Machine Learning Algorithm Based on Big Data

HaiYan Hao*

Hainan College of Software Technology, Qionghai, Hainan, China, 571400

*Corresponding author e-mail: Haiyan163.com

Abstract. Machine learning technology is indispensable for big data. In machine learning, data on a large scale can improve the accuracy of the model. However, complex machine learning algorithms require the key technology of distributed memory computing in time and performance. Big data memory computing can implement the parallel operation of the algorithm, which is conducive to the processing of big data sets by the machine learning algorithm. Hence, a nonlinear machine learning algorithm implemented in the big data memory environment is proposed in this paper, where data compression, biased sampling or loading based on the implementation is optimized. To fully configure resources for the script running by batch, we also implemented a machine learning framework to schedule the optimized algorithm mentioned above. The experimental results showed that the mean error of the three algorithms after optimization was reduced by 40%, and the mean time was reduced by 90%.

Keywords: Data Compression, Biased Sampling, Stochastic Gradient Descent, Neural Network, Support Vector Machine

1. Introduction

The core of big data is to use the value of data, and machine learning is the core technology that harnesses the value of data. With the development of science and technology, computers process and collect hundreds of millions of data daily[1]. Hence, the application of machine learning to process large-scale data and generate more accurate models is crucial, but the training efficiency of the machine learning algorithm in distributed memory computing requires urgent improvement[2-3]. As a big data processing engine, big data can be processed in parallel by using distributed data sets. The most prominent characteristics of big data is memory computing, which is about 100 times faster than the processing speed of big data MapReduce[4-5]. In this paper, mainly a nonlinear machine learning algorithm is implemented in big data memory computing and improves the efficiency of data training by optimizing the multi-layer variable neural network, BPPGDSVM and machine learning algorithm. The big data framework is implemented on this basis to schedule the optimized algorithm. The purpose is to fully obtain system resources to run scripts in batches and produce the results[6].
In this paper, a large data ml scheduling framework of machine-like learning algorithm is implemented to generate parameter scripts in batch for the optimized algorithm, including system parameters, important algorithm parameters, etc. The system parameters are usually obtained by batch reading from XML files, and some parameters of the algorithm are obtained through user input. In this way, an efficient running environment can be configured to fully leverage the system resources to run the optimized algorithm. In the process of algorithm optimization, machine learning algorithm optimizes the algorithm mainly through biased data sampling; SVM optimizes the algorithm primarily through data compression, data loading and data biased sampling; the multi-layer variable neural network implemented in this paper adopt the compression and biased sampling of data in the optimization process. For the training of the algorithm, this paper focuses on the iterative training of nex-dcp30 and gas sensor data set in a big data cluster. The experimental results suggest that the sum of the squared error of the optimized machine learning algorithm is 0.53 and the time is 0.28 where no optimization is performed. The time of the optimized BPPGDSVM is reduced by 96.7% without the loss of accuracy. The minimum mean square error of the multi-layer variable neural network implemented in this paper is 70% lower than that of the single-layer variable neural network implemented in Constantinovoglis laboratory when dealing with the single hidden layer, and the maximum mean square error is 70%. The error is 72 orders of magnitude lower. Through optimization, the mean squared error is 0.61.

2. Big Data Environment

The big data yarn is the framework of resource management and job monitoring, which is responsible for container parallel work. In the big data distributed file system (HDFS), the data set is divided into several blocks and stored in different nodes. Meanwhile, HDFS fully controls these blocks and maintains fault tolerance.

Big data is a fast and general-purpose big data processing engine running in yarn, Apache mesos, or EC2 environment, which is conducive to processing extensive data-parallel iterative computing. The core of big data is to use elastic distributed data set (RDD). RDD is a read-only data set across data partitions. On RDD, perform parallel operations to produce actual results, or convert an RDD to other forms of RDD through application transformation. In further reuse, RDD can be cached in cluster memory to prevent unnecessary storage I / O, thereby accelerating data processing. In this paper, we mainly use the yarn-based big data cluster framework for data set training. The master node of big data is primarily responsible for driver scheduling, and the slave node is primarily responsible for executing tasks in the executor. The big data cluster architecture is shown in Figure 1 as follows.
3. Implementation of machine learning framework

In the machine learning algorithm, big data jobs are synthesized by logs through scripts and parameters. A big data framework of machine learning algorithm-like function is implemented in this paper based on big data to perform distributed memory computing. This framework mainly reads parameters to synthesize and run scripts by batch to output the results to the specified files.

The purpose of the big data framework is to make the parameters involved in data set training diverse. If the parameter settings are unreasonable, it will be a waste of resources, and the best training effect will not be achieved, with even errors such as GC and error parameters. Big data framework can isolate the relationship among users, algorithms and the environment to the greatest extent. Users’ knowledge of the internal processing mechanism of the algorithm and the status of the cluster environment is not required. The results suggest that the big data framework can strictly specify the performance in a distributed environment. When a batch script is running, the big data framework contains many running scripts and saves them to the script library. They wait for resources to execute. These running scripts are equivalent to jobs. When a job uses resources in the system and runs, other jobs must wait in turn. Until the running job finishes running and releases resources, other jobs in the script library have no access to resources for continuous running until the running of all scripts is completed.

4. Processing optimization of data in this paper

4.1. Data compression

In this paper, we use the data compression algorithm to optimize the algorithm. In the process of implementation, we first judge the type of data set and then convert it to the form of a sparse matrix or dense matrix by type. Then data compression is carried out, where the dimensionality reduction equation of SVD [8] is mainly used as follows:

$$A_{m \times n} \approx U_{m \times k} \sum_{i=1}^{k} V_{i \times n}$$  \hspace{1cm} (1)

Based on the equation (1), the dimension of the data is reduced. When a value in the data has little effect on the result, the characteristic value can be deleted. In many cases, the sum of singular values of the first 10% or even 1% account for more than 99% of the sum of all singular values. The singular values...
are arranged in descending order, and the first k is used to approximate the matrix.

### Table 1. Parameter description of data compression algorithm

| Parameter   | Value                    | Specific description                               |
|-------------|--------------------------|---------------------------------------------------|
| nnRatio     |                          | How to extract compressed data                    |
| itqitN      |                          | Represents the number of training times during data compression |
| itqratioN   |                          | Indicates to select one of several attributes for calculation |
| UpBound     |                          | Upper limit value                                 |

In this paper, the data compression part is the operation after data loading and before data training. Data compression can support the data format of the sparse and dense matrices.

#### 4.2. Biased sampling of data

In data partition, the RDD stochastic split method of big data only needs to set the extraction ratio to sample evenly. However, the disadvantage is that it is possible to make the selected samples far away from the center point or discrete through stochastic sampling. Hence, the mean square error of training is substantial, and the training effect is not ideal. Therefore, this paper will use the method of biased sampling to avoid this problem. It is based on the distribution of samples for data extraction, through the density of samples for data extraction. The more concentrated the sample distribution, the greater the probability of sampling, and vice versa. In this paper, we use equation (2) to sample.

\[
k(x_1, \ldots, x_d) = \left(\frac{3}{4}\right)^d \frac{1}{B_1 B_2 \cdots B_d} \prod_{i=1}^{d} \left(1 - \left(\frac{x_i}{B_i}\right)^2\right)
\]

(2)

Where Bi is the bandwidth of the i-th eigenvalue, D is the number of eigenvalues, and Xi is the i-th eigenvalue. Bi can be obtained based on equation (3).

\[
B_i = \sqrt{S_i} \left| S \right|^{-\frac{1}{D+1}}
\]

(3)

#### 5. Experiment and analysis

Software version information used in this paper is as follows: big data version 2.2.0; Java version 1.8.0; Scala version 2.12.3; SBT version 1.0.0.

Server node information: server (32GB memory, 12 cores) \times 5, including one master node and four slave nodes. The system version used is centos7. By setting the VCores value and available memory size in the conf file of big data, the number of VCores of each node is 20, and the corresponding available memory size is about 20GB.

In this paper, two representative data sets are trained and tested. The specific information about the data sets is as follows:

1. The data set in the NASA Earth exchange downscaling climate prediction (nex-dep30) includes 55 year (1950-2005) of historical climate information in the U.S. region (longitude and latitude range are within the coverage region). In this paper, the data information of 1950-1959 is mainly intercepted, which is about 79gb. Each row of data contains one label value and 360 eigenvalues, of which every 36 eigenvalues represent one year's information. Among the 36 eigenvalues, 12 indicate the highest temperature of the earth's surface in 12 months, 12 indicate the lowest temperature of the earth's surface...
in 12 months, and the other 12 indicate the precipitation in 12 months. The data are normalized and stored in the file.

(2) Gas sensor data set under dynamic gas mixing (gas sensor data set in this paper) is mainly collected in a gas transportation platform facility of chemosignals laboratory, Institute of biology, San Diego, University of California. It mainly includes 18 characteristic values and a label value. The first column of feature value indicates the set value of methane (or CO) concentration, the second column indicates the set value of ethylene concentration, and the remaining 16 columns indicate the records of the sensor array. In this paper, 18 eigenvalues are used for training. At the same time, the data is normalized and stored in the file.

(3) Test Results from Neural Network

In this paper, the multi-layer neural network and the single-layer neural network implemented by Constantinos voglis laboratory in 2006 train the nex-dep30 data set respectively to compare the change of MSE and iteration time. The experimental results show that under the same conditions, the minimum mean square error of the multi-layer variable neural network is 70% lower than that of the single-layer neural network in Constantinos voglis laboratory, and the maximum mean square error is 72 orders of magnitude lower, and the convergence speed is faster. In other words, the multi-layer neural network can accurately predict the data set.

Figure 2 shows that MSE obtained by optimization training is significantly reduced. After data compression and biased sampling, the mean MSE of MNN is about 0.61 of the original MNN, about 0.92 of data compression optimization. Hence, MSE has been properly optimized. At the same time, the optimized MNN training time is 0.8 of the original MNN time. In the training of large-scale data, it can shorten the training time effectively and improve operation efficiency. The time gradient comparison is shown in Figure 12 as follows.

![Figure 2. Comparison of time in big data environment](image)

6. Conclusions

The core of this paper is about the implementation and optimization of machine learning algorithm, BPPGDSVM and multilayer neural network code based on the big data memory computing environment while developing a machine learning framework to implement the scheduling of the above optimization algorithm at the same time. The aforementioned work is conducted to achieve the goal of improving
running speed and accuracy. Other technologies will be added to optimize the work in the later stage further.

References
[1] Chen, M. , Hao, Y. , Hwang, K. , Wang, L. , & Wang, L. . (2017). Disease prediction by machine learning over big data from healthcare communities. IEEE Access, 1-1.
[2] Zhou, L. , Pan, S. , Wang, J. , & Vasilakos, A. V. . (2017). Machine learning on big data: opportunities and challenges. Neurocomputing, 237, 350-361.
[3] Ding, J. , Hu, X. H. , & Gudivada, V. . (2017). A machine learning based framework for verification and validation of massive scale image data. IEEE Transactions on Big Data, 1-1.
[4] Gosavi, A. . (2004). A reinforcement learning algorithm based on policy iteration for mean reward: empirical results with yield management and convergence analysis. Machine Learning, 55(1), 5-29.
[5] Tsaih, R. H. , Kuo, B. S. , Lin, T. H. , & Hsu, C. C. . (2018). The use of big data analytics to predict the foreign exchange rate based on public media: a machine-learning experiment. IT Professional, 20(2), 34-41.
[6] M. H. Shahnas, D. A. Yuen, & R. N. Pysklywec. (2018). Inverse problems in geodynamics using machine learning algorithms. Journal of Geophysical Research Solid Earth, 123(1).