Advances and Applications in Mathematical Sciences
Volume 21, Issue 6, April 2022, Pages 3493-3502

A FIRST ORDER ITERATIVE METHOD FOR NUMERICAL OPTIMIZATION OF MACHINE LEARNING MODELS

URVASHI ARORA

Department of Mathematics
Rajdhani College, University of Delhi
Raja Garden, Ring Road
New Delhi-110015, India
E-mail: urvashi.arora@rajdhani.du.ac.in

Abstract

In certain practical situations dealing with machine learning, the input data and the output data are generally known, however the governing model of the problem is unknown. In such cases a hypothetical mathematical function called a hypothesis function is used to represent the unknown model of the machine learning problem and this hypothesis function is formulated on the basis of domain knowledge of input/output. This paper deals with the problem of optimizing the values of parameters occurring in the hypothesis function, so as to improve the given machine learning model. It has been shown that the problem of finding the best parameters for a given model can be solved by considering the corresponding optimization problem. For the solution of the corresponding optimization problem, the first order numerical optimization method/algorithm discussed in this paper is the gradient descent method. Both scalar and vector case for gradient descent method have been discussed.

1. Introduction

Machine learning which a part of artificial intelligence is focused on developing of computer algorithms that are capable of improving themselves by experience and by the use of sample data (training data). Machine learning finds its application for a variety of situations, be it image recognition, speech recognition, email spam, malware filtering, online fraud detection and so on.

In certain types of machine learning problems [1], [17] the input and

2020 Mathematics Subject Classification: 65K10, 68T07.
Keywords: Machine learning, Optimization, Gradient descent.
Received October 15, 2021; Accepted November 24, 2021

output data is known, but the model of the problem which governs the input/output data is unknown. This model is formulated by representing it in the form of a hypothetical mathematical function often called hypothesis function, which is constructed on the basis of domain knowledge (that is the knowledge of the specific field or discipline in context of input/output data). This hypothetical function representing the so far unknown machine learning model has some particular user defined form involving parameters/unknowns. This model is continuously trained or improved upon, till it reaches the required accuracy. The training of the model is done by the availability of more and more training data and by improving upon the value of the parameters occurring in the hypothesis function. A lot of literature is available on machine learning problems [12], [2], [18].

The improving or optimizing of the parameters occurring in the hypothesis function can be done by using mathematical optimization algorithms, which work on the principle of selection of the best choice, on the basis of some criteria, out of the available options. Since the exact form of the function governing the model is not known, but is being improved upon constantly, one has to totally rely on the numerical input/output data for optimization of parameters in the function. It is therefore appropriate to use numerical techniques/algorithms that can make use of this input/output data for optimization of parameters. Thus obtaining the optimized value of parameters occurring in the hypothetical function/machine learning model primarily rests on two factors: the sample data available and the method/algorithm used for optimization [22].

Obtaining a meaningful collection of data or a dataset [19] which can be readily used in an optimization method is a difficult and expensive procedure. Producing meaningful data requires tremendous effort, as it needs a large amount of time and manpower to convert a raw/unlabeled data in the form of, say, medical reports of patients or collection of photos of people or articles in a newspaper, in a useful form as input/output variables with respect to the context of the machine learning model being constructed. Therefore for meeting the requirement of a meaningful/labeled dataset for optimizing the hypothetical function, such dataset can either be created from raw/unlabeled data obtained from random collection of data from reliable sources like hospitals/laboratories/banks/internet etc. (at the cost of the time involved) or

by using already created repositories of data sets like UK government public data,healthdata.gov, Google trends, google finance, Amazon web services public data sets, gap minder and many more such repositories [14], [8], [24], [25].

Out of the several optimization algorithms available in literature, with each algorithm having its own advantages and disadvantages, the choice made for a particular algorithm is a trade-off between various factors such as availability of suitable dataset, computational time used etc. [4], [20], [23]. The numerical algorithm discussed in this paper for optimizing the hypothetical function is the gradient descent algorithm or the method of steepest descent, with Cauchy as its original attributor [10]. Gradient descent method is a popular, easy to implement, first order optimization algorithm that can be used for optimizing a linear as well as a nonlinear hypothesis function. Several variants of the gradient descent algorithm are available in literature. The choice of the variant used for optimization depends upon the amount of input/output data used for updating the parameters in the hypothesis function [16]. A lot of literature is available on the discussion of gradient descent and the recent trends in its regard [15], [13], [21].

The paper studies the machine learning mechanism for which the optimization method, namely the gradient descent method discussed can be suitably used. It is described how the problem of determining a suitable machine learning model gets reduced to an optimization problem, which can be solved by mathematical methods/optimization algorithms. The gradient descent method which is used for finding the optimal parameters occurring in the hypothesis function representing the machine learning model, has been described for both scalar and vector case.

2. Mechanism of Machine Learning

2.1. Steps in determination of machine learning model

In the traditional or classical programming, the input data is provided to the system along with the program/rules to be implemented to get the solution/output data. However, in certain types of machine learning problems, namely, supervised learning problems [1], [17] both input data and output data are provided to the system to figure out the governing rules or

more precisely to find out the corresponding model which describes the relation between input and output data.

Here we consider those machine learning problems in which the user has input and output data, in possession as numerical values, obtained by experimentation etc. What is required is to construct an appropriate model or precisely a mathematical function which can appropriately establish the relation between the input and output data. For this purpose, the following broad steps are used in the machine learning process.

Step 1. Collection of large data pairs.

The first step is choosing a large data set consisting of pairs of the form (x, y) where 'x' is the input vector and 'y' is the output vector. It is worth mentioning here, that 'y' denotes the ground truth or the factual truth, which is the value of output that exists in reality and has been obtained by experimentation or observation etc.

Step 2. Guessing the model or the form of hypothesis function.

This step involves guessing a model or mathematical function called hypothesis function denoted by $h(x, w)$ where 'x' is the input vector and 'w' is the vector of parameters. The model or hypothesis function is guessed or rather constructed by the user on the basis of his/her domain knowledge or experience. The significance of this hypothesis function is that it describes the possible relationship between the input data 'x' and output data 'y'. The model or hypothesis function $h(x, w)$ may involve parameters/unknowns $w_1, w_2, w_3 \ldots$ which are also called weights. As an example, a hypothesis function $h(x, w)$ may look like, say,

$$h(x, w) = w_0 + w_1 x_1 + w_2 x_1 x_2$$

where $x = (x_1, x_2)^t$ is input vector; $w = (w_1, w_2, w_3)^t$ is parameter vector

The hypothesis function may be linear or non linear or quadratic or exponential function etc. Once the hypothesis function has been decided upon by the user, it remains fixed during the learning process. The only task which remains to be done is improving upon the values of parameters $w_1, w_2, w_3 \ldots$ occurring in the hypothesis function to make the model or the hypothesis

function better. This improvising of parameters is done using appropriate algorithms and this whole process can be called as a training the machine. Thus a typical machine learning model learns only the parameters or weights.

Step 3. Learning the parameters through feedback.

In this step, an iterative method is used to optimize the parameter 'w' occurring in the user defined hypothesis function $h(x, w)$. For this an arbitrary guess 'w' is made first and then for input data 'x' and this guessed 'w' the corresponding value $\hat{y}$ is calculated from the hypothesis function $h(x, w)$ i.e., $\hat{y} = h(x, w)$. Generally, this $\hat{y}$ obtained from hypothesis is a different from the ground truth y. So an objective function $J(y, \hat{y}(w))$ representing the difference between y and $\hat{y}$ is defined. This objective function $J(w)$, sometimes called the cost function or loss function needs to be minimized. Thus using an iterative algorithm such as the gradient descent algorithm starting with an initial guess 'w', the iterative method is applied successively to improve upon the value of 'w' so that the objective function is minimized. Ideally, $y = \hat{y}$ is the condition that is required i.e., $J(w) = 0$. However achieving the optimal value of 'w' so that this condition is satisfied is practically difficult. So the iterative method is stopped, when the desired accuracy is reached and in that case the machine learning model is said to be trained.

3. Optimizing Parameters via the Gradient Descent Method

It has already been seen in the previous section that the problem of finding the right model for machine learning is finally reduced to optimization problem of minimizing the objective/cost function $J(w)$ of parameter (s) 'w'.

In machine learning problems, the function $J(w)$ representing the difference between the ground truth 'y' and hypothetical output $\hat{y}$ is not known generally as an analytical function that is, the explicit form of $J(w)$ is not known. $J(w)$ is known in the form of numerical values and therefore optimizing $J(w)$ for the parameter 'w' is done by using numerical iterative

techniques/algorithms rather than analytical methods for optimization [11], [9]. The iterative method discussed here is the gradient descent method which can be used efficiently to find the optimal value of 'w' when $J(w)$ is just known as black box (i.e., the form of $J(w)$ is hidden or not readily available). The gradient descent method with Cauchy as its original attributor is also called the method of steepest descent [10]. A similar method was proposed by Hadamard in 1907 [7]. The gradient descent method is a first order method in the sense that it uses the first order derivative(s) related to the objective function [5], [3]. We discuss the gradient descent method for two cases when 'w' is a scalar (i.e., a number) and 'w' is a vector.

3.1. Gradient Descent method (for scalar case)

For a scalar value of 'w' the iterative formula for the gradient descent method is:

$$w^{(k+1)} = w^{(k)} - \alpha\left(\frac{dJ}{dw}\right) \tag{1}$$

where α is a positive arbitrary parameter called learning rate and is chosen suitably by the user so that the method in (1) converges (i.e., the cost function $J(w)$ gets minimized, or equivalently (dJ/dw) tends to zero, as the method proceeds; (dJ/dw) on the right hand side of (1) is calculated at $w^{(k)}$).

Thus in this method, the task is to improve upon the guess 'w' by starting with an initial guess $w^{(0)}$ in the beginning and then obtaining $w^{(1)}, w^{(2)}, w^{(3)} \ldots$ by the use of equation (1) successively, till the desired accuracy is reached.

3.2. Gradient descent method (for vector case)

As a generalization of the scalar case, if 'w' is a vector, that is, 'w' has more than one component, then the general gradient descent algorithm formula is:

$$w^{(k+1)} = w^{(k)} - \alpha(\nabla_w J) \tag{2}$$

where $w^{(k)}$ is the value of 'w' at the k^{th} iteration, $w^{(k+1)}$ is the value of 'w'

at the $(k+1)^{\text{th}}$ iteration, 'α' which is called the learning rate is an arbitrary positive constant chosen by user so that the method in (2) converges to the optimal value of w, the vector $\nabla_w J$ or grad J is gradient of J with respect to w; $\nabla_w J$ on the right hand side of (2) is calculated at $w^{(k)}$).

The method specified in (2) is applied as follows:

Step 1. Fixing α, ε and the stopping criteria

In this step three things are fixed:

(i) The parameter α

α is a hyper parameter that must be set before the learning starts. An appropriate value of 'α' is chosen so that either $J(w)$ decreases or vector $\nabla_w J$ tends to 0 as the method (2) is applied. Choosing an appropriate hyper parameter 'α' is a problem in its own right.

(ii) Choosing the accuracy ε and the stopping criteria.

After choosing the value ε as, say, 10^{-6} i.e., up to 6 decimal places, the user is required to choose a stopping criteria which can be chosen in any of the following forms:

Applying the method till $| J(w^{(k+1)}) - J(w^{(k)}) | \leq \varepsilon$

or applying the method till $\| \nabla_w J(w^{(k)}) \| \leq \varepsilon$

or applying the method till $\| w^{(k+1)} - w^{(k)} \| \leq \varepsilon$

for a suitable norm chosen by the user.

Step 2. Starting the method using an initial guess $w^{(0)}$

Using an initial guess $w^{(0)}$ which is guessed by the user and the formula in (2), $w^{(1)}$ is calculated as: $w^{(1)} = w^{(0)} - \alpha \nabla_w J$

Since J in machine learning is only available as a black box i.e., no explicit expression is available for J, therefore $\nabla_w J$ which is grad J is generally calculated numerically by approximating it using finite difference approximations for calculating $(\partial J / \partial w_i)$ where $w_1, w_2, w_3 \ldots$ are components of vector w.

Step 3. Calculating successive values of w

Using $w^{(1)}$, the value of $w^{(2)}$ is calculated by use of formula (2). Successive values of $w^{(k)}$ are calculated till the stopping criterion is satisfied.

3.3. Illustration of the gradient descent method

We illustrate the gradient descent method for an arbitrary cost function $J(w)$ whose analytical form is known. Let $J(w) = w_1^2 + +w_2^2 + 4$; $\vec{w} = (w_1, w_2)^t$. It can be noted that the minimum of the cost function $J(w)$ is 4 and it is attained for $w_1 = 0, w_2 = 0$ i.e., for $\vec{w} = (w_1, w_2)^t = (0, 0)^t$

$$\text{Now vector } \nabla_w J(w) = \begin{pmatrix} \frac{\partial J}{\partial w_1} \\ \frac{\partial J}{\partial w_2} \end{pmatrix} = \begin{pmatrix} 2w_1 \\ 2w_2 \end{pmatrix} \tag{3}$$

Now using (3) in the iterative formula for gradient descent method for the vector case, that is, using (3) in formula (2), we get the corresponding iterative method in equations (4) and (5) taken together as:

$$w_1^{(k+1)} = w_1^{(k)} - \alpha(2w_1^{(k)}) \tag{4}$$

$$w_2^{(k+1)} = w_2^{(k)} - \alpha(2w_2^{(k)}) \tag{5}$$

Choosing $w^{(0)} = (3, 4)^t$ and $\alpha = 0.1$, say, On applying the method given in equations (4) and (5), we have the following results as shown in Table 1

Table 1. Iterative values of the parameter.

k	$w_1^{(k)}$	$w_2^{(k)}$	$w_1^{(k+1)}$	$w_2^{(k+1)}$	J
0	3	4	2.4	3.2	29
1	2.4	3.2	1.92	2.56	20
2	1.92	2.56	1.536	2.048	14.24

It can be noted from the last column of Table 1 that for the value of $\alpha = 0.1$, taken arbitrarily the cost function J gets minimized with each

iteration. In fact after thirty iterations the value of J gets optimized. For this particular value of α the method converges, but in general for an arbitrary value of α, the method may diverge, or even oscillate.

4. Conclusion

In this paper we have discussed the machine learning mechanism for which the optimization method, namely the gradient descent method which is a first order iterative method for finding the local minimum of a function (whether linear or nonlinear) can be suitably used. It has been described how the problem of determining a suitable machine learning model from a hypothetical mathematical function with parameters gets reduced to a numerical optimization problem, which can be solved by using sample data and the gradient descent optimization algorithm. It has been discussed as to how gradient descent method can be used efficiently to find the optimal values of parameters, when the objective function to be optimized is unknown or not readily available. It has been seen that the application of the numerical optimization gradient descent algorithm requires setting of hyper parameter α. Choosing the right value of 'α' is a part of algorithm design because the optimization method may converge (slowly/rapidly), diverge or oscillate depending upon the value of learning rate 'α' chosen. In neural networks and deep learning research problems, the design of hyper parameter and finding the optimal hyper parameters is problem open to future research [6], [16].

References

[1] Asad Abdi, Three types of Machine Learning Algorithms, (2016). DOI:10.13140/RG.2.2.26209.10088.

[2] E. Alpaydin, Introduction to Machine Learning, 3rd Edition, The MIT Press, (2014).

[3] L. Bottou, Large-scale machine learning with stochastic gradient descent, In Proceedings of COMPSTAT'2010. Physica-Verlag/Springer, Heidelberg (2010), 177-186.

[4] L. Bottou, F. E. Curtis and J. Nocedal, Optimization methods for large-scale machine learning, Siam Review 60(2) (2018), 223-311.

[5] C. Darken, J. Chang and J. Moody, Learning rate schedules for faster stochastic gradient search, Neural Networks for Signal Processing II, Proceedings of the IEEE Workshop, (September) (1992), 1-11.

[6] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. A. Ranzato, A. Senior, P. Tucker, K. Yang and A. Y. Ng, Large scale distributed deep networks, Advances in Neural Information Processing Systems (NIPS) (2012), 1-11.

[7] Jacques Hadamard, Mémoire sur le problème danalyse relatif à Véquilibre des plaques élastiques encastrées. Mémoires présentés par divers savants éstrangers à l'Académie des Sciences de l'Institut de France 33 (1908).

[8] P. S. Janardhanan, Project repositories for machine learning with TensorFlow, Procedia Computer Science 171 (2020), 188-196.

[9] N. S. Kambo, Mathematical Programming Techniques, Revised Edition, Affiliated East-West Press, (1991).

[10] C. Lemaréchal, Cauchy and the Gradient Method, Doc Math Extra (2012), 251-254.

[11] S. Marshland, Machine Learning: An algorithm Perspective, CRC, (2009).

[12] T. Mitchel, Machine Learning, Mc Graw- Hill, (1997).

[13] D. Newton, F. Yousefian and R. Pasupathy, Stochastic gradient descent: recent trends, Recent Advances in Optimization and Modeling of Contemporary Problems (2018), 193-220.

[14] F. Prior, J. Almeida, P. Kathiravelu, T. Kurc, K. Smith, T. J. Fitzgerald and J. Saltz, Open access image repositories: high-quality data to enable machine learning research, Clinical radiology 75(1) (2020), 7-12.

[15] S. J. Reddi, A. Hefny, S. Sra, B. Poczos and A. Smola, On variance reduction in stochastic gradient descent and its asynchronous variants, arXiv preprint arXiv:1506.06840, (2015).

[16] S. Ruder, An overview of gradient descent optimization algorithms, Retrieved from http://arxiv.org/abs/1609.04747, (2016).

[17] Shagan Sah, Machine Learning: A Review of Learning Types (2020). DOI:10.20944/preprints202007.0230.v1

[18] S. Shalev, Shwartz and S. Ben-David, Understanding Machine Learning from Theory to Algorithms, Cambridge University Press, (2014).

[19] C. Snijders, U. Matzat and U.-D. Reips, Big Data: Big gaps of knowledge in the field of Internet, International Journal of Internet Science 7 (2012), 1-5.

[20] S. Sun, Z. Cao, H. Zhu and J. Zhao, A survey of optimization methods from a machine learning perspective, IEEE Transactions on Cybernetics 50(8) (2019), 3668-3681.

[21] W. Wei, B. Zhou, R. Maskeliūnas, R. Damaševičius, D. Połap and M. Woźniak, Iterative design and implementation of rapid gradient descent method, In International Conference on Artificial Intelligence and Soft Computing, Springer, Cham. (2019), 530-539.

[22] A. Wissner-Gross, Datasets Over Algorithm, Edge.com. Retrieved 8 January (2016).

[23] X. S. Yang, Optimization algorithms, In Computational optimization, methods and algorithm. Springer, Berlin, Heidelberg (2011), 13-31.

[24] G. L. Zhang, H. H. Lin, D. B. Keskin, E. L. Reinherz and V. Brusic, Dana-Farber repository for machine learning in immunology, Journal of Immunological Methods, 374(1-2) (2011), 18-25.

[25] https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research.