Volume 6, Issue 03, March 2021 ISSN (online): 2456-0006
International Journal of Science Technology Management and Research
Available online at: www.ijstmr.com

Improving Regression Analysis with Gradient Descent Algorithm for Machine Learning

Vinita Patidar
P.G. Research Scholar
C.S.E. Department
JIT Borawan Khargone

Mr. Sachine Mahajan
Assistant Professor
C.S.E. Department
JIT Borawan Khargone

Abstract: ***Accuracy is the word with which we are most concerned, while we are dealing with problems related to machine learning and artificial intelligence. Any rate of errors cannot be tolerated while dealing with real-world problems and neither should they be compromised. We need optimization algorithms to evaluate model and judge whether the model is performing according to our needs or not. The evaluation can be made easy by calculating the cost function. It is basically a mapping function that tells us about the difference between the desired output and what our model is computing. We can accordingly correct the model and avoid any kind of undesired activities. 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 proposed a Gradient Descent based optimisation approach to find best fitted regression line for the given data set.***

Keywords: Accuracy, Optimisation, Regression, Error, Cost

I. INTRODUCTION

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. A **derivative** is a term that comes from calculus and is calculated as the slope of the graph at a particular point. The slope is described by drawing a tangent line to the graph at the point. So, if we are able to compute this tangent line, we might be able to compute the desired direction to reach the minima. We will talk about this in more detail in the later part of the article. Let us now put all these learning into a mathematical formula. In the equation, y = m* x+b, 'm' and 'b' are its parameters. During the training process, there will be a small change in their values. Let that small change be denoted by δ**.** The value of parameters will be updated as **m=m-δm** and **b=b-δb** respectively. Our aim here is to find those values of m and b in y = m * x+b, for which the error is minimum i.e values which minimize the cost function.

Gradient descent is an optimization algorithm which is mainly used to find the minimum of a function. In machine learning, gradient descent is used to update parameters in a model. Parameters can vary according to the algorithms, such as coefficients in Linear Regression and weights in Neural Networks. Let us relate gradient descent with a real-life analogy for better understanding. Think of a valley you would like to descend when you are blind-folded. Any sane human will take a step and look for the slope of the valley, whether it goes up or down. Once you are sure of the downward slope you will follow that and repeat the step again and again until you have descended completely (or reached the minima).

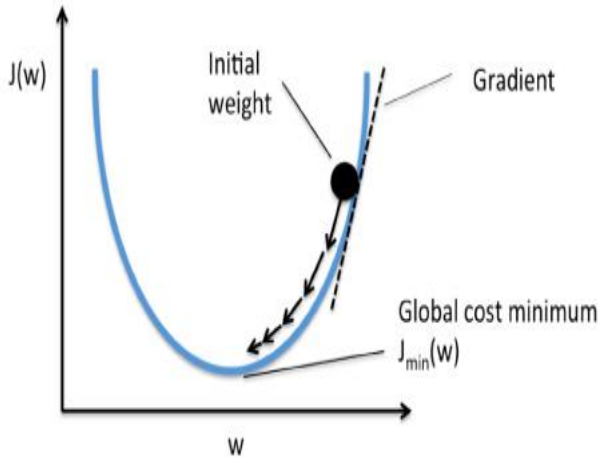

Figure 1 Working of gradient descent

II. TYPES OF GRADIENT DESCENT ALGORITHMS

There are three types of Gradient Descent Algorithms:
1. Batch Gradient Descent
2. Stochastic Gradient Descent
3. Mini-Batch Gradient Descent

1. Batch Gradient Descent
It is the first basic type of gradient descent in which we use the complete dataset available to compute the gradient of cost function .As we need to calculate the gradient on the whole dataset to perform just one update, batch gradient descent can be very slow and is intractable for datasets that don't fit in memory.

- In the batch gradient descent, to calculate the gradient of the cost function, we need to sum all training examples for each steps
- If we have 3 million samples (m training examples) then the gradient descent algorithm should sum 3 million samples for every epoch. To move a single step, we have to calculate each with 3 million times!
- Batch Gradient Descent is not good fit for large datasets

2. Stochastic Gradient Descent (SGD)
Batch Gradient Descent turns out to be a slower algorithm. So, for faster computation, we prefer to use stochastic gradient descent. The first step of algorithm is to randomize the whole training set. Then, for updating of every parameter we use only one training example in every iteration to compute the gradient of cost function. As it uses one training example in every iteration this algorithm is faster for larger data set. In SGD, one might not achieve accuracy, but the computations of results are faster.
After initializing the parameter with arbitrary values we calculate gradient of cost function using following relation:

- In stochastic Gradient Descent, we use one example or one training sample at each iteration instead of using whole dataset to sum all for every steps
- SGD is widely used for larger dataset trainings and computationally faster and can be trained in parallel
- Need to randomly shuffle the training examples before calculating it
- Python code implementation for SGD in below

3. Mini-Batch Gradient Descent
Mini batch algorithm is the most favorable and widely used algorithm that makes precise and faster results using a batch of ' m ' training examples. In mini batch algorithm rather than using the complete data set, in every iteration we use a set of 'm' training examples called *batch* to compute the gradient of the cost function. Common mini-batch sizes range between 50 and 256, but can vary for different applications.

- Reduces the variance of the parameter updates, which can lead to more stable convergence.
- Can make use of highly optimized matrix, which makes computing of gradient very efficient.

II. 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

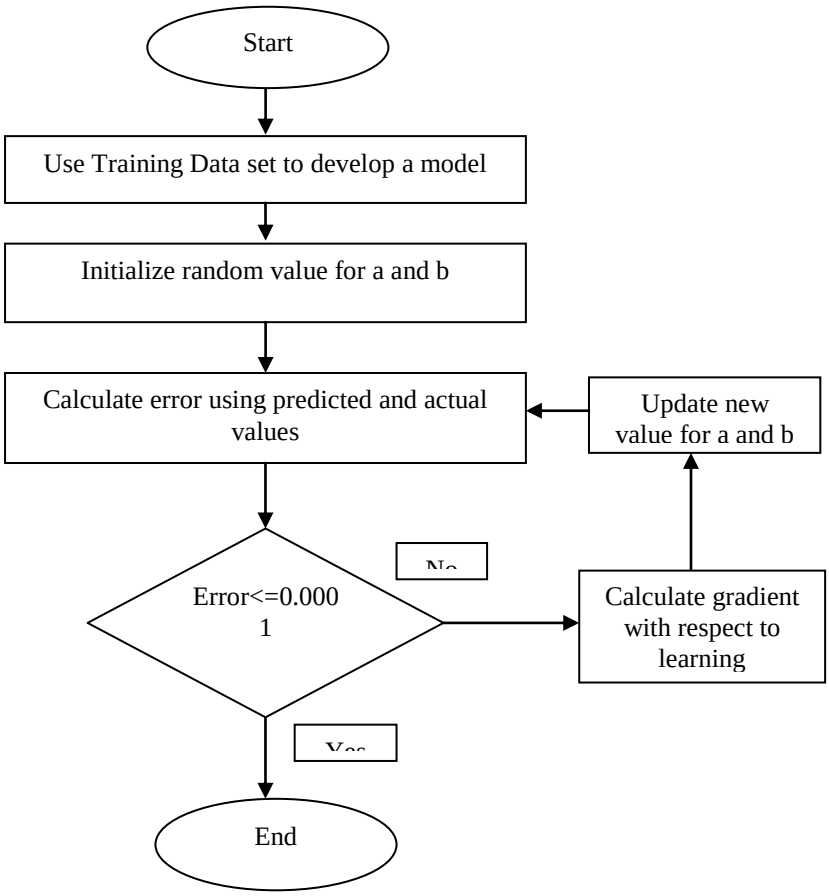

Figure 2 Architecture of proposed approach

V. IMPLEMENTATION

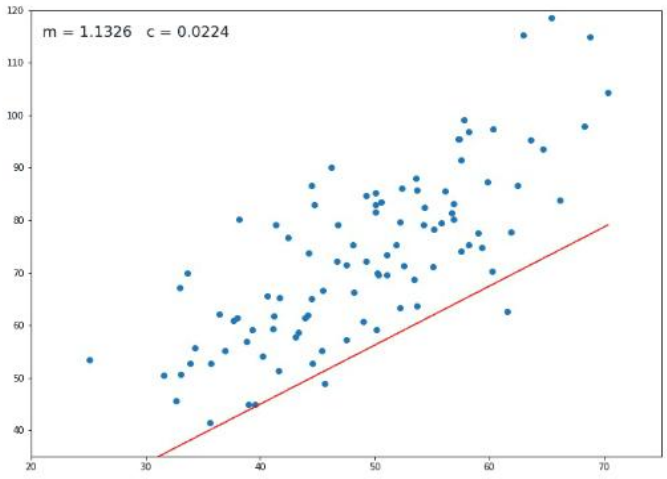

Figure 3 Implementation of proposed work

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

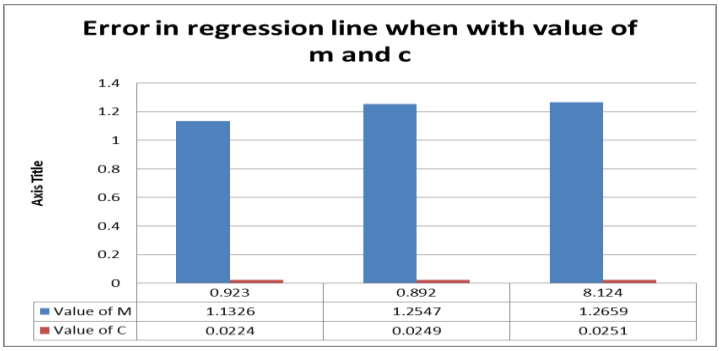

Figure 4 Reducing error by optimization

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