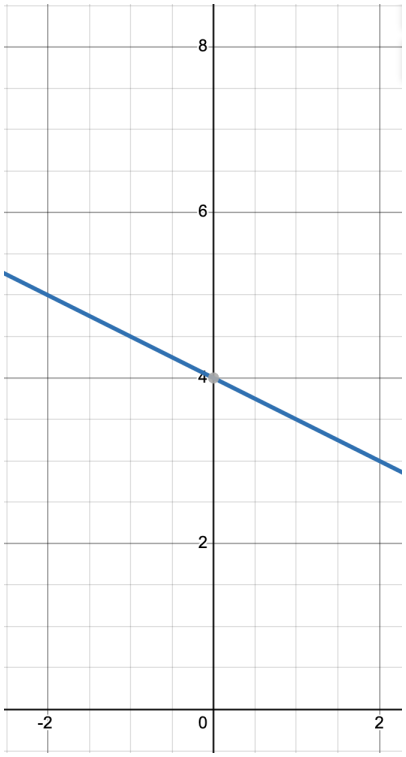
1. Differentiable
2. Convex.

A differentiable function is when there exists a derivative at each point in the domain of the function. Examples of differentiable functions are:



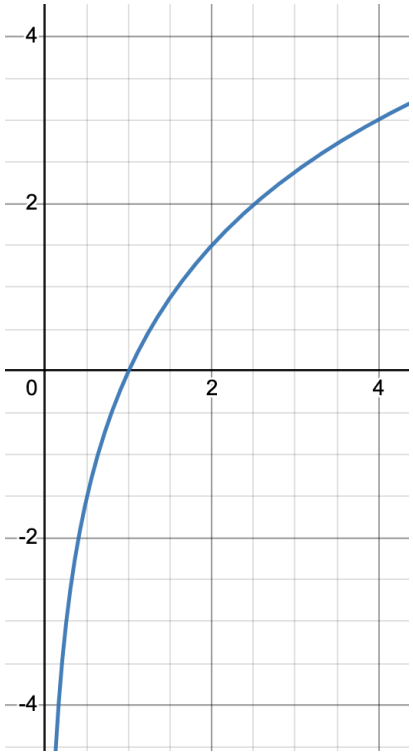
$f(x) = x^3 + 4$

$\frac{d}{dx} f(x) = 3x^2$



$f(x) = -\frac{1}{2}x + 4$

$\frac{d}{dx} f(x) = -\frac{1}{2}$



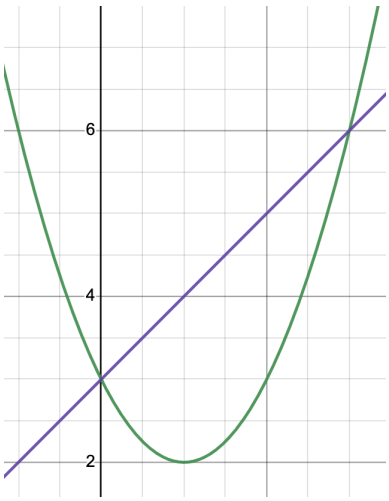
$f(x) = 5 \log x\,;\ x \in [0, \infty]$

$\frac{d}{dx} f(x) = \frac{5}{x}\,;\ x \in [0, \infty]$

The next requirement is that the function be a convex function. Professor Lara defined in lecture that a function $f: R^d \rightarrow R$ with domain D is convex if D is a convex set and if for all vectors $x, y \in D$ and $0 \leq \alpha \leq 1$, we have:

$$f(\alpha x + (1 - \alpha)y) \leq (1 - \alpha)f(y)$$

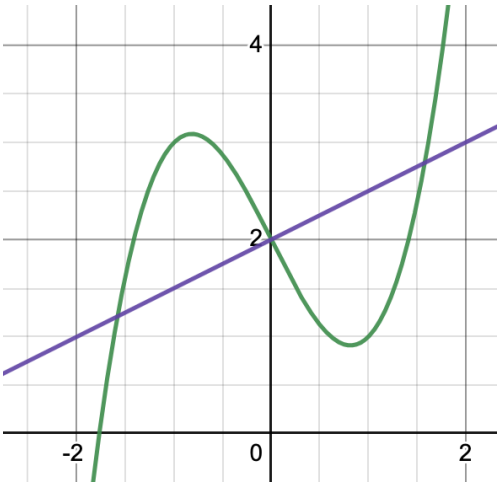
An example of a convex function is



$$f(x) = (x - 1)^2 + 2$$

where the intersecting points are (x,f(x)) and (y,f(y)).

An example of a function that is not convex is



$$f(x) = x^3 + 2x + 2$$

The reason why we must make sure that the function is convex is because if the function is not convex, then the gradient descent might potentially find a local minimum of the function instead of the global minimum.

Function

$$x_k = x_{k-1} - \alpha \nabla f(x_{k-1})$$ where α is the learning rate.

How to Solve

Before we begin, we will first choose an initial point within the domain x_0, a learning rate α, and a stopping criteria, in this case we will use a maximum number of iterations K. We will calculate the $\nabla f(x_0)$ to get the gradient of the function at x_0, which we remember is the direction of the greatest ascent. Then, we will multiple this by the learning rate α, to get the size of the "step" we will take. Then, we will subtract this from the initial point to get the next point x_1 that is in the direction of greatest descent and thus towards the minimum. We will continue these steps K number of times, at which point x_k will theoretically be the x coordinate of the minimum of f (or close to it).

Alternative Stopping Criteria

The Gradient Descent needs a stopping criteria otherwise the method will continue an infinite loop since there is nothing informing when the function should end. Although in the previous section, we chose the maximum number of iterations as the stopping criteria, there are other things that could be used as stopping criteria. The possible stopping criterion include:

1. Running until the norm gradient reaches below a threshold ε

   $$||\nabla f(x_n)||_2 \leq \varepsilon \text{ where } \varepsilon > 0$$

2. Running until there is a change in the cost function that is below a threshold ε

   $$|f(x_k) - f(x_{k-1})| \leq \varepsilon$$

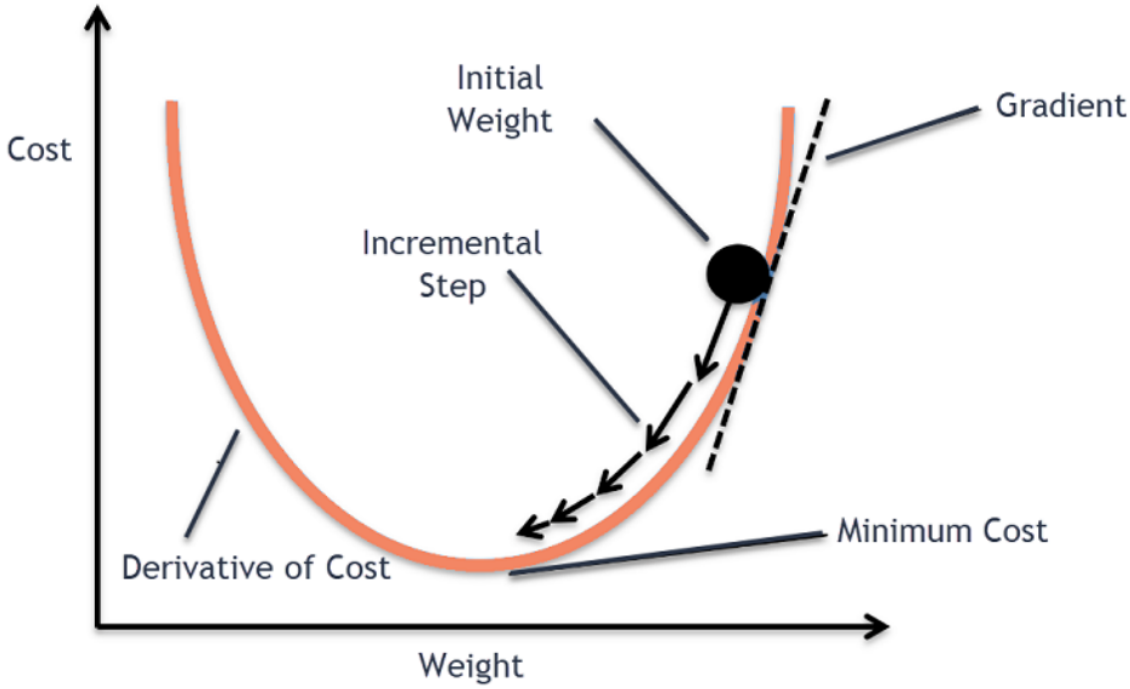
3. Running until there is a change in the gradient of the cost function below a threshold ε

   $$|\nabla f(x_k) - \nabla f(x_{k-1})| < \varepsilon$$

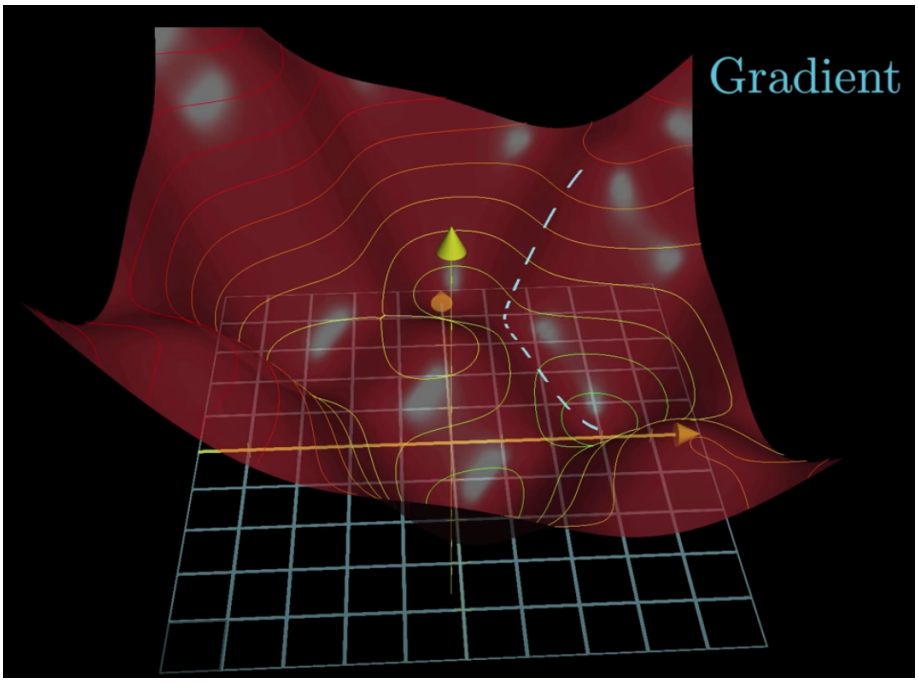
4. Reaching a maximum number of iterations

   $$k < K$$

Visualizations of Gradient Descent



Source: Clairvoyant



Source: 3Blue1Brown

Pseudocode for Gradient Descent:

Inputs: α: learning rate (step size)

K: max number of iterations (or another stopping criteria)

x_0: initial point

For k = 1,...,K:

$$x_k = x_{k-1} - \alpha \nabla f(x_{k-1})$$

Output: x_k

How does Gradient Descent Apply to Data Science, Machine Learning and Mathematical Modeling?

Gradient Descent is an iterative function that goes in the direction of steepest descent in order to efficiently reach the minimum value of your function. The function we wish to minimize is called the cost function. In Machine Learning, the cost function is the model that the data scientist wishes to analyze. One way in which Gradient Descent is utilized in Data Science and Machine Learning is through finding how well the model fits the training data. The more minimized the cost function is, the better the model fits the training data. Gradient Descent helps test the cost function to see how accurate it is at predicting data, which helps Data Scientists create a more accurate model.

Resources

Lemaréchal, Claude. *Cauchy and the Gradient Method*. https://www.math.uni-bielefeld.de/documenta/vol-ismp/40_lemarechal-claude.pdf.

Kwiatkowski, Robert. “Gradient Descent Algorithm - a Deep Dive.” *Medium*, Towards Data Science, 13 July 2022, towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21.

“Gradient Descent, How Neural Networks Learn | Chapter 2, Deep Learning.” *YouTube*, YouTube, 16 Oct. 2017, www.youtube.com/watch?v=IHZwWFHWa-w&ab_channel=3Blue1Brown. Accessed 2 Dec. 2022.

IBM Cloud Education. “What Is Gradient Descent?” *IBM*, www.ibm.com/cloud/learn/gradient-descent.