Volume 6, Issue 07, July 2021 ISSN (online): 2456-0006
International Journal of Science Technology Management and Research
Available online at: www.ijstmr.com

Gradient Descent Algorithm Better Approach for Optimizations

Neha Sitole
M Tech 4th Semester
C.S.E. Department
Lord Krishna College of Technology Indore

Mr. Rahul Pawer
Assistant Professor
C.S.E. Department
Lord Krishna College of Technology Indore

Abstract: ***Optimization may be defined as the process by which an optimum is achieved. It is all about designing an optimal output for problems with the use of resources available. Optimization in machine learning is slightly different. In most of the cases, we are aware of the data, the shape and size, which also helps us know the areas we need to improve. But in machine learning we do not know how the new data may look like, this is where optimization acts perfectly. Optimization techniques are performed on the training data and then the validation data set is used to check its performance Gradient descent is one of the most popular algorithms to perform optimization and by far the most common way to optimize neural networks. It is an iterative optimization algorithm used to find the minimum value for a function. Gradient Descent Algorithm helps us to make these decisions efficiently and effectively with the use of derivatives. In this paper we provide an approach with Gradient Descent Algorithm for optimisation.***

Keywords: Gradient Descent , Optimisation, Minimum , Efficiency ,Training ,Validation

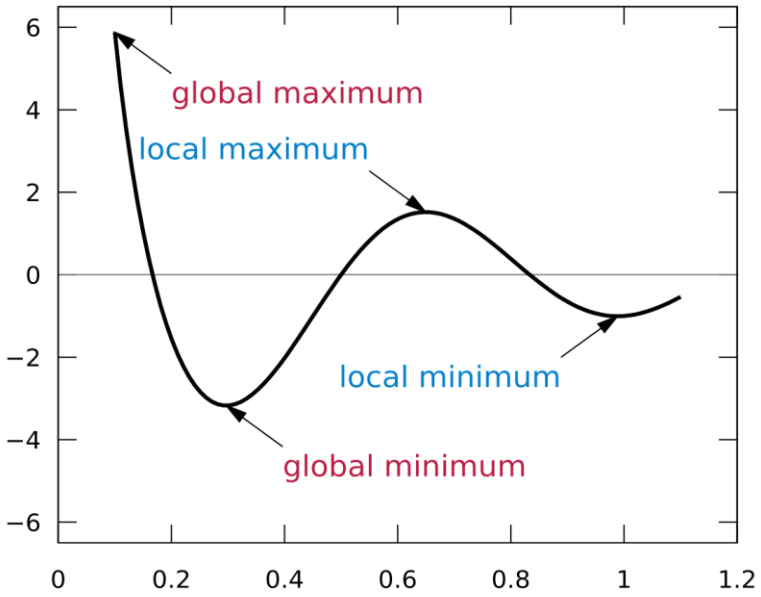
I. Introduction

Gradient Descent is an iterative process that finds the minima of a function. This is an optimization algorithm that finds the parameters or coefficients of a function where the function has a minimum value. Although this function does not always guarantee to find a global minimum and can get stuck at a local minimum

To find the local minimum of a function using gradient descent, we must take steps proportional to the negative of the gradient (move away from the gradient) of the function at the current point. If we take steps proportional to the positive of the gradient (moving towards the gradient), we will approach a local maximum of the function, and the procedure is called **Gradient Ascent.**

Error = Y(Predicted)-Y(Actual)

A machine learning model always wants low error with maximum accuracy, in order to decrease error we will intuit our algorithm that you're doing something wrong that is needed to be rectified, that would be done through Gradient Descent.



IJSTMR © *2021* | *All Rights Reserved*

Figure 1 Global minimum and local minimum in Gradient Descent

II. CHALLENGES IN EXECUTING GRADIENT DESCENT

Gradient Descent is a sound technique which works in most of the cases. But there are many cases where gradient descent does not work properly or fails to work altogether. There are three main reasons when this would happen:

Data challenges
Gradient challenges
Implementation challenges

Data Challenges

If the data is arranged in a way that it poses a non-convex optimization problem. It is very difficult to perform optimization using gradient descent. Gradient descent only works for problems which have a well-defined convex optimization problem.
Even when optimizing a convex optimization problem, there may be numerous minimal points. The lowest point is called global minimum, whereas rest of the points are called local minima. Our aim is to go to global minimum while avoiding local minima.
There is also a saddle point problem. This is a point in the data where the gradient is zero but is not an optimal point. We don't have a specific way to avoid this point and is still an active area of research.

Gradient Challenges

If the execution is not done properly while using gradient descent, it may lead to problems like vanishing gradient or exploding gradient problems. These problems occur when the gradient is too small or too large. And because of this problem the algorithms do not converge.

Implementation Challenges

Most of the neural network practitioners don't generally pay attention to implementation, but it's very important to look at the resource utilization by networks. For eg: When implementing gradient descent, it is very important to note how many resources you would require. If the memory is too small for your application, then the network would fail.
Also, its important to keep track of things like floating point considerations and hardware / software prerequisites.

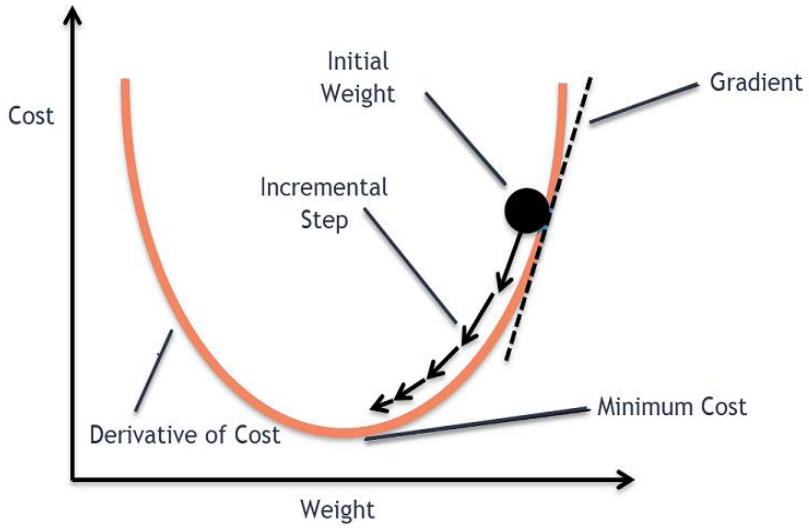


Figure 2 Learning rate in Gradient Descent

III. Literature survey

In 2013 Ilya Sutskever James Martens proposed "On the importance of initialization and momentum in deep learning". Deep and recurrent neural networks (DNNs and RNNs respectively) are powerful models that were considered to be almost impossible to train using stochastic gradient descent with momentum. They showed that when stochastic gradient descent with momentum uses a well-designed random initialization and a particular type of slowly increasing schedule for the momentum parameter, it can train both DNNs and RNNs(on datasets with long-term dependencies) to levels of performance that were previously achievable only with Hessian-Free optimization. They found that both the initialization and the momentum are crucial since poorly initialized networks cannot be trained with momentum and well-initialized networks perform markedly worse when the momentum is absent or poorly tuned. Training these models suggests that previous attempts to train deep and recurrent neural networks from random initializations have likely failed due to poor initialization schemes[1].

In 2014 Yiming Ying and Massimiliano Pontil proposed "Online gradient descent learning algorithm". They consider the least-square online gradient descent algorithm in a reproducing kernel Hilbert space (RKHS) without an explicit regularization term. They presented a novel capacity independent approach to derive error bounds and convergence results for this algorithm. The essential element in our

analysis is the interplay between the generalization error and a weighted cumulative error. They showed that, although the algorithm does not involve an explicit RKHS regularization term, choosing the step sizes appropriately can yield competitive error rates with those in the literature[2].

In 2015 Diederik P. Kingma Jimmy Lei Ba proposed "ADAM: A Method For Stochastic Optimization". They introduced Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The method is straightforward to implement, is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters. The method is also appropriate for non-stationary objectives and problems with very noisy and/or sparse gradients. The hyper-parameters have intuitive interpretations and typically require little tuning. Some connections to related algorithms, on which Adam was inspired, are discussed. They also analyze the theoretical convergence properties of the algorithm and provide regret bound on the convergence rate that is comparable to the best known results under the online convex optimization framework. Empirical results demonstrate that Adam works well in practice and compares favorably to other stochastic optimization methods[3].

In 2016 Marcin Andrychowicz, Misha Denil and Sergio Gómez Colmenarejo proposed "Learning to learn by gradient descent by gradient descent". The move from hand-designed features to learned features in machine learning has been wildly successful. In spite of this, optimization algorithms are still designed by hand. They showed how the design of an optimization algorithm can be cast as a learning problem, allowing the algorithm to learn to exploit structure in the problems of interest in an automatic way. They implemented by LSTMs, outperform generic, hand-designed competitors on the tasks for which they are trained, and also generalize well to new tasks with similar structure. They demonstrate this on a number of tasks, including simple convex problems, training neural networks, and styling images with neural art. They showed how to cast the design of optimization algorithms as a learning problem, which enables us to train optimizers that are specialized to particular classes of functions[4].

In 2017 Stephan Mandt and Matthew D. Hoffman proposed "Stochastic Gradient Descent as Approximate Bayesian Inference". Stochastic Gradient Descent with a constant learning rate (constant SGD) simulates a Markov chain with a stationary distribution. With this perspective, they derive several new results. (1) Constant SGD can be used as an approximate Bayesian posterior inference algorithm. Specifically ,how to adjust the tuning parameters of constant SGD to best match the stationary distribution to a posterior, minimizing the Kullback Leibler divergence between these two distributions(2) They demonstrate that constant SGD gives rise to a new variation EM algorithm that optimizes hyper parameters in complex probabilistic models. (3) They also showed how to tune SGD with momentum for approximate sampling. (4) They analyze stochastic-gradient MCMC algorithms. For Stochastic-Gradient Langev Dynamics and Stochastic-Gradient Fisher Scoring, we quantify the approximation errors due to finite learning rates. (5) They used stochastic process perspective to give a short proof of averaging is optimal[5].

In 2018 Loucas Pillaud-Vivien and Alessandro Rudi proposed "Statistical Optimality of Stochastic Gradient Descent on Hard Learning Problems through Multiple Passes". They considered stochastic gradient descent (SGD) for least-squares regression with potentially several passes over the data. While several passes have been widely reported to perform practically better in terms of predictive performance on unseen data, the existing theoretical analysis of SGD suggests that a single pass is statistically optimal. While this is true for low-dimensional easy problems, they showed that for hard problems, multiple passes lead to statistically optimal predictions while single pass does not; they also showed that in these hard models, the optimal number of passes over the data increases with sample size. They illustrated results on synthetic experiments with non-linear kernel methods and on a classical benchmark with a linear model[6].

In 2019 Dokkyun Yi ,Sangmin Ji and Sunyoung Bu proposed "An Enhanced Optimization Scheme Based on Gradient Descent Methods for Machine Learning". A learning process of machine learning consists of finding values of unknown weights in a cost function by minimizing the cost function based on learning data. The existing methods used to find the minimum values usually use the first derivative of the cost function. When even the local minimum (but not a global minimum) is reached, since the first derivative of the cost function becomes zero, the methods give the local minimum values, so that the desired global minimum cannot be found. They modified one of the existing schemes the adaptive momentum estimation scheme by adding a new term, so that it can prevent the new optimizer from staying at local minimum. They introduced an enhanced optimization scheme based on the popular optimization scheme, Adam, for non-convex problems induced from the machine learning process. Most existing optimizers may stay at a local minimum for non-convex problems when they meet the local minimum before meeting a global minimum. Even they have some difficulty in finding the global minimum within a complicated non-convex system[7].

In 2020 Nam D. Vo ,Minsung Hong and Jason J. Jung proposed **"**Implicit Stochastic Gradient Descent Method for Cross-Domain Recommendation System**".** The previous recommendation system applied the matrix factorization collaborative filtering (MFCF) technique to only single domains. Due to data sparsity, this approach has a limitation in overcoming the cold-start problem. They focused on discovering latent features from domains to understand the relationships between domains (called domain coherence).This approach uses potential knowledge of the source domain to improve the quality of the target domain recommendation. They consider applying MFCF to multiple domains. Mainly, by adopting the implicit stochastic gradient descent algorithm to optimize the objective function for prediction, multiple matrices from different domains are consolidated inside the cross-domain recommendation system (CDRS). Additionally, we design a conceptual framework for CDRS, which applies to different industrial scenarios for recommenders across domains[8].

IV. Problem Statement

- If the data is arranged in a way that it poses a non-convex optimization problem. It is very difficult to perform optimization using gradient descent. Gradient descent only works for problems which have a well-defined convex optimization problem.
- Even when optimizing a convex optimization problem, there may be numerous minimal points. The lowest point is called global minimum, whereas rest of the points are called local minima. Our aim is to go to global minimum while avoiding local minima.
- There is also a saddle point problem. This is a point in the data where the gradient is zero but is not an optimal point. We don't have a specific way to avoid this point and is still an active area of research.

Proposed Approach

Steps used in the proposed approach are

Step 1: Initialize the weights (a & b) with random values and calculate Error (SSE)

Step 2: Calculate the gradient i.e. change in SSE when the weights (a & b) are changed by a very small value from their original randomly initialized value. This helps us move the values of a & b in the direction in which SSE is minimized.

Step 3: Adjust the weights with the gradients to reach the optimal values where SSE is minimized

Step 4: Use the new weights for prediction and to calculate the new SSE

Step 5: Repeat steps 2 and 3 till further adjustments to weights doesn't significantly reduce the Error

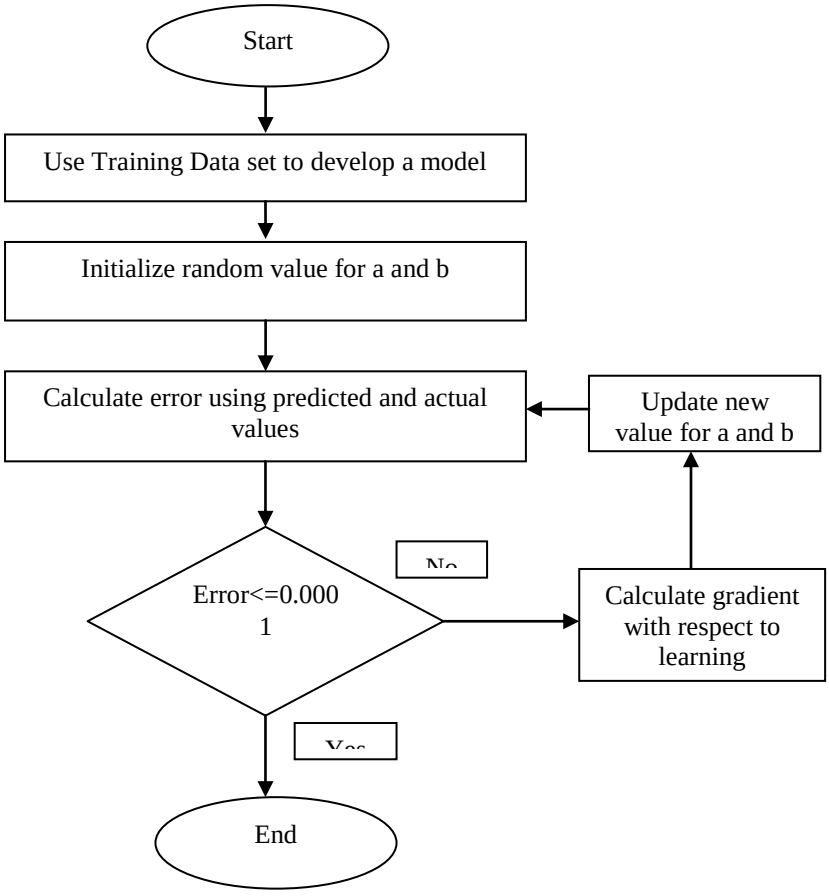


Figure 2 Architecture of proposed approach

IMPLEMENTATION

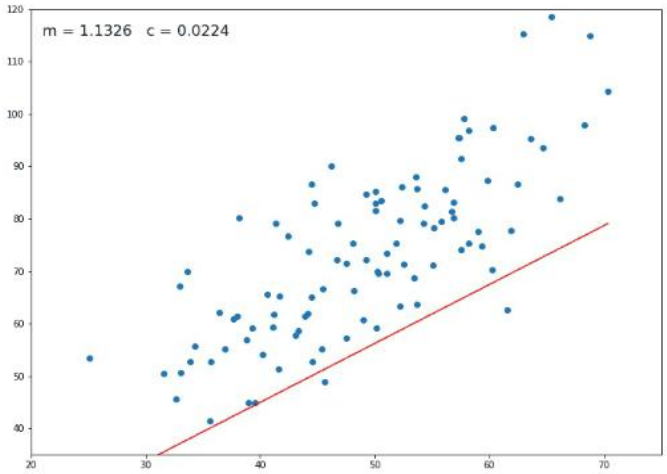


Figure 3 Implementation of proposed work

We evaluate the performance of proposed approach .We implemented the proposed approach with 1000 records for house size and house price. In the proposed work for a new house, given its size (X), what will its price (Y) be. We used VB dot net as front end to design user interface. We used SQL server 2010 R2 to store data set. We calculate the value of correction coefficient with and without outlier

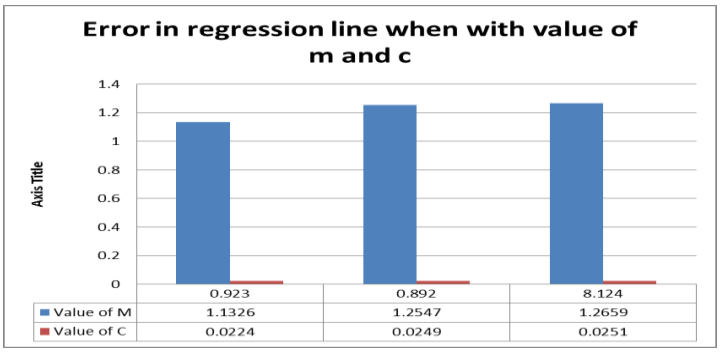


Figure 4 Reducing error by optimization

Conclusion

Optimization may be defined as the process by which an optimum is achieved. It is all about designing an optimal output for problems with the use of resources available. Optimization in machine learning is slightly different. In most of the cases, we are aware of the data, the shape and size, which also helps us know the areas we need to improve. But in machine learning we do not know how the new data may look like, this is where optimization acts perfectly. Optimization techniques are performed on the training data and then the validation data set is used to check its performance. Gradient descent is one of the most popular algorithms to perform optimization and by far the most common way to optimize neural networks. It is an iterative optimization algorithm used to find the minimum value for a function. Gradient Descent Algorithm helps us to make these decisions efficiently and effectively with the use of derivatives.

.

REFERENCE

1. Ilya Sutskever James Martens "On the importance of initialization and momentum in deep learning" Proceedings of the 30 th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).
2. Yiming Ying and Massimiliano Pontil Online gradient descent learning algorithm Department of Computer Science, University College London Gower Street, London, WC1E 6BT, England, UK fying, m.pontilg@cs.ucl.ac.uk
3. Diederik P. Kingma and Jimmy Lei Ba_ ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION arXiv: 1412.6980v9 [cs.LG] 30 Jan 2017
4. Marcin Andrychowicz, Misha Denil, Sergio Gómez Colmenarejo1 Learning to learn by gradient descent by gradient descent 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.
5. Stephan Mandt and Matthew D. Hoffman Stochastic Gradient Descent as Approximate Bayesian Inference Journal of Machine Learning Research 18 (2017) 1-35 Submitted 4/17; Revised 10/17; Published 12/17
6. Loucas Pillaud-Vivien Alessandro Rudi Statistical Optimality of Stochastic Gradient Descent on Hard Learning Problems through Multiple Passes 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

7. Dokkyun Yi 1, Sangmin Ji 2 and Sunyoung Bu An Enhanced Optimization Scheme Based on Gradient Descent Methods for Machine Learning Hongik University, Sejong 30016, Korea Received: 8 June 2019; Accepted: 17 July 2019; Published: 20 July 2019.
8. Nam D. Vo 1 , Minsung Hong 2 and Jason J. Jung 1,* Implicit Stochastic Gradient Descent Method forCross-Domain Recommendation System Big Data Research Group,Western Norway Research Institute, Box 163, NO-6851 Sogndal, Norway; 21 March 2020; Accepted: 26 April 2020; Published: 29 April 2020

***IJSTMR © 2021** | All Rights Reserved*