Understanding Optimization Algorithms in Machine Learning

Supriya Secherla · Follow

Published in Towards Data Science

5 min read · Jun 18, 2021

Mathematics behind two important optimization techniques in machine learning

Table of Contents:

1. INTRODUCTION
2. MAXIMA AND MINIMA
3. GRADIENT DESCENT
4. LEARNING RATE
5. GRADIENT DESCENT IN LOGISTIC REGRESSION
6. STOCHASTIC GRADIENT DESCENT
7. CONCLUSION
8. REFERENCES

Image by Author

. . .

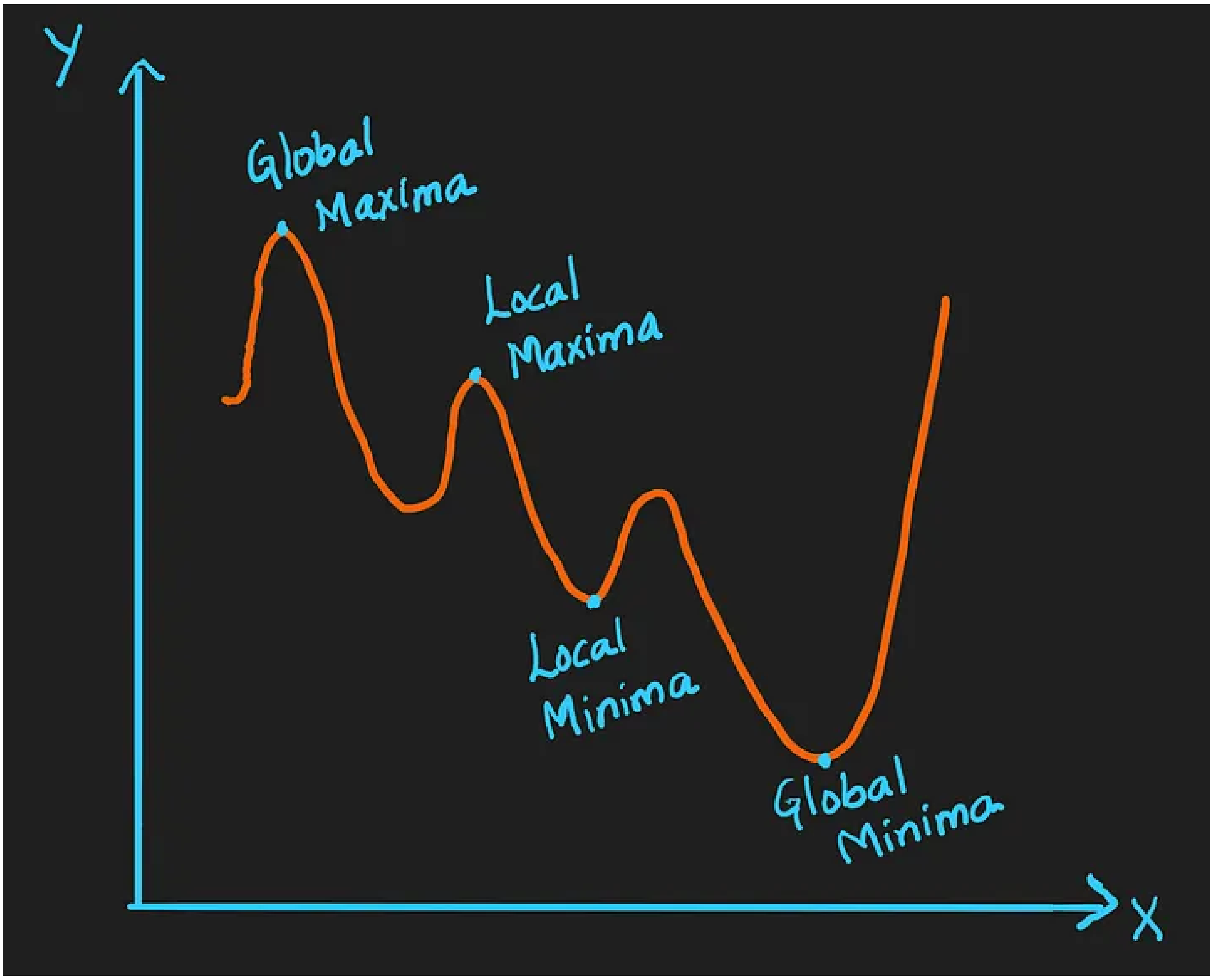
1. **INTRODUCTION**

Optimization is the process where we train the model iteratively that results in a maximum and minimum function evaluation. It is one of the most important phenomena in Machine Learning to get better results.

Why do we optimize our machine learning models? We compare the results in every iteration by changing the hyperparameters in each step until we reach the optimum results. We create an accurate model with less error rate. There are different ways using which we can optimize a model. In this article, let's discuss two important Optimization algorithms: **Gradient Descent and Stochastic Gradient Descent Algorithms**; how they are used in Machine Learning Models, and the mathematics behind them.

2. MAXIMA AND MINIMA

Maxima is the largest and Minima is the smallest value of a function within a given range. We represent them as below:



Minima and Maxima (Image by Author)

Global Maxima and Minima: It is the maximum value and minimum value respectively on the entire domain of the function

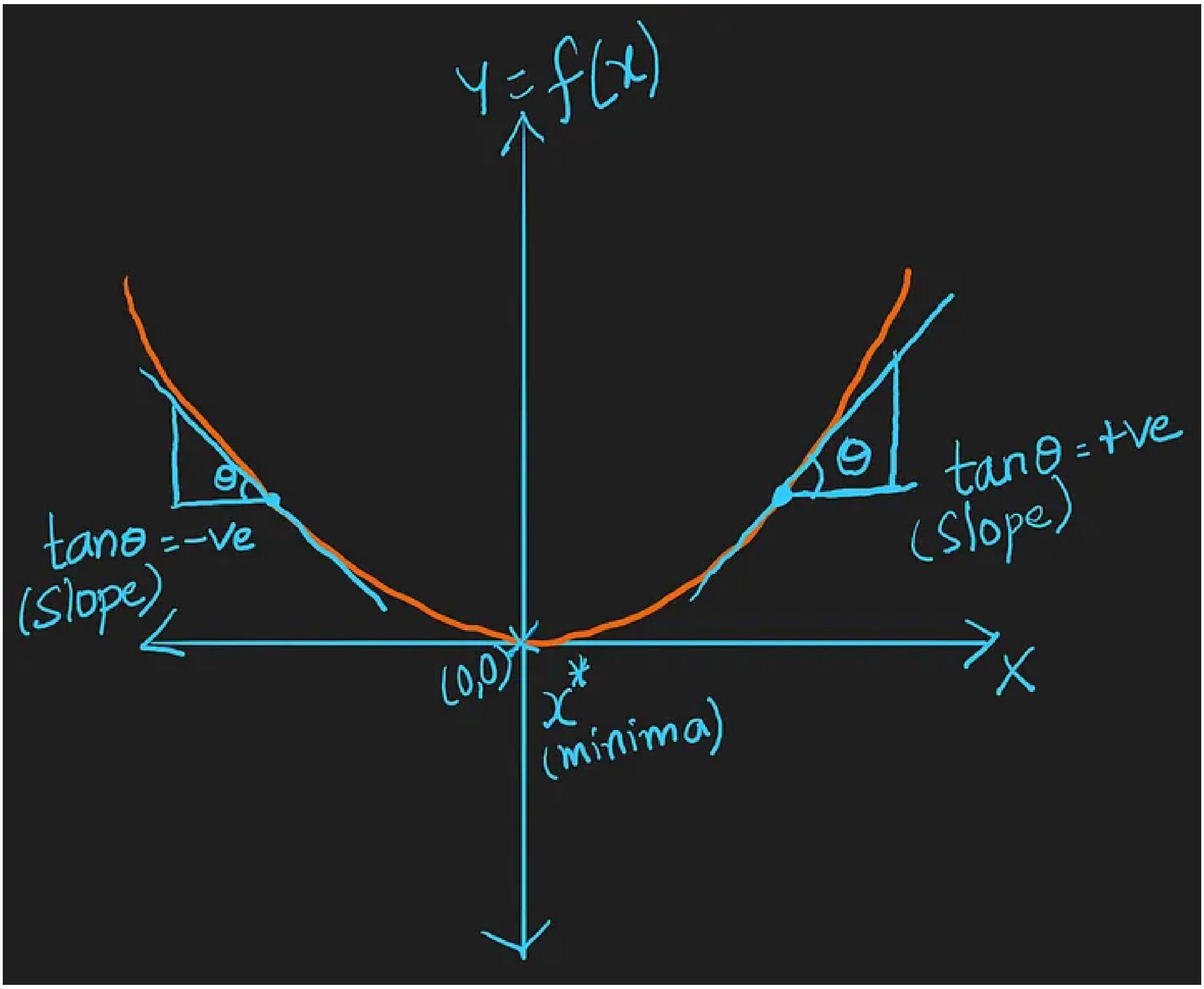
Local Maxima and Minima: It is the maximum value and minimum value respectively of the function within a given range.

There can be only one global minima and maxima but there can be more than one local minima and maxima.

3. GRADIENT DESCENT

Gradient Descent is an optimization algorithm and it finds out the local minima of a differentiable function. It is a minimization algorithm that minimizes a given function.

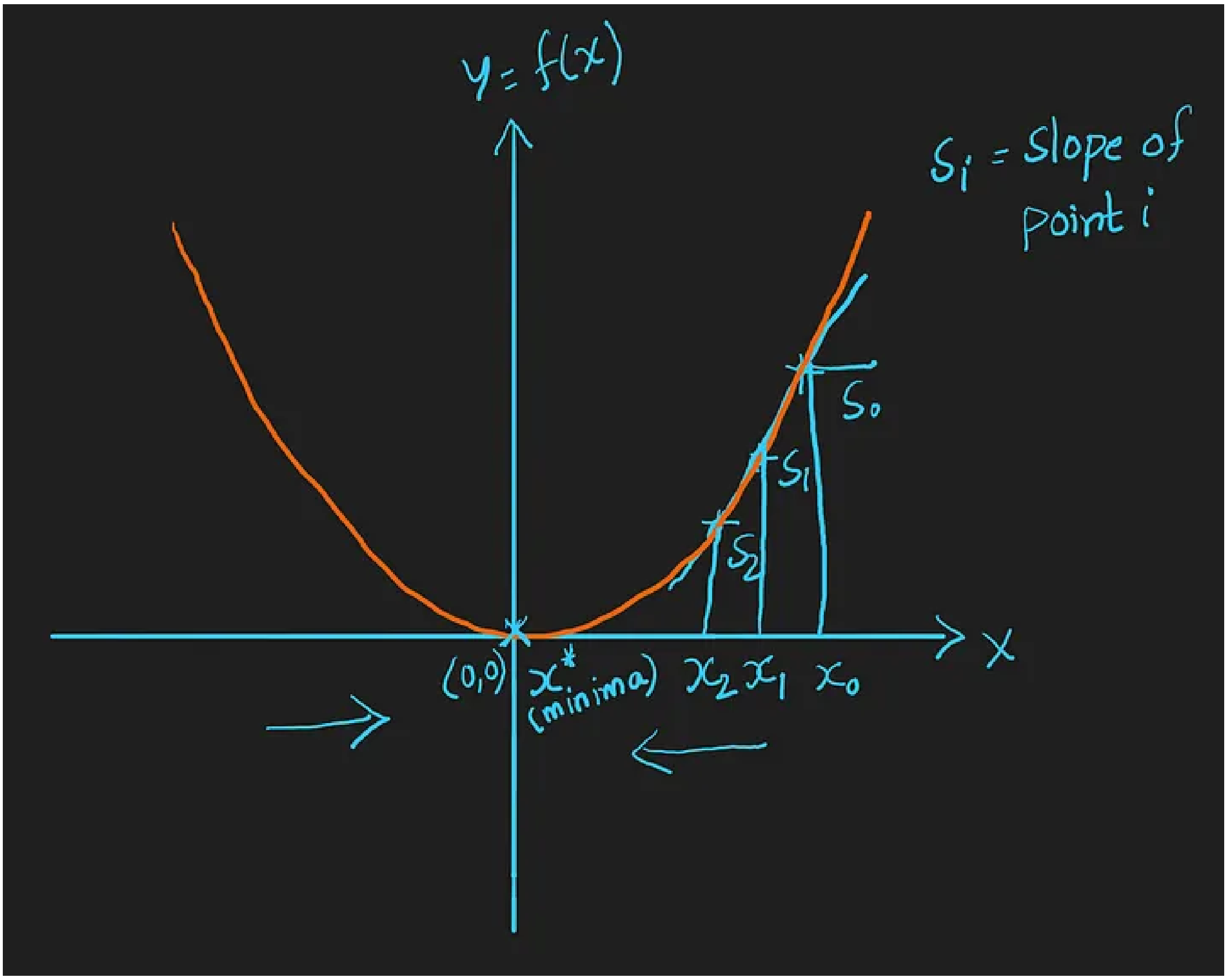
Let’s see the geometric intuition of Gradient Descent:



Slope of Y=X² (Image by Author)

Let’s take an example graph of a parabola, Y=X^2

Here, the minima is the origin(0, 0). The slope here is Tanθ. So the slope on the right side is positive as 0<θ<90 and its Tanθ is a positive value. The slope on the left side is negative as 90<θ<180 and its Tanθ is a negative value.



Slope of points as moved towards minima (Image by Author)

One important observation in the graph is that the slope changes its sign from positive to negative at minima. As we move closer to the minima, the slope reduces.

So, how does the Gradient Descent Algorithm work?

Objective: Calculate X*- local minimum of the function Y=X^2.

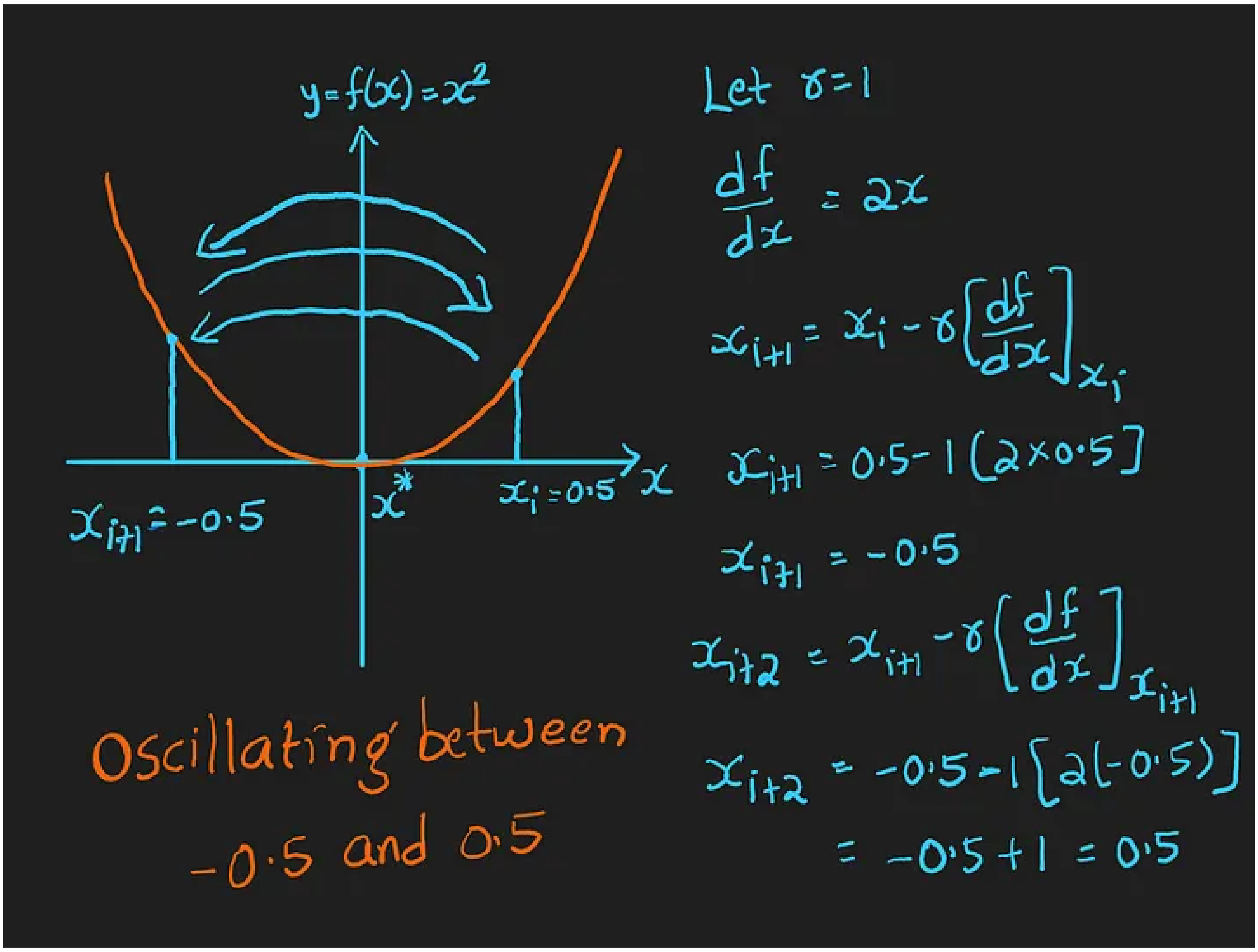
- Pick an initial point X$_0$ at random
- Calculate X$_1$ = X$_0$-r[df/dx] at X$_0$. r is Learning Rate (we'll discuss *r* in Learning Rate Section). Let us take r=1. Here, df/dx is nothing but the *gradient.*

- Calculate $X_2 = X_1$-r[df/dx] at X_1.
- Calculate for all the points: X_1, X_2, X_3, ……., X_{i-1}, X_i
- General formula for calculating local minima: $X_i = (X_{i-1})$-r[df/dx] at X_{i-1}
- When (X_i — X_{i-1}) is small, i.e., when X_{i-1}, X_i converge, we stop the iteration and declare X* = X_i

4. LEARNING RATE

Learning Rate is a hyperparameter or tuning parameter that determines the step size at each iteration while moving towards minima in the function. For example, if r = 0.1 in the initial step, it can be taken as r=0.01 in the next step. Likewise it can be reduced exponentially as we iterate further. It is used more effectively in deep learning.

What happens if we keep r value as constant:



$y=f(x)=x^2$
$x_{i+1}=-0.5$
x^*
$x_i=0.5$
x
Oscillating between
-0.5 and 0.5
Let r=1
$\frac{df}{dx} = 2x$
$x_{i+1} = x_i - r\left[\frac{df}{dx}\right]_{x_i}$
$x_{i+1} = 0.5-1(2\times0.5]$
$x_{i+1} = -0.5$
$x_{i+2} = x_{i+1} - r\left(\frac{df}{dx}\right]_{x_{i+1}}$
$x_{i+2} = -0.5-1[2(-0.5)]$
$= -0.5+1 = 0.5$

Oscillation Problem (Image by Author)

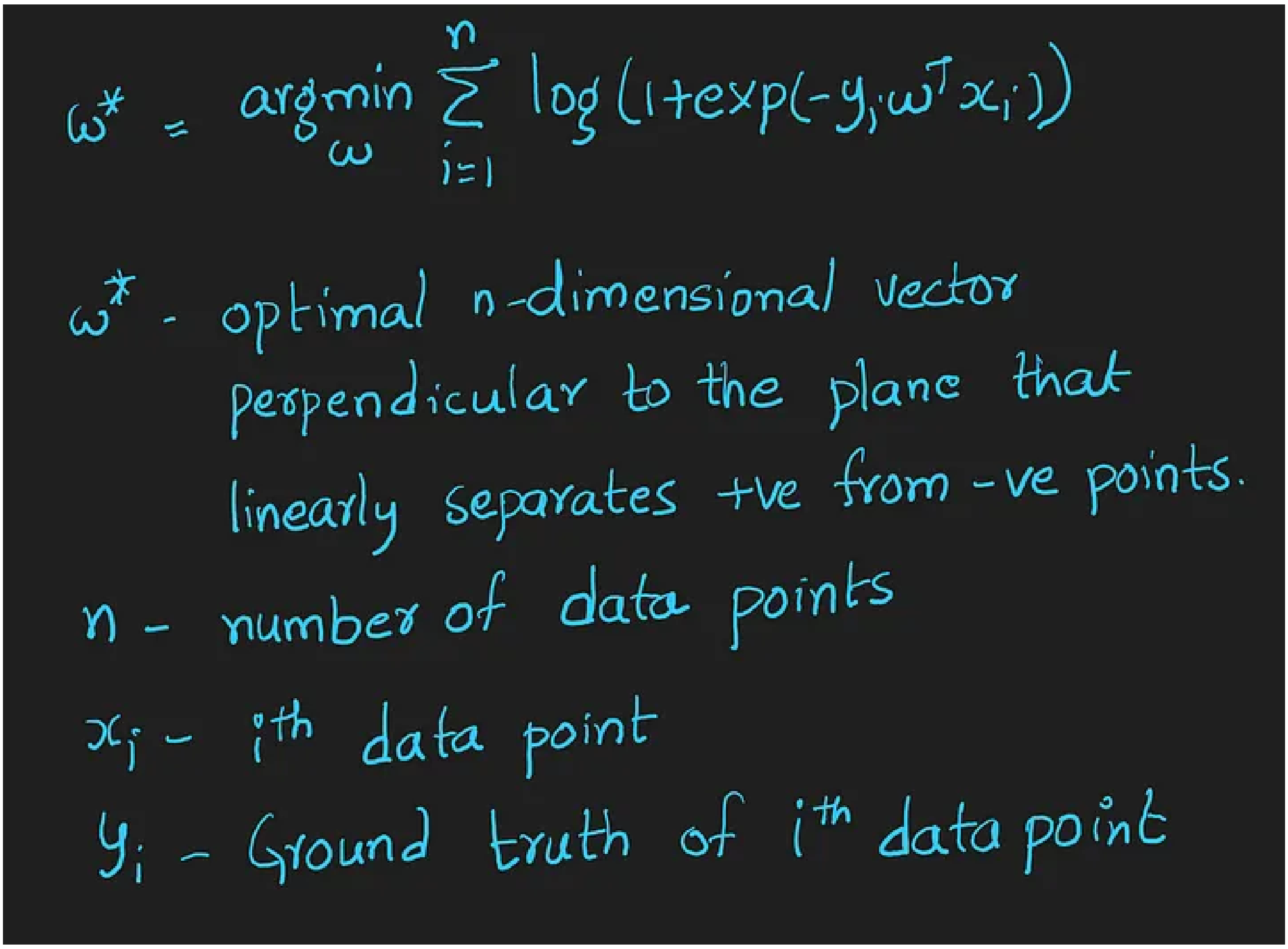
In the above example, we took r=1. As we calculate the points X_i, X_i+_1, X_i+_2,....to find the local minima, X*, we can see that it is oscillating between X = -0.5 and X = 0.5.

When we keep r as constant, we end up with an *oscillation problem*. So, we have to reduce the "r" value with each iteration. Reduce the *r* value as the iteration step increases.

Important Note: Hyperparameters decide the bias-variance tradeoff. When *r* value is low, it could overfit the model and cause high variance. When *r* value is high, it could underfit the model and cause high bias. We can find the correct *r* value with *Cross Validation* technique. Plot a graph with different learning rates and check for the training loss with each value and choose the one with minimum loss.

5. GRADIENT DESCENT IN LOGISTIC REGRESSION

The formula for the optimal plane in logistic regression after applying sigmoid function is:



$$w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} \log(1+\exp(-y_i w^T x_i))$$

w^* - optimal n-dimensional vector perpendicular to the plane that linearly separates +ve from -ve points.

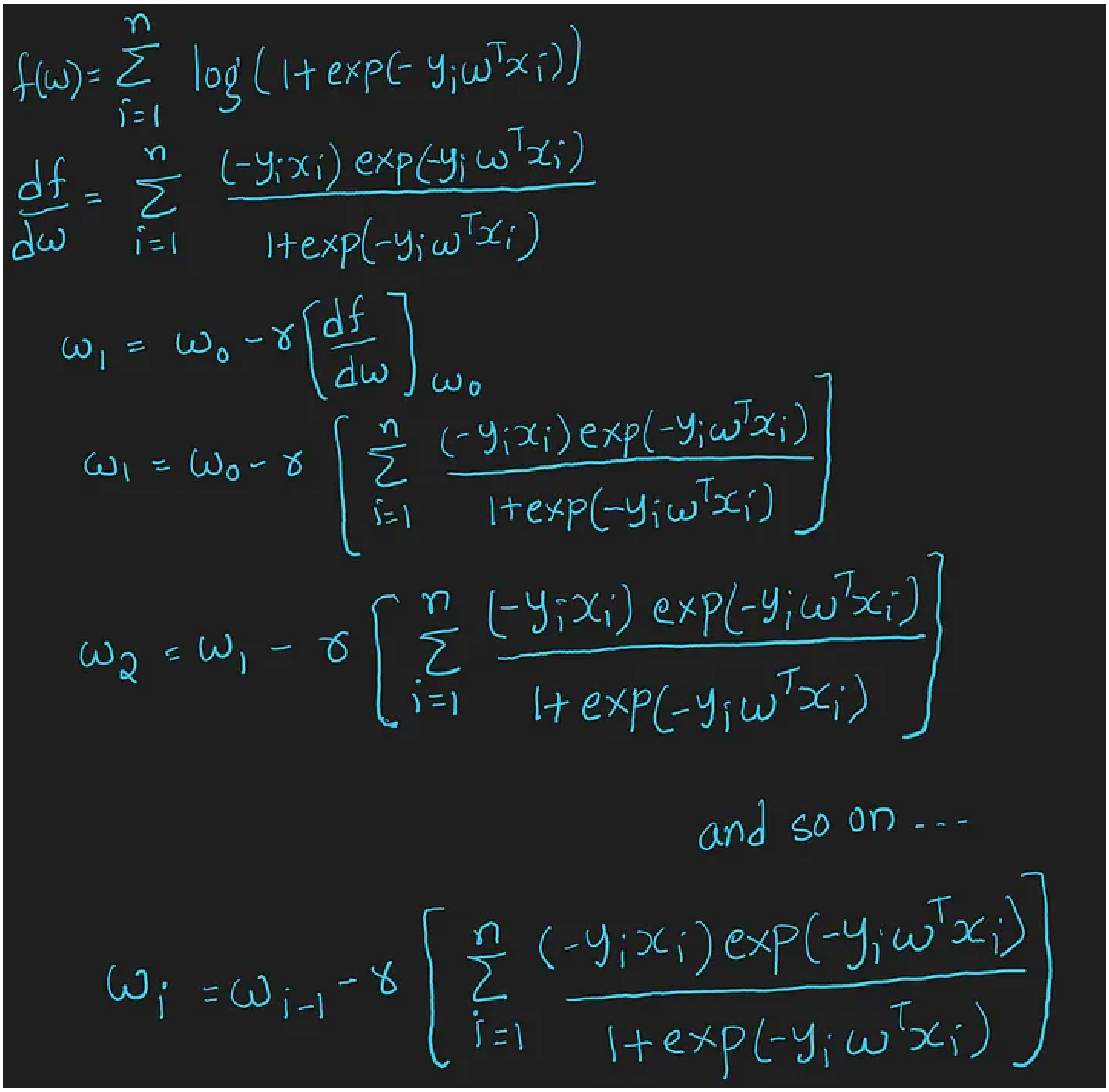
n - number of data points

x_i - i^{th} data point

y_i - Ground truth of i^{th} data point

Optimal Plane — Logistic Regression (Image by Author)

Apply Gradient Descent Algorithm on Logistic Regression:



$$f(\omega) = \sum_{i=1}^{n} \log(1+\exp(-y_i \omega^T x_i))$$

$$\frac{df}{d\omega} = \sum_{i=1}^{n} \frac{(-y_i x_i)\exp(-y_i \omega^T x_i)}{1+\exp(-y_i \omega^T x_i)}$$

$$\omega_1 = \omega_0 - \gamma \left[\frac{df}{d\omega}\right]_{\omega_0}$$

$$\omega_1 = \omega_0 - \gamma \left[\sum_{i=1}^{n} \frac{(-y_i x_i)\exp(-y_i \omega^T x_i)}{1+\exp(-y_i \omega^T x_i)}\right]$$

$$\omega_2 = \omega_1 - \gamma \left[\sum_{i=1}^{n} \frac{(-y_i x_i)\exp(-y_i \omega^T x_i)}{1+\exp(-y_i \omega^T x_i)}\right]$$

and so on ...

$$\omega_i = \omega_{i-1} - \gamma \left[\sum_{i=1}^{n} \frac{(-y_i x_i)\exp(-y_i \omega^T x_i)}{1+\exp(-y_i \omega^T x_i)}\right]$$

Gradient Descent in Logistic Regression (Image by Author)

We'll calculate W_0, W_1, W_2, …., W_{i-1}, W_i to find W*. When (W_{i-1} — W_i) is small i.e., when W_{i-1}, W_i converge, we declare W* = W_i

The **disadvantage** of Gradient Descent:

When n(number of data points) is large, the time it takes for *k* iterations to calculate the optimum vector becomes very large.

Time Complexity: O(kn^2)

This problem is solved with Stochastic Gradient Descent and is discussed in the next section.

5. STOCHASTIC GRADIENT DESCENT(SGD)

In SGD, we do not use all the data points but a sample of it to calculate the local minimum of the function. Stochastic basically means Probabilistic. So we select points randomly from the population.

- **SGD in Logistic Regression**

$$\omega_1 = \omega_{i-1} - \gamma \left[\sum_{i=1}^{m} \frac{(-y_i x_i)\exp(-y_i \omega^T x_i)}{1+\exp(-y_i \omega^T x_i)} \right]$$

$$1 < m < n$$

Stochastic Gradient Descent in Logistic Regression (Image by Author)

Here, *m* is the sample of data selected randomly from the population, *n*

Time Complexity: O(km^2). m is significantly lesser than n. So, it takes lesser time to compute when compared to Gradient Descent.

6. CONCLUSION

In this article, we discussed Optimization algorithms like Gradient Descent and Stochastic Gradient Descent and their application in Logistic Regression. SGD is the most important optimization algorithm in Machine Learning. Mostly, it is used in Logistic Regression and Linear Regression. It is extended in Deep Learning as Adam, Adagrad.

7. REFERENCES

[1] Maxima and Minima: https://en.wikipedia.org/wiki/Maxima_and_minima

[2] Gradient Descent: https://en.wikipedia.org/wiki/Gradient_descent

Optimization
Gradient Descent
Stochastic Gradient
Learning Rate
Maxima And Minima

Written by Supriya Secherla

55 Followers · Writer for Towards Data Science

Software Engineer at IBM India. Data Science Enthusiast. https://www.linkedin.com/in/supriya-secherla-58b392107/

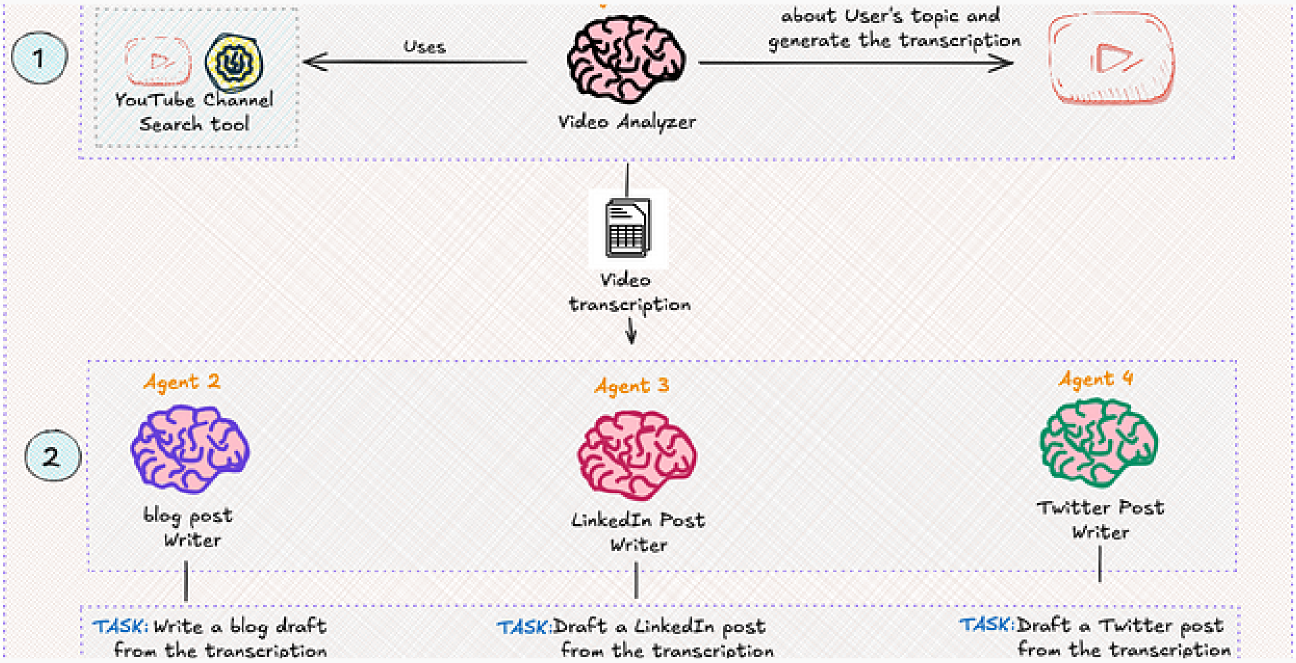
More from Supriya Secherla and Towards Data Science

Supriya Secherla in Towards Data Science

Different Imputation Methods to Handle Missing Data

Imputation methods to handle missing values in the dataset.

Jun 12, 2021 👏 108



1
YouTube Channel Search tool
Uses
Video Analyzer
about User's topic and generate the transcription
Video transcription
2
Agent 2
blog post Writer
Agent 3
LinkedIn Post Writer
Agent 4
Twitter Post Writer
TASK: Write a blog draft from the transcription
TASK: Draft a LinkedIn post from the transcription
TASK: Draft a Twitter post from the transcription

Zoumana Keita in Towards Data Science

AI Agents — From Concepts to Practical Implementation in Python

This will change the way you think about AI and its capabilities

✦ Aug 12 1.1K 13

Ahmed Besbes in Towards Data Science

What Nobody Tells You About RAGs

A deep dive into why RAG doesn't always work as expected: an overview of the business value, the data, and the technology behind it.

✦ Aug 23 1.5K 22

Bernd Wessely in Towards Data Science

Avoid Building a Data Platform in 2024

Why articles about ‘Building a Data Platform’ are mostly misleading

✦ Aug 13 389 12

See all from Supriya Secherla

See all from Towards Data Science