Survey of Optimization Techniques for Machine Learning

[1]Kushal Shah, TY-Comp, VIIT Pune
[2]Rohit Kumar Shaw, TY-Comp, VIIT Pune
[3]Saurabh Kulkarni, TY-Comp, VIIT Pune
[4]Aniket Navghare, TY-Comp, VIIT Pune

Abstract- The fields of machine learning and mathematical programming are increasingly intertwined. Optimization of machine learning algorithms plays a significant role in training the models efficiently and applying them effectively. Machine learning algorithms can take a lot of time to train without the use of optimization techniques. Optimization algorithms help us to minimize (or maximize) an objective function which is simply a mathematical function. Several researchers have identified the flaws of previously used models to come up with more optimized or better algorithms over the period. A combined survey of various optimization techniques will help to identify a suitable algorithm for different kinds of problems. In this paper, we have compared different optimization techniques and identified limitations and advantages of different algorithms. We observed the impact of each algorithm on the efficiency of the model and its impact on training time. We have studied the impact of these algorithms on different types and sizes of data sets. Next, we summarize the applications and developments of optimization methods in machine learning fields. We have displayed the need to move to the more recently developed algorithms which have advantage over the precedent algorithms.

Key words- Machine learning, optimization method, deep neural network.

I. Introduction

In recent years, data is generated at tremendous volume and velocity. It acts as a boosting agent for technology like machine learning which, attracting a great number of researchers and practitioners. It attracts researchers across the globe and plays a significant role in many fields, such as machine translation, speech recognition, image recognition, recommendation system, etc. Optimization is one of the core components of machine learning. The essence of most machine learning algorithms is to build an optimization model and learn the parameters in the objective function from the given data. In the era of immense data, the effectiveness and efficiency of the numerical optimization algorithms play a very important role in knowing the patterns data follow. Optimization is the most essential ingredient in the recipe for machine learning algorithms. It starts with defining some kind of loss function/cost function and ends with minimizing it using several optimization techniques. The choice of an optimization algorithm can make a difference between getting good accuracy in hours or days. Optimizers come in picture when we try to make our predictions as correct as possible. They bind together the loss function and model parameters by updating the model as a result of the output of the loss function. The internal features of a model play a massive role in efficiently and effectively training model and producing accurate results. This is why we use different optimization techniques and algorithms to update and calculate appropriate and optimum values of such a model's parameters which influence our model's learning process and the output of a model. The normal stochastic gradient descent method [1], is widely used. Compared to the first-order optimization methods, high order methods converge much faster.

II. METHODOLOGY

Gradient descent is the most popular algorithm used for performing optimization in neural networks. Gradient descent is an iterative machine learning algorithm. It is also the foundation for other optimization algorithms. There are deep learning libraries which exist with the implementations of various algorithms to optimize gradient descent. The traditional batch gradient descent will calculate the gradient of the whole data set but will perform only one update. It was the first order implementation of the optimization techniques used. In comparison, (SGD) [1], descent updates the parameters for each observation which increases the number of updates. It is usually a much faster technique. It reduces the variance in the parameter updates, which can ultimately lead us to much better and stable convergence. Another recent approach is known as momentum based gradient descent [2], where a term is introduced known as momentum [3], which helps to reach the global minima faster without getting stuck at local minima. Nesterov accelerated gradient (NAG) [4], where we first make a big jump based on the previous momentum then calculate the gradient and then make a correction which results in a parameter update. Adagrad [5], lies in the realm of adaptive learning rate methods where there is a parameter introduced with the learning rate which modifies the learning rate with the iterations. AdaDelta [6], limits the window of accumulated past gradients to some fixed size w. Instead of inefficiently storing w previous squared gradients, the sum of gradients is recursively defined as a decaying mean of all past squared gradients. Adam [7], is an adaptive learning rate optimization algorithm that's been designed specifically for training deep neural networks. The algorithm leverages the power of adaptive learning rates methods to find individual learning rates for each parameter. It has advantages of [5], and a combination of (SGD) [1], with momentum [2]. Ref [7], is an adaptive learning rate method, which means, it computes individual learning rates for different parameters. Its name is derived from adaptive moment estimation as [7], uses estimations of first and second moments of a gradient to adapt the learning rate for each weight of the neural network.

III. DISCUSSION AND RESULTS

Batch gradient descent is the most basic and popular algorithm for the optimization of machine learning algorithms. It converges to the global minimum for convex error surfaces and to a local minimum for non-convex surfaces. But this algorithm is very slow and hard to control for datasets which are very large and don't fit in the memory. How big or small of an update to do is determined by the learning rate. Another thing while using standard batch gradient descent is that it computes redundant updates for large data sets. Ref [1], on the other hand frequently updates parameters which leads to high variance and causes the loss function to fluctuate to different intensities. This helps us to discover new and possibly better local minima but due to the frequent updates and fluctuations, it ultimately complicates the convergence to the exact minimum and will keep overshooting due to the frequent fluctuations. Also at a lower learning rate, the convergence pattern is the same as batch gradient descent.

The limitations of the batch gradient and (SGD) [1], are overcome by momentum-based gradient descent [2]. The momentum term increases for dimensions, whose gradients point in the same directions and reduces updates for dimensions whose gradients change directions [3]. This means it does parameter updates only for relevant examples. This reduces the unnecessary parameter updates which lead to faster and stable convergence and reduced oscillations. But a major limitation of this technique is at the minima, it has the highest momentum, hence it overshoots and misses the minima. This effect is overcome by the (NAG) [4]. It provides anticipatory updates that prevent the algorithm to go too fast and miss the minima. It makes it more responsive to changes. It can now effectively look ahead by calculating the gradient not with respect to our current parameters but with respect to the approximate future position of our parameters and then update the parameters. But it can more optimize as it uses a constant learning rate.

Ref [5], introduces varying learning rate where there is no need to manually tune the learning rate. Most implementations use a default value of 0.01 and leave it at that. Its main weakness is that its learning rate is always decreasing and decaying. This happens due to the accumulation of each squared gradients in the denominator since every added term is positive. The accumulated sum keeps growing during training. This, in turn, causes the learning rate to shrink and eventually become so small, that the model just stops learning entirely and stops acquiring new additional knowledge. Ref [6], limits the window of accumulated past gradients to some fixed size w, hence it calculates different learning rates for each parameter preventing vanishing (decaying) learning rates.

Ref [7], works well in practice and compares favorably to other adaptive learning method algorithms as it converges very fast and the learning speed of the model is rapid and efficient. Also, it rectifies every problem that is faced by other optimization techniques such as vanishing learning rate, slow convergence or high variance in the parameter updates which leads to fluctuating loss function.

IV. CONCLUSION

We have initially gone through the variants of gradient descent, in which [7], is more popular. We have studied various algorithms that are used for optimizing [1], [2], [4], [5], [6], [7], to optimize gradient descent. This paper is an overview of the frequently used optimization technique algorithms in which we investigated the behavior of these algorithms in different machine learning scenarios. We describe the theoretical basis of optimization methods as well as the research progress in recent years. Then we describe the method of the optimization methods in different machine learning optimization algorithms and the approaches to improve their performance. We have concluded that the [7], method works the best in practical applications of the machine learning problem statements. It converges to minima in a short amount of time and is more efficient than other discussed algorithms.

REFERENCES

[1] H. Robbins and S. Monro, "A stochastic approximation method," The Annals of Mathematical Statistics, pp. 400–407, 1951.

[2] N. Qian, "The momentum term in gradient descent learning algorithms," Neural networks, 12(1):145–151, 1999.

[3] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "The importance of initialization and momentum in deep learning," In Proceedings of the 30th international conference on machine learning (ICML-13), pages 1139–1147, 2013.

[4] Aleksandar Botev, Guy Lever, David Barber, "Nesterov's Accelerated Gradient and Momentum as approximations to Regularised Update Descent," Department of Computer Science University College London, 2016.

[5] John Duchi, Elad Hazan, Yoram Singer, "Adaptive Subgradient methods for online learning and Stochastic Optimization," Journal of Machine Learning Research 12, 2011.

[6] Zeiler, Matthew, "ADADELTA: An adaptive learning rate method," 1212, 2012.

[7] Kingma, Diederik & Ba, Jimmy, "Adam: A Method for Stochastic Optimization," International Conference on Learning Representations, 2014.