

Research And Implementation of Iterative MapReduce Based On BP Algorithm

Yu Yang^{1, a}, Longjun Zhang¹

¹ Department of Communication Engineering, Engineering College of Armed Police Force, Xi'an, China.

^amiaoyude@163.com

Keywords: MapReduce, BP algorithm, iteration, Control system in the K/TGR radioswitch.

Abstract. The Error Back Propagation is iterative, the way of implementing it using the iterative MapReduce framework is presented. To solve the shortage of the traditional MapReduce framework in iterative program, the iterative MapReduce framework has added a transmitting module. Through the simulation of the control system in the K/TGR radioswitch to get the training sample. Training the sample based on the traditional framework and the iterative framework on Hadoop. The experimental results show that the BP algorithm based on the iterative framework demonstrated faster speed and higher accuracy. The iterative framework can effectively reduce the training time and avoid the problem in iterative calculation.

Introduction

Error Back Propagation (BP) algorithm is the basic algorithm in the field of fault diagnosis, and it is widely used in the field of fault diagnosis[1]. With the improvement of the complexity of the mechanical equipment, the data scale of the BP algorithm to deal with the fault samples is increasing, and how to deal with large scale data is becoming more and more important. MapReduce computing model is a distributed parallel computing model proposed by Google. It can organize the cluster to deal with large scale data sets, which has a huge advantage for the rapid processing of large scale data[2]. MapReduce is a distributed computing framework in cloud computing platform. It also can accurately and timely provide the powerful support for fault diagnosis.

BP algorithm is an iterative algorithm, and the traditional MapReduce computing framework exists defects in the iterative algorithm of the operation. In the process of operation, each iteration requires a task to be submitted, the efficiency is low, and the computation cost is high. Therefore, this paper uses an iterative MapReduce framework to carry out the optimization of the BP algorithm[4], and the data transfer function of the transfer module is used to solve the problem of multiple tasks submitted. Finally, the training samples obtained from the simulation experiment are compared with the BP algorithm based on the traditional MapReduce framework and the iterative MapReduce framework. In the experiment, the latter is more than 10 times of the former, and the correct rate is higher than the former 1% ~ 3%, which shows the advantage of the iterative framework used in the BP algorithm.

MapReduce parallel computing model

Traditional MapReduce framework. MapReduce computation model of the basic idea is "divide and conquer", through the "map" and "reduce" to implement the parallel computation. Map task is to partition the part of the original data for the specified operation. Reduce tasks for each Map task generated intermediate results for the mixed operation, all the final results generated by the Reduce task after a simple process to get the final result set.

Iterative MapReduce framework. Traditional MapReduce framework for the processing of the iterative algorithm has some shortcomings[3]. In the iterative algorithm, the output of each iteration is the input of the next iteration. The traditional MapReduce framework is used to compute the

weights to the nodes of the original Map task for the next round of calculation. This avoids the need to submit a task and restart a MapReduce calculation of the drawbacks of each iteration.

The BP algorithm is composed of three stages by using the iterative MapReduce framework. The functions of each stage are as follows.

Map stage: Map function receives sample data. According to the BP algorithm formula, forward propagation calculation and the calculation of the back propagation adjust weights are carried on. The weight value of each sample is recorded as $\Delta\omega$. Intermediate key value pair $\langle \text{key} = \omega, \text{value} = \Delta\omega \rangle$ is generated.

Reduce stage: Receiving intermediate key value same as ω is used as input value. The average value of all sample weight values is calculated and the average value is used as the final weight. According to the adjustment formula, the new weights are calculated, and the formula is as follows.

$$\omega_{\text{new}} = \omega_{\text{old}} - \sum_{i=1}^n \Delta\omega_i / n \quad (6)$$

N is sample number. $\Delta\omega_i$ is sample offset i .

Communication stage: When the actual output does not meet the expected conditions need to be calculated, the communication module transfers the latest weight data to the Map node, and the new round of the Map function is performed. When meeting the expected conditions, the communication module stops working.

Experimental results and analysis

BP algorithm sample data acquisition. A certain type of VHF radio air switch control system is simulated in MATLAB and the sample data required are got. According to the characteristics of the system structure, only the radio switch state is considered, and the failure mode of the radio switch can be obtained mainly from three kinds of modes. It includes Damage failure, Constant Gain Failure, Constant Deviation. The output of the sensor and controller is collected at $t=10s, 20s, 30s, 40s, 50s$ and is used as the BP algorithm training sample data. Data collected from $t=10s, 20s, 30s, 40s, 50s$ are shown in table 1. In the table each failure mode is listed as a representation of the two samples. The required number of sample data is got by adjusting in MATLAB.

Table 1 List of sample data

Fault type	Sensor					Actuator				
	t=10s	t=20s	t=30s	t=40s	t=50s	t=10s	t=20s	t=30s	t=40s	t=50s
Switch damage failure	0.3	0.3	0.3	0.3	0.3	0.7051	1.2186	1.6	1.6	1.6
	0.4	0.4	0.4	0.4	0.4	0.5851	1.0126	1.5708	1.6	1.6
Switch Constant Gain Failure	0.1729	0.3162	0.4256	0.4277	0.4277	0.9549	1.6	1.6	1.6	1.6
	0.2201	0.4102	0.5244	0.5382	0.5382	0.9211	1.4081	1.6	1.6	1.6
Switch Constant Deviation	0.8169	0.9446	1.0296	1.0541	1.0114	0.2059	0.2097	0.3403	0.2912	0.2806
	1.2488	1.2491	1.2495	1.25	1.25	0	0	0	0	0

Experimental environment. Experiments use Hadoop cloud computing environment. Hardware configuration contains one main nodes and twelve slave nodes. CPU using Core Quad Intel 2 2.4GHz CPU Q6600, 16GB memory. Operating system kernel is Linux, the Java version is 1.6.0-26, the Hadoop version is 1.2.0.

Experimental results. In the case of two different frameworks, the sample number is different, the time consumed by the MapReduce algorithm based on the traditional MapReduce and the iterative BP algorithm is completed. The algorithm is repeated 20 times, through the comparison and found that in

the case of fewer samples, the time required to complete the training of the two frameworks is not large. But when the number of samples increased to more than 10^5 , based on iterative framework of BP algorithm training time is significantly less than the training time of the traditional algorithm framework. Because when the number of samples is lower than 10^3 , algorithm iterations is less. The superiority of the iterative framework is not reflected. As the sample size increases, the number of iterations of the algorithm also increases, then the iterative framework for the traditional framework of the advantages of the embodiment, the consumption of time is significantly less than the latter.

The two framework of the BP algorithm training error, as shown. Through the contrast found that two different under the framework of training algorithm error number difference, based on the training error of the BP algorithm of iterative MapReduce than traditional MapReduce based training error is reduced in a certain degree, but not the extent. Because the iterative framework is mainly to the multiple iterations of the task mechanism has been optimized, to a certain extent, to improve the training of the correct rate. But there is no special training to improve the correct rate to do the work, so in this regard, the iterative framework than the traditional advantages of the framework is not obvious.

Conclusion

BP algorithm is commonly used in the field of fault diagnosis algorithm, the MapReduce framework is applied to the BP algorithm to solve the problem of slow computing speed, low efficiency in large data environment, can effectively improve the real-time fault diagnosis. In this paper, based on the analysis of the BP algorithm, according to the characteristics of the BP algorithm, the iterative MapReduce framework of the transfer module is applied to the algorithm. Based on the traditional framework and the iterative framework of the BP algorithm to do a comparative experiment, the results show that the iterative MapReduce framework with respect to the traditional MapReduce framework, can effectively reduce the time cost of the algorithm. In the correctness of the algorithm, the iterative framework also has a certain degree of improvement than the traditional framework. However, the research also found that the iterative framework in the data size is less than the obvious advantages of the embodiment, in order to improve the accuracy of the algorithm is not enough to play the function of the iterative framework. These questions will be more concrete research in the next step of work.

Summary

This work was financially supported by Engineering University of Armed Police Force basic research fund (WJY201307), Department of Communication Engineering background research fund(XJY201405).

References

- [1] Qi Bensheng, IEEE, J. Commun. (2010) 1460-1463.
- [2] Dean J, Ghemawat S, Communications of the ACM, J. Commun. 51(2008) 107-113.
- [3] RaoB T Reddy LSS, International Journal of Computer Applications,J. Commun. 34(2012) 28-32.
- [4]SyafaruddinKaratepe E,Hiyama T,16th International Conference on Intelligent System Application to Power System,J. Commun. (2011) 1-6.
- [5]Ekanayake J, LI H, ZHANG BJ, et al, Proceeding of the 19th ACM International Symposium on High Performance Distributed Computing,J. Commun. (2010) 810-818.