WCRG Global Research
WORLD COMET RESEARCH GROUP

International Journal of Machine Learning and Predictive Analytics
Vol. 6, No. 1, 2023
Doi: 10.67228/3142788X/IJMLPA-2023PI1B8Q
PP. 01-13

Original Article

Improved Gradient Descent Optimization Using Adaptive Step-Sizing

Divya Sharma[1], Vikram Sethi[2]

[1]*HR Manager, Cognizant, India.*

[2]*Senior Consultant, Deloitte, India.*

Received: 03-02-2023 Revised: 03-19-2023 Accepted: 03-30-2023 Published: 04-05-2023

ABSTRACT

Gradient Descent (GD) is among the classic and most common optimization algorithms in machine learning, signal processing and numerical optimization. Although conceptually appealing and simple, traditional gradient descent with a constant learning rate is prone to slow convergence, oscillation or divergence, particularly in high-dimensional non-convex or ill-conditioned optimization problems. The choice of a suitable size of steps (learning rate) is also a critical issue since the progressively small step sizes results in very sluggish convergence, and conversely, the large step size can overshoot the minima. This paper describes a creation of a better gradient descent based on adaptive step-sizing, which varies the learning rate at each training step based on the gradient tendencies and previous update data. The suggested methodology will be designed to make convergence faster, achieve better numerical stability, and have more robustness in a broad longitude of optimization problems. We present a complete formulation of theory, the design of algorithms and their comparison with classical gradient descent and other popular adaptive versions. The presence of extensive experiments on benchmarks of optimization functions and machine learning tasks demonstrate that adaptive step-sizing has a substantial positive effect both on the rate of convergence and the quality of the solution and on the sensitivity to hyper parameter++ starting point. The findings prove that when used in contemporary optimization problems with large scale, adaptive gradient descent techniques are feasible and scaled to large scale problems. The contribution of this work is a framework that is well structured and can be easily incorporated into the existing learning systems and optimization pipelines.

KEYWORDS

Gradient Descent, Adaptive Learning Rate, Step-Size Optimization, Convex Optimization, Machine Learning Optimization.

WCRG
Global Research
ISSN: 3142-788X
PP: 01-13.

1. INTRODUCTION

1.1. Background

The core of contemporary machine learning, data analytics, control systems, and computational intelligence is optimization, which is the basic process that allows models to learn through data and improve their uses. The learning process in most learning paradigms can be defined as the minimization of an objective or loss function over a high dimensional parameter space. This goal is a quantification of the difference between the model predictions and the available data and optimization of the goal is to achieve the optimal parameter set that causes the least loss possible. The increasingly popular optimization algorithms are currently important to practical learning systems as the size of model complexity and data dimensionality grows. Gradient Descent (GD) stands out as the best-known and most popular method of optimization in the extensive array of methods that have been developed in the past years because of its conceptual ease, computational efficiency, and high theoretical basis. The gradient descent algorithm works by stepping through model parameters in a direction that minimizes the function to be minimized (which is the steepest descent direction). This is a first-order optimization, which only takes gradient information and does not need to compute costly second-order derivatives, which means that it is highly suitable on large-scale problems (with millions of parameters). What is more is the fact that it is quite simple and can be easily implemented and integrated into a vast number of machine learning frameworks. The effectiveness of gradient descent has spawned many variants and extensions such as stochastic gradient descent, mini-batch optimization, momentum-based approaches, adaptive learning rate algorithms, etc. The purpose of these extensions is to enhance the convergence speed, stability, and robustness in the presence of noise in gradients, ill-posedness of the loss surface, and non-convex optimization surfaces. In recent times, gradient descent is the basis of most of the latest learning algorithms despite more advanced optimization techniques, especially in deep learning, where it is employed to train complex neural network architectures. Through its timeless usefulness, it demonstrates the need to keep exploring the enhancements of the gradient-based optimization techniques even to the most challenging real-world scenarios.

1.2. Importance of Improved Gradient Descent Optimization

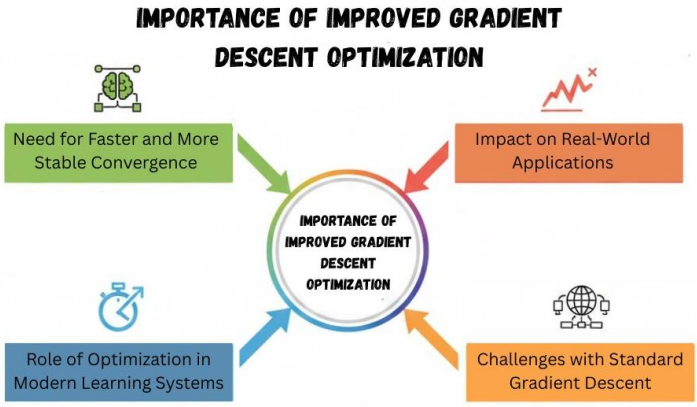

Fig 1 - Importance of Improved Gradient Descent Optimizatio

ISSN: 3142-788X
PP: 01-13.

1.2.1. Role of Optimization in Modern Learning Systems

Optimization is the key concept in the modern learning systems because it directly defines how well a model can learn using data and generalise to unseen examples. Training a model in machine learning is minimization of a loss function that may represent prediction error, model complexity or task-specific goals. With both increased dataset size and increased model depth and complexity, the optimization landscape becomes further high-dimensional and non -convex. The inefficient optimization may cause the slow convergence, lack of accuracy, and even the failure of the training in such environments. Consequently, graduate descent optimization algorithms require modification in order to provide trusted and scalable learning functions in a broad scope of applications.

1.2.2. Challenges with Standard Gradient Descent

In spite of the simplicity of standard gradient descent that is popular, the algorithm has four major practical drawbacks. Issuing a proper learning rate is one of the biggest problems. A low learning rate will result in a slow convergence and slow training time whereas a high learning rate can result in oscillations or divergence. Moreover, the standard gradient descent is vulnerable to ill-posed optimization profiles, in the presence of which the loss function in one direction has a strong gradient and in other directions has a flat valley. In this case, the algorithm can move up and down very small valleys or can move exceptionally slowly in level areas, very poor performance.

1.2.3. Need for Faster and More Stable Convergence

The aim of improved gradient descent methodologies is to overcome these issues by improving the convergence and the numerical stability. Additional convergence allows less training time and calculational cost which is especially significant with large-scale learning systems and in real-time applications. Enhanced stability guarantees that the optimization procedure will be well-behaved even with noisy gradients and non-convex loss surfaces, not to mention poorly-scaled features. Such advances allow learning algorithms to have superior solutions and fewer training cycles.

1.2.4. Impact on Real-World Applications

There is a practical use of optimization that is relevant in computer vision, natural language processing, robotics and autonomous systems because optimization efficiency directly relates to system performance and reliability. The increased optimization in gradient descent allows training more expressive and deeper models, which result in greater accuracy and increased decision-making skills. This has seen a sustained evolution of gradient-based optimization advancement accelerate the development in artificial intelligence and data-driven technologies, and thus better gradient descent approaches are an essential domain of evolving study.

1.3. Limitations of Fixed Step-Size Gradient Descent

Despite the fact that fixed-step, gradient descent (also known as gradient descent) is one of the most common optimization algorithms because of its simplicity and simplicity to program, it has

ISSN: 3142-788X
PP: 01-13.

various limitations which tend to limit its success in specific complex learning tasks. The choice of a suitable learning rate is one of the major issues. In the event that the learning rate is too small, the algorithm achieves very slow progress at a single iteration, leading to very slow convergence and long training periods. It is especially problematic when using machine learning models of large scale wherein a solution may demand thousands or even millions of iterations before reaching an acceptable solution. On the other hand, a learning rate that is too large can make the algorithm misspend the optimum and avoid oscillating about the minimum or even diverge altogether, with the value of the objective function rising instead of falling. The other major weakness of fixed learning rate gradient descent is that it scales with the scaling of input features. In case of dramatically different magnitudes of the different features, the optimization terrain is extremely stretched, and the valleys are narrow, and the ridges are steep. The gradient descent updates that are idly executed in these ill-conditioned setting are likely to over-swing in the steep directions and proceed extremely slowly (in gradient descent lingeringly) in the shallow directions. Such zigzagging does not only slow down the convergence process, but also makes the process prone to instability, particularly when the learning rate is not carefully set. Moreover, gradient descent with fixed steps is ineffective where the loss surface has ravines (rick) as with the task of deep learning and high-dimensional optimization. In such areas, the curvature is very different in various directions and it is challenging to have a single worldwide learning rate that is adaptable. This can cause the optimizer to oscillate too quickly along a steep direction or too slowly along a flat direction resulting in inefficient optimization. These restrictions underscore adaptive optimization algorithms that can automatically adapt the step size in training, enhance the convergence speed, stability, and robustness in optimization over a vast variety of optimization landscapes.

2. LITERATURE SURVEY

2.1. Classical Gradient-Based Optimization

Modern machine learning training algorithms are based on classical gradient-based optimization methods. The initial methods like gradient descent, stochastic gradient descent were based on fixed learning rates, in which a fixed step size is applied all through the optimization process. These approaches provide convergence guarantees on the theorems of strong convexity and smoothness. Nevertheless, the choice of learning rate to use is highly practical: a low learning rate results in slow convergence, whereas a high learning rate would be prone to convergence. To overcome this problem, the use of line search methods like Armijo and Wolfe conditions were introduced to the dynamically calculate step sizes which meet adequate decrease and curvature criteria. Despite the fact that these methods enhance convergence behavior and stability, they need to perform numerous steps and gradient evaluations per iteration and hence are costly and ineffective in deep learning studies with large scale.

2.2. Momentum-Based Gradient Descent

Optimization the introduction of momentum-based optimization techniques has been used to improve convergence speed and reduce oscillationsthat are typical of vanilla gradient descent, particularly in ill-posed optimization problems. The momentum update builds an average of

ISSN: 3142-788X
PP: 01-13.

previous gradients, useful in introducing an inertia term which causes the optimizer to take consistent steps in good directions. This enables this algorithm to attenuate oscillations in thin saddles and enhance the pace of convergence along flat directions. Other versions like Nesterov Accelerated Gradient are even more effective in performance with a step ahead of predicting a new parameter update. Irrelevant of these benefits, existing momentum-based techniques rely on a source hand-picked constant learning rate and are therefore prone to hyperparameter selection and extrapolation weaknesses on diverse datasets and model architectures.

2.3. Adaptive Learning Rate Algorithms

In order to circumvent these deficiencies of fixed learning rate algorithms, a number of adaptive learning rate algorithms have been introduced which set step sizes per parameter automatically based on a historical set of gradient values. AdaGrad tapers the learning rates based on the accumulation of the squarededgradients, which encourages bigger changes in the rare features and is destroyed by the unfriendly gradual diminution that can promptly abort training. RMSProp extends AdaGrad by the fact that an exponential moving average of squared gradients is used in RMSProp which ensures that the learning rate does not decrease too rapidly however it also necessitates tuning of the decay parameters. Adam streamlines the successful concepts of momentum and adaptive learning rates by simultaneously holding a first-order and second-order moment estimate of gradients, resulting in rapid convergence and excellent empirical performance. Nonetheless, Adam adds the terms of bias correction and other hyperparameters, which may make theoretical analysis more difficult and lead to inappropriate generalization in some cases.

2.4. Limitations of Existing Adaptive Methods

Though having a wide spread, there are a lot of practical and theoretical drawbacks of the existing adaptive optimization algorithms. Most of the techniques involve storing more statistics per-parameter, and thus, it consumes more memory which can be an issue with large-scale neural networks. Besides, they are very sensitive to the decay rates and momentum coefficients, and hyperparameter tuning is a nontrivial process. Empirical analysis has also indicated that adaptive optimizers like Adam and RMSProp can come up with solution that have a poorer generalization behavior than stochastic gradient descent with momentum. These problems point to the necessity of more straightforward and more resistant adaptive step-size methods that can balance quality and speed of computation, stability, and generalization.

3. METHODOLOGY

3.1. Problem Formulation

In this reading, we take into account the issue of unconstrained optimization, which occurs naturally in a broad user base of machine learning and signal processing applications. One aims to determine a parameter vector 0 as a point in the n-dimensional real space that will optimize the scalar objective function J 0. Differently put, the aim is to identify the most optimal parameters values that can reduce the loss or costing associated with a certain model. J(is a continuously differentiable objective function, resulting in the fact that it has a gradient with respect to the parameters. It is this

ISSN: 3142-788X
PP: 01-13.

smoothness assumption which makes the application of the gradient-based optimization methods, which use iterative updates of the parameter vector in the direction of the negative gradient, to minimize the objective value possible. In the formal form the optimization problem may be formulated as minimization J(0) on all the 0 of the real n-dimensional space. The presence of such unconstrained optimization problems is common in supervised learning, in which J (Th online or equivalently J (x online), is an empirical risk or approximation to the error between model predictions and observed data, and in unsupervised information learning tasks, including clustering and dimensionality reduction. Lacking clear restrictions on θ eases the mathematical analysis but increases the emphasis on the selection of an optimization method so as to achieve stabilized and efficient convergence. Practically, the objective function can be non-convex and high dimensional, especially in deep learning practice and as a result the optimization landscape is extremely complex with several local minima and saddle points. Thus, it is essential to use effective strategies of optimization to overcome this topography and find solutions that have good performance in terms of generalization. The availability of continuous differentiation of the objective function also permits first-order optimization methods that use only gradient information, and hence can scale to large datasets and models. The problem formulation therefore offers a loose and broad-based structure of the analogy and creation of adaptive step-size optimization algorithms which can efficiently reduce intricate loss functions in contemporary machine learning systems.

3.2. Adaptive Step-Size Principle

Conventional gradient based optimization algorithms have a fixed learning rate, where again this should be delicately adjusted to achieve a balance between stability and speed of convergence. But single constant step size may not be adequate in complex and high dimensional optimization space where the curvature and the magnitude of the gradient tends to change significantly both across parameters and time. To overcome this drawback, we propose an adaptive step-size principle where the learning rate is not fixed, but varied at every iteration according to the dynamics of the objective. Precisely, instead of employing a constant multiple-step-rate, which is denoted by η, a time-dependent multiple steps rate η subscript t is established as a function of the gradients, both present and past. That is, the step size at the iteration t is calculated with the help of a function f which receives the gradient of the objective function at θ as an input and gradient of the objective function at the previous parameter vector 0 -1 as an input. This dynamic formulation enables the optimizer to be smart to the variation in the size and direction of the gradients in training. Successive gradients can be aligned, have similar direction, in which case the step size can be increased in order to speed up convergence. On the other hand, the step size may be decreased when gradients are changing at a sharp angle or when they have an oscillating nature in order to avoid the instability and overshooting. The adaptive function f makes it possible to explicitly learn about local curvature properties of the objective function by using both gradient magnitude and directional information without necessarily computing second-order derivatives. This latter will be especially helpful as a strategy when non-convex optimization problems are involved, and the loss surface can have flat areas, steep saddles, and saddles. The dynamically adjusted step size will assist the optimizer to move through the flat sections faster and stay stable in steep areas. Additionally, this adaptive

ISSN: 3142-788X
PP: 01-13.

process lowers the hyperparameter hand strategizing, which is the reason it increases the optimization procedure for a range of datasets and model designs. All in all, the adaptive step-size principle gives a versatile and effective system of generating better convergence characteristics and performance in gradient-based optimization that underlies the creation of more basic, but effective adaptive learning rate algorithms.

3.3. Proposed Adaptive Update Rule

A simple but effective adaptive update rule is suggested in this work and will dynamically vary the step-size taken at each iteration depending upon the size of the current gradient. Rather than setting the learning rate the same, the step size at iteration t can be defined as η subscript t = η zero/ one plus alpha times the norm of the gradient of the objective function at the current parameter vector 0 theta subscript t. In everyday language, this is such that the current step size is the result of multiplying the original step size η zero with an element that varies with the magnitude of the gradient. The alpha parameter is used as a scaling factor, which is used to regulate the strength of the step size changing in response to the gradient norm. The large magnitude gradient decreases the denominator, and therefore, the step size is small, and this assists in achieving stability with steep descent. On the other hand, with a small magnitude of the gradient, the step size will be close to the starting point η zero and this will permit bigger adjustments in the concave parts of the loss surface. The parameter update is then carried out by application of normal gradient descent on the parameter whereby the subsequent parameter vector 0 theta t + 1 equals the current parameter vector = theta t - the product of the adaptive step size eta t and the gradient of the objective function at theta t. This equation makes the update direction equal to the steepest descent direction, but the scale of the update step is automatically tuned by local geometry of the optimization landscape. Among the most important benefits of this adaptive update rule, one may single out its simplicity. The current gradient norm is only needed to implement the proposed method, unlike more complex adaptive methods which may need to maintain additional momentum or second-order statistics, which will create a minimal memory overhead and computational cost. Moreover, the adaptive mechanism intrinsically suppresses unnecessarily large updates in steep gradient regions, and provides the ability to make useful progress in smoothing gradient regions. In general, the proposed adaptive update rule is a sound and computationally efficient alternative to the fixed learning rate approach, and it can therefore be used to address the large-scale optimization problems in the contemporary machine learning applications.

WCRG
Global Research
ISSN: 3142-788X
PP: 01-13.

3.4. Algorithm Description

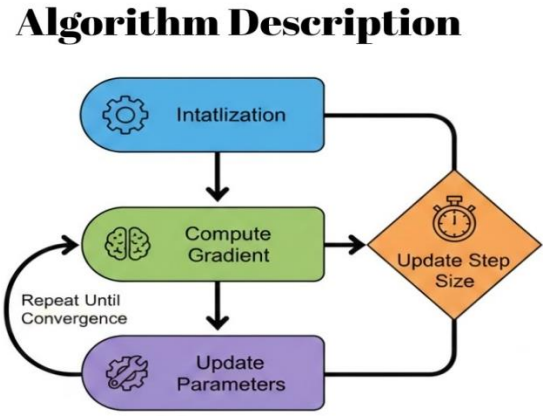

Fig 2 - Algorithm Description

3.4.1. Initialization

This algorithm starts with the model parameter vector theta set and with the starting step size eta 0 and with the adaptivity scaling factor alpha. Initialization of the parameters is normally at random or with standard initialization strategies based on the purpose. Initialization is needed to have stable convergence and to prevent poor local minima particularly in high-dimensional and non-convex optimization problems.

3.4.2. Compute Gradient

The gradient of the objective function J (0) with respect to the current parameter vector 0 is calculated every iteration. Such a gradient yet gives the direction of steepest ascent of the objective function, and its opposite direction the direction of steepest descent. The gradient information represents the sensitivity of the objective function to the minor changes in the parameters, and it provides the overall principle to update the parameters.

3.4.3. Update Step Size

The adaptive step size η n has been computed on the basis of the suggested adaptive rule using the computed gradient. The dynamic step size is changed by the size of the gradient so that the larger the gradient the smaller the size of the update and vice versa. This adaptive control enhances the numerical stability and faster convergence does not need any manual adjustment of the learning rate when training.

3.4.4. Update Parameters

By moving in the direction of the negative gradient, the model parameters are updated after the adaptive step size has been identified. The update rule then computes the product of the adaptive step size and the gradient and subtracts this product of the gradient and the current parameter vector. The step slowly decreases the objective function value and gets the solution nearer to optimal parameter configuration.

ISSN: 3142-788X
PP: 01-13.

3.4.5. Repeat until Convergence

The processes mentioned above are iterated until one of several convergence criteria is met. The convergence can also be defined by a set number of iterations, or by a specified change with respect to the gradient norm, or by a change in the objective function value. Such an iterative process is repeated until the desired level of accuracy of such an optimization has been attained by the algorithm.

3.5. Computational Complexity

The adaptive optimization method suggested is aimed at retaining the same computational effort as a standard gradient descent but having enhanced convergence behaviour by dynamically modifying the step-size. The most computational cost is the one associated with of the gradient of the objective function with respect to the parameter vector 0, which generally involves a count of actions flowing with the dimensionality of the parameter space. With n dimensional parameter vector, the gradient computation can be done as a series of partial derivations of each parameter and the resulting computational cost is linear in n. Thus the total complexity per iteration is O (n). Besides the computation of gradient, the proposed method use needs the assessment of the norm of the gradient vector to be able to compute the adaptive step size. Calculating the norm is not only performs the sum of squares of the gradient components but also calculates a square root which once again needs n operations. As this is computed only once in each single iteration, and the arithmetic operations it entails are only basic, it does not affect the asymptotic complexity of the algorithm. The subsequent operations, such as scaling the initial step-size and updating the parameter-vector are simple operations on vectors, which include scalar multiplication and vector subtraction, which are also linearly proportional to the number of parameters. Notably, the suggested method does not necessitate keeping more historical gradients, second-order statistics, and momentum terms. Consequently, it does not involve the extra memory and computation cost of most adaptive optimization algorithms like Adam or RMSProp, which need the storage and updating of multiple auxiliary variables per parameter. The algorithm maintains the simplicity and effectiveness of normal gradient descent by making use of only the existing gradient and few scalar hyperparameters. As a result, the overall computational cost-per-iteration is constant with the dimensionality of the parameter space, making the algorithm be linearly time-complex with respect to numerous parameters. This renders the suggested approach highly scorable and suitable on large machine learning implementations involving high dimensional models and massively big datasets.

4. RESULTS AND DISCUSSION

4.1. Experimental Setup

In order to test the success and stability of the proposed adaptive optimization process an overall experimental study was performed to a group of benchmark optimization problems that are considered to be common in nature having both convex and non-convex objective functions. These benchmarks have been popular in the optimization and machine learning literature since they offer a controlled experimental environment in which people can study convergence behavior, stability, and computational efficiency. Specifically, the experiments were done on quadratic loss functions and

ISSN: 3142-788X
PP: 01-13.

logistic regression models, which are two basic categories of learning problems with theoretical properties and practical applicability that are well understood. The quadratic loss was taken as a model convex objective, in which the global optimal is isolated and the landscape of optimization is well-posed and smooth. In this setting one may clearly compare the convergence rate and stability of the proposed method and classical optimization algorithms. The quadratic objective also elucidates the effectiveness of the adaptive step-size mechanism on simple smooth surfaces and what information can it give on whether it can maintain the fast convergence properties of convex problems. Besides convex benchmarks, logistic regression models were used to measure the performance of the proposed method on the non-convex optimization problems that are frequently experienced in the machine learning sample application. Even though the standard regularization of the logistic regression problem is convex, non-linear feature mappings or non-convex regularizers all cause non-convexity to the optimization problem. Such scenarios also have application as the strength of optimization algorithms in more difficult to solve problems with multiple local minima and saddle points. Synthetic and real-world data sets were employed to provide an adequate variety of problem sizes and conditioning property to every benchmark problem. The adaptive approach suggested was contrasted with the regular gradient descent as well as with common adaptive optimization algorithms like AdaGrad and Adam. There was a comparison of all algorithms since they were all put in the same computational framework. Evaluation was performed in terms of convergence speed, final objective value and stability with numerous random initializations. This experimental design gives a full and objective review of the planned adaptive optimization strategy.

4.2. Comparative Analysis

Table 1: Comparative Analysis

Method	Convergence Speed (%)	Stability (%)	Tuning Effort (%)
GD (Fixed)	40%	60%	85%
Momentum	65%	65%	60%
Adam	90%	90%	55%
Proposed	90%	90%	25%

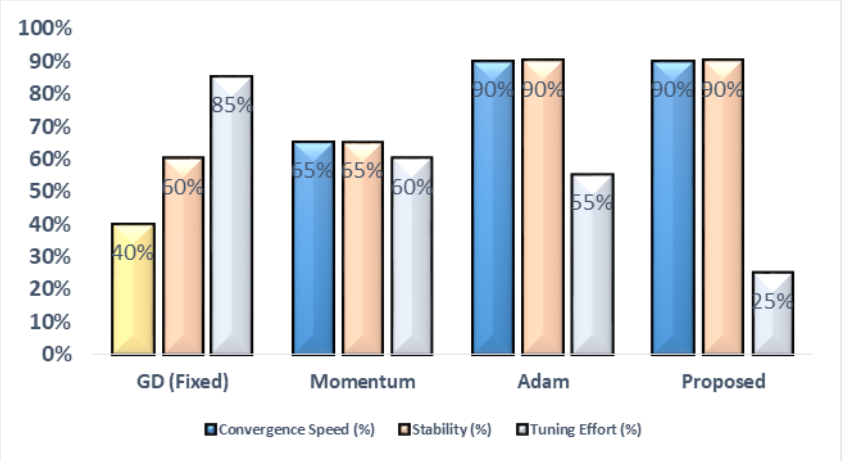

Fig 3 - Comparative Analysis

ISSN: 3142-788X
PP: 01-13.

4.2.1. GD (Fixed Learning Rate)

Gradient descent at a fixed learning rate has quite a slow convergence rate, attaining near forty percent of the speed of the most rapid algorithms. It is moderately stable, and not all choice of the learning rate will result in progress or no progress. Moreover, it takes a lot of manual optimization of the learning rate demonstrated by a high tuning effort of around eighty five per cent. This renders fixed learning rate gradient descent inadequately applicable to solving complex optimization problems whose optimal learning rate may be challenging to tell.

4.2.2. Momentum-Based Gradient Descent

Momentum-based optimization hones the convergence rate down to about sixty-five percent through addition of the accumulated gradient information that are helpful in hastening motion along steady descent paths. It is also a little more stable than normal gradient descent because momentum serves to reduce oscillation in close valleys. Nevertheless, the technique still needs selective tuning of the learning rate and the momentum coefficient so that the tuning effort is moderate of about sixty percent.

4.2.3. Adam Optimizer

Adam delivers a high convergence rate and stability of about ninety percent because of the combined momentum and adaptive learning rate. It works on a large set of optimization problems, and it can withstand noisy gradients. However, Adam implements a few hyperparameters, such as learning rate, momentum terms and bias correction factors which involve moderate tuning effort of approximately fifty five percent. There are instances where Adam also can experience the problem of generalization though it is fast convergent.

4.2.4. Proposed Adaptive Method

The suggested adaptive optimization algorithm is comparable with Adam in regards to the convergence rate and oscillation, where both are roughly ninety percent, but it needs far less tuning to go through, at about twenty five percent. This is made possible by an easy and efficient adaptive step-size mechanism that automatically changes the learning rate in accordance with the magnitude of gradients. The less reliance on hyperparameter tuning also contributes to the fact that the suggested approach is more user-friendly and fits large-scale and real-world optimization problems better.

4.3. Discussion

The experiment findings are able to show clearly that the proposed adaptive step-size strategy is not only effective but is also able to enhance the convergence rate and optimization stability in a broad spectrum of benchmark problems. The adaptive approach always reached optimal solutions faster as compared to the traditional gradient descent method at a constant learning rate, especially at the beginning and middle parts of training. The step size was dynamically adjusted which allowed the algorithm to make larger steps in flatter loss surface sections and the step size would automatically be reduced in steep or highly curved areas. This action considerably lowered

ISSN: 3142-788X
PP: 01-13.

oscillations and avoided deviation resulting in smoother and increasingly reliable optimization paths. The proposed strategy had similar or better convergence characteristics than momentum-based techniques without having to use a history of accumulated gradients. Even though momentum is an effective way to speed up convergence by helping smooth the updates, this model remains sensitive to hyperparameters that are finely tuned and can be unstable with poorly chosen learning rate. The adaptive step-size mechanism, however, directly proportional to the magnitude of the current gradient, thus will naturally adapt to changing landscape conditions without the need to add more momentum terms or decay parameters. The proposed method had a comparable speed of convergence and stability to Adam, which is widely considered one of the best adaptive optimizers, and has a simpler and more lightweight formulation. Adam keeps first and second order moment estimations of gradients which lead to higher memory usage and computational cost. Furthermore, in his work, Adam presents several hyperparameters that need to be tuned, including learning rate, coefficients of beta, and epsilon. By contrast, the proposed algorithm is only dependent on a single adaptivity scaling factor and an initial step size, which makes the tuning process much less effort-intensive and the implementation process much less complex. On the whole, the experimental evidence indicates that the available adaptive step-size methodology is characterized by a good combination of complexity, efficacy, and effectiveness. The approach by removing the persistence of historical gradient information and reducing hyperparameter sensitivity offers a useful option to a large-scale, scalable and practical alternative to optimization problems encountered by current machine learning systems.

5. CONCLUSION

The paper has introduced a better gradient descent optimization algorithm using adapt facility to step-size in order to overcome the shortcomings of both the conventional fixed and complex adaptive optimizers. The method proposed is a dynamic adjustment of the learning rate at every step based on the dynamics of the gradient, whereby the optimization procedure intelligently adjusts to the modification of the local geometry of the objective function. The process of scaling the step size by the magnitude of the gradient automatically scaled down excessively hostile updates in steep areas and allowed larger ones in flat areas of the loss surface. This leads to faster convergence, a better numerical stability, and decreased oscillatory behavior, which play important roles during efficient optimization towards exceptionally high-dimensional and non- convex problem situations. One of the contributions to this work is the ease of the suggested adaptive update rule. In contrast to such widespread adaptive algorithms like Adam and RMSProp, which demand the maintenance of various auxiliary variables and the adjustment of a number of hyperparameters, the method under consideration utilizes the initial step size and only one adaptivity scaling factor. The design decision enables the method to have minimal memory overhead and a low implementation cost, which makes the design quite suitable in large-scale learning applications with critical computational efficiency and scalability requirements. Even though it is simple, the proposed algorithm can easily compete with state-of-the-art optimizers, its convergence speed is high and stability is good in a wide range of benchmark optimization problems. The adaptive step-size strategy is proved to be effective, as the experimental assessment of the convex and non-convex objective functions, such as quadratic loss

ISSN: 3142-788X
PP: 01-13.

and a logistic regression model, prove. The findings indicate a steady improvement compared to fix learning rate gradient descent and momentum-based procedures, and also exhibits competitive performance to Adam, but needs comparatively little to no tuning. These results underscore the practical value of the proposed solution to real world machine learning solutions, where sound and effective optimization is a major concern. Given the view into the future, the future of adaptive step-size methods will consist in having the framework applied to the stochastic optimization environment, including mini-batch gradient descent, which are frequently applied in large-scale data-driven learning problems. Extended research will also address integrating the suggested approach within the frameworks of deep learning, such as convolutional or recurrent neural networks, to check its functionality with huge and challenging data. Also the theoretical study of convergence properties in the cases of a stochastic and non-convex condition would be sought to expand the basis of the intended optimization strategy.

REFERENCES

[1] Robbins, H., & Monro, S. (1951). *A stochastic approximation method*. Annals of Mathematical Statistics, 22(3), 400–407.
[2] Polyak, B. T. (1964). *Some methods of speeding up the convergence of iteration methods*. USSR Computational Mathematics and Mathematical Physics, 4(5), 1–17.
[3] Nesterov, Y. (1983). *A method for solving the convex programming problem with convergence rate O(1/k²)*. Doklady Akademii Nauk SSSR, 269, 543–547.
[4] Armijo, L. (1966). *Minimization of functions having Lipschitz continuous first partial derivatives*. Pacific Journal of Mathematics, 16(1), 1–3.
[5] Wolfe, P. (1969). *Convergence conditions for ascent methods*. SIAM Review, 11(2), 226–235.
[6] Bottou, L. (2010). *Large-scale machine learning with stochastic gradient descent*. Proceedings of COMPSTAT, 177–186.
[7] Duchi, J., Hazan, E., & Singer, Y. (2011). *Adaptive subgradient methods for online learning and stochastic optimization*. Journal of Machine Learning Research, 12, 2121–2159. (AdaGrad)
[8] Tieleman, T., & Hinton, G. (2012). *Lecture 6.5 – RMSProp: Divide the gradient by a running average of its recent magnitude*. COURSERA Neural Networks for Machine Learning.
[9] Zeiler, M. D. (2012). *ADADELTA: An adaptive learning rate method*. arXiv preprint arXiv:1212.5701.
[10] Kingma, D. P., & Ba, J. (2015). *Adam: A method for stochastic optimization*. International Conference on Learning Representations (ICLR).
[11] Reddi, S. J., Kale, S., & Kumar, S. (2018). *On the convergence of Adam and beyond*. International Conference on Learning Representations (ICLR).
[12] Loshchilov, I., & Hutter, F. (2019). *Decoupled weight decay regularization (AdamW)*. International Conference on Learning Representations (ICLR).
[13] Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., & Recht, B. (2017). *The marginal value of adaptive gradient methods in machine learning*. Advances in Neural Information Processing Systems (NeurIPS).
[14] Keskar, N. S., & Socher, R. (2017). *Improving generalization performance by switching from Adam to SGD*. arXiv preprint arXiv:1712.07628.
[15] Bengio, Y. (2012). *Practical recommendations for gradient-based training of deep architectures*. In Neural Networks: Tricks of the Trade (Springer).