Research on the Application of Gradient Descent Algorithm in Machine Learning

Xin Wang
[1.] School of Electrical Engineering, Xi'an Shiyou University
[2.] Shaanxi Provincial Key Lab of Oil and Gas Well Measurement and Control Technology
Xi'an, China
e-mail:xinW_776@126.com

Liting Yan
[1.] School of Electrical Engineering, Xi'an Shiyou University
[2.] Shaanxi Provincial Key Lab of Oil and Gas Well Measurement and Control Technology
Xi'an, China
e-mail:19211030267@stumail.xsyu.edu.cn

Qizhi Zhang
[1.] School of Electrical Engineering, Xi'an Shiyou University
[2.] Shaanxi Provincial Key Lab of Oil and Gas Well Measurement and Control Technology
Xi'an, China
e-mail: zhangqz@xsyu.edu.cn

2021 International Conference on Computer Network, Electronic and Automation (ICCNEA) | 978-1-6654-4486-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCNEA53019.2021.00014

***Abstract*—The gradient descent algorithm is a type of optimization algorithm that is widely used to solve machine learning algorithm model parameters. Through continuous iteration, it obtains the gradient of the objective function, gradually approaches the optimal solution of the objective function, and finally obtains the minimum loss function and related parameters. The gradient descent algorithm is frequently used in the solution process of logical regression, which is a common binary classification approach. This paper compares and analyzes the differences between batch gradient descent and its derivative algorithms — stochastic gradient descent algorithm and mini- batch gradient descent algorithm in terms of iteration number, loss function through experiments, and provides some suggestions on how to pick the best algorithm for the logistic regression binary task in machine learning.**

Keywords—Gradient Descent, Logistic Regression, Machine Learning

I. INTRODUCTION

Gradient descent is an iterative optimization algorithm that finds the smallest value of a function. Through continuous iteration, it obtains the gradient of the objective function, gradually approaches the optimal solution of the objective function, and finally obtains the smallest loss function and related parameters. The conventional gradient descent algorithm trains all of the samples every time, which extends the training time and may affect the final training effect. As a result, two new gradient descent algorithms are developed: the stochastic gradient descent algorithm and the Mini-batch gradient descent algorithm[1].

In a general sense, a logistic regression analysis model is a linear regression analysis model that is commonly used in data mining, economic forecasting, medicine, and other fields. It's essentially a conditional probability-based discrimination model. At present, logistic regression is mainly used in the medical field, such as exploring the main factors leading to a certain disease or predicting the possibility of the occurrence of the disease according to the factors affecting the disease [2].

II. GRADIENT DESCENT ALGORITHM

The gradient descent method is the most basic method for solving unconstrained optimization problems, as it considers the negative gradient direction to be the minimum of the objective function [3]. During training for algorithms like neural networks and regression analysis, gradient descent is commonly used to minimize the loss function.

A. Gradient descent algorithm derivation

The general linear regression problem can be expressed as:

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n = \sum_{i=0}^{n} \theta_i x_i = \theta^T x \qquad (1)$$

Where: $h_\theta(x)$ is the predicted value of linear

978-1-6654-4486-6/21/$31.00 ©2021 IEEE
DOI 10.1109/ICCNEA53019.2021.00014

regression; θ_0 is the Bias; $\theta_1 \cdots \theta_n$ is the parameters that need to be solved; $x_1 \cdots x_n$ is the sample attribute.

In fact, the predicted value of linear regression is not capable of correspond to the real value, and there is often an error between the predicted value and the real value. We call this error item ε.

$$y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)} \tag{2}$$

Where: i is sample number; $\theta^T x^{(i)}$ is predictive value; $y^{(i)}$ is true value. It is assumed that the error terms are independently identically distributed and follow a Gaussian distribution with a mean of zero and a variance of θ^2.

Equation (2) is substituted into the Gaussian distribution expression of the error term ε:

$$p(y^{(i)} \mid x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}) \tag{3}$$

Take logarithm on both sides of Equation (3), and finally get the loss function:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \tag{4}$$

Where: m is number of training samples; $x^{(i)}$ is attributes of the sample; $y^{(i)}$ is the predicted value of each sample.

Take the partial derivative of the loss function $J(\theta)$ with respect to θ_j and set the partial derivative equal to zero:

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} = 0 \tag{5}$$

According to Equation (5), the calculation formula for solving the parameter vector is:

$$\theta_j^* = \theta_j + \alpha \frac{1}{m} \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)} \tag{6}$$

The iterative algorithm of Equation (6) is also known as the batch gradient descent algorithm.

B. Improved gradient descent algorithm

Batch gradient descent algorithm needs to consider all samples in the training process. As a result, this algorithm will select the overall best path in each iteration, making it easier to find the best solution. However, if the number of samples and attribute values is huge, the input matrix made up of samples will be large, resulting in a slow iteration speed and failure to achieve the desired effect.

1) *Stochastic gradient descent algorithm*

The stochastic gradient descent algorithm randomly selects one sample from all samples for iterative training in each iteration [4][5]. This approach needs less computation in each iteration for large-scale sample data, and the convergence speed is clearly faster than other algorithms, resulting in high performance. The iterative formula of stochastic gradient descent algorithm is:

$$\theta_j^* = \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)} \tag{7}$$

2) *Mini-batch gradient descent algorithm*

Although the convergence speed of the stochastic gradient descent algorithm is fast, each iteration of a randomly selected sample cannot guarantee the direction of convergence for each iteration [6]-[7]. The final outcome could be worse if the randomly chosen sample points are irregular points. The advantages and drawbacks of batch gradient descent and stochastic gradient descent algorithms are thoroughly considered in mini-batch, and the required number of samples from all the samples are chosen for training in each iteration. The iterative formula of gradient descent in mini-batch:

$$\theta_j^* = \theta_j + \alpha \frac{1}{64} \sum_{k=i}^{i+63} (y^{(k)} - h_\theta(x^{(k)})) x_j^{(i)} \tag{8}$$

III. LOGICAL EGRESSION

Logical regression is a probabilistic regression model, which is a kind of generalized linear regression model [8]-[9]. It mainly reflects a mapping relationship between independent variables and dependent variables with dichotomous properties, that is, the known independent variables can be used to predict the values of a group of discrete variables.

1) *Sigmoid function*

Logical regression maps any input to the interval [0,1] by introducing the Sigmoid function. The regression predicted value is mapped to the Sigmoid function to complete the transformation from numerical value to probability value, resulting in the classification prediction. The sigmoid function as follows:

$$g(z)=\frac{1}{1+e^{-z}} \tag{9}$$

It can be seen from the sigmoid function that the range of independent variable z is $(-\infty,+\infty)$, and the range of function value is [0,1] . The relationship between independent and dependent variables is:

$$g(z)=\begin{cases}1 & , \quad z>0\\ 0.5 & , \quad z=0\\ 0 & , \quad z<0\end{cases} \tag{10}$$

Because of the Sigmoid function's unique relationship between the independent and dependent variables, we can consider all samples with function values greater than or equal to 0.5 to be positive when making classification predictions and all samples with a function value less than 0.5 are classified as negative.

2) Logic regression solving

The prediction function $h_\theta(x)$ of linear regression can be written:

$$h_\theta(x)=g(\theta^T x)=\frac{1}{1+e^{-\theta^T x}} \tag{11}$$

Therefore, the probability of obtaining the positive is $h_\theta(x)$, the probability of obtaining the negative is $1-h_\theta(x)$.

$$p(y\mid x;\theta)=(h_\theta(x))^y(1-h_\theta(x))^{1-y} \tag{12}$$

So, the likelihood function is:

$$L(\theta)=\prod_{i=1}^{m}p(y\mid x;\theta)=\prod_{i=1}^{m}(h_\theta(x_i))^{y_i}(1-h_\theta(x_i))^{1-y_i} \tag{13}$$

The logarithm is taken on both sides of the equation (13), and the iterative formula of the parameter is obtained according to the gradient decrease method is:

$$\theta_j^{*}=\theta_j-\alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)x_j^{(i)} \tag{14}$$

The logical regression model predicts not only the actual "category," but also its approximate probability estimation, also known as "confidence." Furthermore, the classification is directly modelled, and there is no need to assume data distribution in advance to avoid problems caused by incorrect distribution.

IV. CASE ANALYSIS

The instance selected data set is a data set that predicts whether the student is admitted. The student's two test scores are the input independent variable, and the probability of admission is the output dependent variable. The dataset consists of 100 samples. Partial data of the sample are shown in Table 1.

TABLE I. PARTIAL SAMPLE DATA

	Score 1	Score 2	Admitted
1	95.8615507093572	38.22527805795094	0
2	75.01365838958247	30.60326323428011	0
3	82.30705337399482	76.48196330235604	1
4	69.36458875970939	97.71869196188608	1
5	39.53833914367223	76.03681085115882	0

Where: 0 indicates that the student not admitted;1 means the student admitted.

1) Comparison of convergence speed of different gradient descent algorithms with the same number of iterations

Suppose the linear fitting function is:

$$h_\theta(x)=\theta_0+\theta_1x_1+\theta_2x_2 \tag{15}$$

According to the parameter update formulas of three different gradient descent algorithms, 8000 iterations were carried out, and the output results are shown in Table 2.

TABLE II. GRADIENT DESCENT ALGORITHM PARAMETER VALUES

	θ_0	θ_1	θ_2
BGD	-0.00048126	0.00777571	0.00305656
SGD	-0.00049396	0.00782276	0.00268035
Mini-BGD	-0.00048275	0.00770747	0.00294887

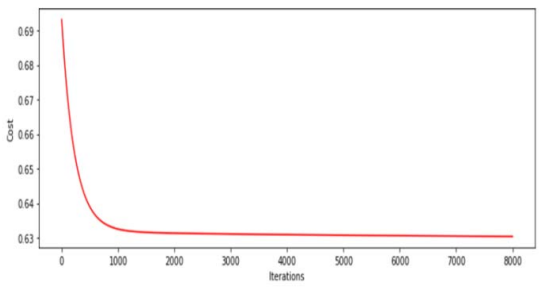

Fig. 1. Diagram of loss and number of iterations under BGD

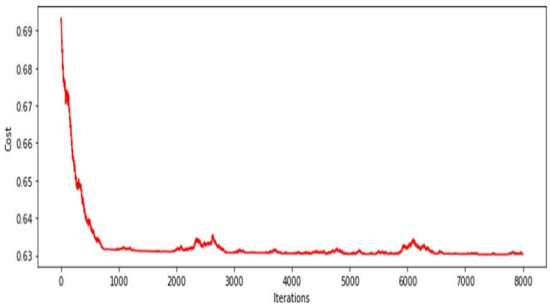

Fig. 2. Diagram of loss and number of iterations under SGD

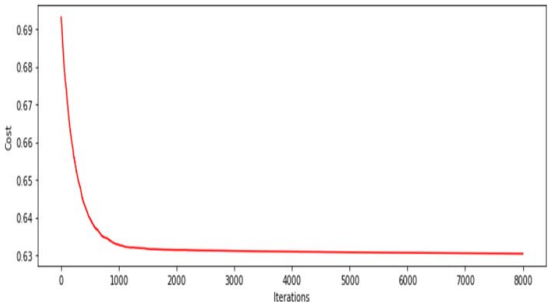

Fig. 3. Diagram of loss and number of iterations under Mini-BGD

It can be obtained from the graph: The loss function of stochastic gradient descent algorithm has obvious fluctuation near the optimal point, which indicates that it has more noise points. The convergence rate is slightly faster when the mini-batch gradient descent method is used.

2) The Influence of Iteration Number and Learning Rate on Gradient Descent Algorithm

It can be seen from the gradient descent method's derivation process that iteration times and learning rate have a significant impact on the final loss. If the learning rate is too slow, there will be too many iterations. If the learning rate is too fast, the local minimum can be missed, resulting in an inability to converge.

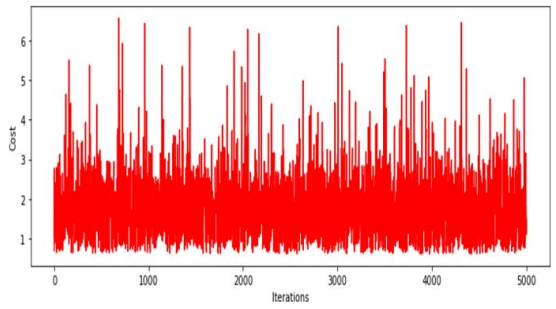

Fig. 4. SGD, Iterations=5000, learning rate=0.001

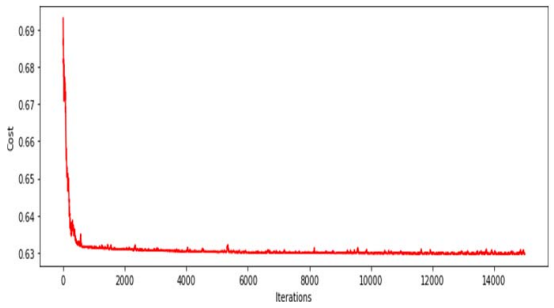

Fig. 5. SGD, Iterations=15000, learning rate=0.000002

The loss function fluctuates significantly when the number of iterations is small and the learning rate is high, as seen in Figure 4 and Figure 5. The loss function has modified significantly after increasing the number of iterations and decreasing the learning rate. However, the stability remains low, necessitating a slow learning pace to meet it.

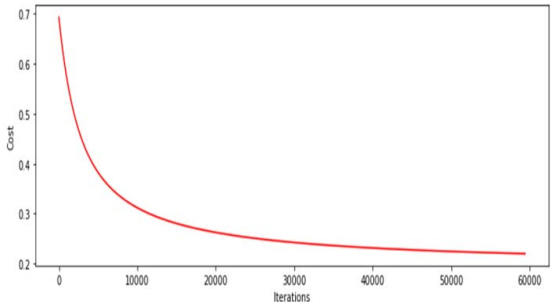

Fig. 6. Diagram of loss and number of iterations under Mini-BGD

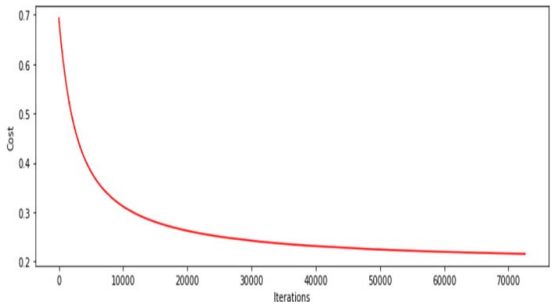

Fig. 7. Diagram of loss and number of iterations under SGD

In order to reduce the loss function smaller, the data samples are standardized. It can be seen from the results in Figure 6 and Figure 7, the loss value can reach 0.22 after data standardization, which is a great improvement compared with the previous one. Stochastic gradient descent is faster, but requires more iterations; Despite the fact that mini-batch gradient descent is slower than

stochastic gradient descent, the number of iterations is nearly 10,000 times lower.

V. CONCLUSIONS

Different gradient descent methods can be used in realistic applications depending on the data sets. The stochastic gradient descent method has more noise points when the training data set sample size is small, and the fluctuation near the optimal point is evident, making it easy to fall into the local optimal solution; When the sample size of the training data set is large, the batch gradient descent algorithm's complexity is high, and parameter iteration's convergence rate is slow. When the stochastic gradient descent method is used to set the parameters, however, it typically only takes a few iterations to achieve a better fitting effect. Furthermore, the mini-batch gradient descent algorithm is obviously superior to other algorithms in terms of loss function and convergence speed.

When using gradient descent method to optimize parameters, the selection of learning rate and iteration times is particularly important. The loss function will converge slowly if the learning rate is too low; if the learning rate is too high, it is likely to exceed the global optimum. Continuously, if the number of iterations is too large, the loss function will fall into the local optimal solution; if the number of iterations is too low, the loss function will fluctuate a lot, which is not good for the final result.

ACKNOWLEDGMENT

This research is supported by the foundation project: Xi'an Shiyou University's Graduate Innovation and Practical Ability Training Program.

REFERENCES

[1] Chen X W, Lin X. Big data deep learning: challenges and perspectives [J]. IEEE access, 2014, 2: 514-525.

[2] Langer D L, Van der Kwast T H, Evans A J, et al. Prostate cancer detection with multi‐parametric MRI: Logistic regression analysis of quantitative T2, diffusion‐weighted imaging, and dynamic contrast‐enhanced MRI [J]. Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine, 2009, 30(2): 327-334.

[3] Qu Q, Zhang Y, Eldar Y C, et al. Convolutional phase retrieval via gradient descent[J]. IEEE Transactions on Information Theory, 2019, 66(3): 1785-1821.

[4] Zhou F, Cong G. On the convergence properties of a $ K $-step averaging stochastic gradient descent algorithm for nonconvex optimization [J]. arXiv preprint arXiv:1708.01012, 2017.

[5] Manogaran G, Lopez D. Health data analytics using scalable logistic regression with stochastic gradient descent [J]. International Journal of Advanced Intelligence Paradigms, 2018, 10(1-2): 118-132.

[6] Huo Z, Huang H. Asynchronous mini-batch gradient descent with variance reduction for non-convex optimization[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2017, 31(1).

[7] Yazan E, Talu M F. Comparison of the stochastic gradient descent based optimization techniques[C]//2017 International Artificial Intelligence and Data Processing Symposium (IDAP). IEEE, 2017: 1-5.

[8] Sur P, Candès E J. A modern maximum-likelihood theory for high-dimensional logistic regression[J]. Proceedings of the National Academy of Sciences, 2019, 116(29): 14516-14525.

[9] Kuha J, Mills C. On group comparisons with logistic regression models [J]. Sociological Methods & Research, 2020, 49(2): 498-525.