IMPLEMENTATION OF PARALLELIZING MULTI-LAYER NEURAL NETWORKS BASED ON CLOUD COMPUTING

Hai Wang
Yi Wang
Dongming MA
State Grid Jibei Information & Telecommunication Company
Zhangjiakou
wangh2011@yahoo.com

Abstract. Background: Cloud computing, as a technology developed under the rapid development of modern network, is mainly used for processing large-scale data. The traditional data mining algorithms such as neural network algorithm are usually used for processing small-scale data. Therefore, the calculation of large-scale data using neural network algorithm must be based on cloud computing. Materials and Methods: Firstly a Hadoop system was established taking MapReduce as the programming framework. Then the parallelized traditional data mining algorihtm was investigated based on cloud computing cluster to verify its feasibility in processing large-scale data. Finally, the speed and training precision of the algorithm were tested. Results: It was feasible to process large-scale data with cloud computing based parallelizing multi-layer neural network algorithm. The speed of parrallel processing was faster if data size was larger, especially if the sample was in a size of more than 1 million. It was more superior to the serial back propagation network in training preciseness. Conclusion: Parallelizing multi-layer neural network based on cloud computing platform can process large-scale data effectively in the perspectives of time and quality.

Keywords: Cloud computing, neural networks, layered representation, parallelization, speed, training preciseness, data, feasibility.

Significance statement

This study realized multi-layer neural network parallelization based on cloud computing and applied it for processing large-scale data. Compared to the serial back propagation algorithm, the algorithm showed high operation speed and training precision in processing the sample in a size of more than 0.6 million.

1. Introduction

Since cloud computing technology first came into being in 2007, it has been developed greatly [1]. With the further development of the information society, data explosion has stepped into our life. As the conventional data mining algorithms cannot process such large-scale and complex data [2-3], large-scale data need to be processed with the help of distributed parallel computing technology.

Cloud computing is capable of processing the massive data, therefore, combining cloud computing with the conventional data mining algorithm is a preferable solution to the problem of data processing. So far, quite a few scholars have carried out relevant researches.

For instance, in 2012, Aizenberg et al. [4] performed a thorough research on low density parity check (LDPC), multinuclear parallel idea and artificial neural network; by combining parallel idea and neural network technology with LDPC decoding effectively, they designed and implemented LDPC parallel decoding algorithm, LDPC decoding algorithm based on general neural network and LDPC decoding algorithm based on multilayer perceptron. In 2014, Nunez et al. [5] applied the modified artificial neural network back propagation algorithm to the identification of physical parameters of multi-layer soil mass in deep foundation pit excavation engineering. Compared with the conventional back analysis method, this algorithm was easy to master and implement. In 2012, Krawczak et al. [6] put forward a new algorithm that integrated genetic algorithm with adaptive deformation gradient learning algorithm and could be applied in multilayer feedforward neural network training. Based on the experiment on Hadoop cloud computing platform, this paper presents a study on the parallelization of multi-layer neural network, suggesting that the parallelized network is advantageous in dealing with large-scale data for its high speed and high quality. Therefore, this study contributes to the development of this field to some extent.

2. Methodology

2.1 Parallelization of neural networks

For the parallelization of neural networks, there were two approaches. The first one was node parallelization, i.e., network nodes were distributed on different machine nodes so as to achieve parallelization. However, with this approach, the integrity of similar networks would be destroyed after node distribution [7, 8]. In cloud computing, programmers could not directly control the specific nodes; therefore, parallelization of this kind of nodes could not be achieved in cloud computing. The second one was data parallelization. With this approach, each compute node had a complete network at the same initial state.

2.2 Implementation of MapReduce

As an open-source distributed cloud computing system, Hadoop is established based on MapReduce programming framework and Google file system (GFS). Operating on Hadoop, users can avoid lowlevel details of parallel programming, such as automatic parallelization, load balancing and disaster management; in addition, users can process large-scale data using simple parallel programming framework. So far, Hadoop has become the most representative mainstream platform for distributed computing [9-11].

Hadoop distributed file system (HDFS), MapReduce and Common compose the Hadoop system. With clear structure, this kind of parallel programming framework is easy to use. On the large-scale cluster composed of thousands of nodes, as for the writing of application programs on the basis of this framework, parallel processing of data set can be implemented securely and reliably, and the fault tolerance is high [12-15].

As a functional calculation programming framework, MapReduce imitates the Map and Reduce functions of LIST Processing. It simplifies the parallel computing, and finally, the process is divided into mapping stage and reduction stage. Map refers to mapping stage: a group of input data is replaced with key value pairs [16]; according to the set rules, a list of new key value pairs are mapped and used as the input data of Reduce. Reduce refers to the reduction stage; according to the shared key groups, all the mapping is sorted, reduced and simplified; the final results are saved in Hadoop file system.

Cluster configuration of MapReduce The command "sudo vi/etc/profile" was executed, and the following information was added to the end:

```
export JAVA_HOME=/usr/hadoop/jdk1.7.0_25

export JRE_HOME=$JAVA_HOME/jre

export PATH=$JAVA_HOME/bin:$PATH
```

By executing java-version, it could be observed whether the configuration was successful.

Network configuration

In the network configuration files, static Internet protocol (IP), gateway and domain name server (DNS) were set.

The command "sudo vi /etc/network/interfaces" was implemented and the following content was added:

auto eth1
iface eth1 inet static
address 192.168.1.24
gateway 192.168.1.1
netmask 255.255.255.0
dns-nameservers 210.47.208.8

The command "sudo ifup eth1" was implemented; after restarting the network, the command "ifconfig" was implemented to check if the network configuration was correct. The command "sudo vi /etc/hostname" was implemented, so that the hostname was changed; for example, it could be renamed as "hadoop01". However, ubuntu of the original name had to be deleted; instead, only the set name remained.

Installation and configuration of Hadoop

Hadoop-1.1.2 was installed; Hadoop environment variables were added; "sudo vi /etc/profile" was implemented; the following content was added:

```
export HADOOP_HOME=/usr/hadoop/hadoop-1.1.2
export PATH=$PATH:$HADOOP_HOME
export PATH=$PATH:$HADOOP_HOME/bin
```

The file "hadoop-env.sh" was modified; the command "sudo vi /usr/hadoop/hadoop-1.1.2/conf / hadoop-env.sh" was implemented; the installation path of jdk was altered- export

```
JAVA\_HOME=
/usr/hadoop/j jdk1.7.0\_25; HADOOP_HOME
```

was added:

```
export
HADOOP_HOME=/usr/hadoop/hadoop-1.1.2/ export PATH=$PATH:$HADOOP_HOME/bin.
```

The configuration file "core-site.xml" was modified. The following command was implemented:

```
sudo vi /usr/hadoop/hadoop-1.1.2/conf / core-site.xml.
<configuration>
<property>
<name>hadoop.tmp.dir</name>
<value>/tmp/fs</value> /* the folder of hadoop file system was set */
</property>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
/* applied port number clamp of Namenode in HDFS*/
</property>
</configuration>
```

The file "hdfs-site.xml" was modified.

The following command was implemented:

```
sudo vi /usr/hadoop/hadoop-1.1.2/conf / hdfs-site.xml.
<configuration>
<property>
<name>dfs.replication</name>
<value>4</value> /* duplicate count of data block saved in HDFS */
</property>
</configuration>
```

The file "hdfs-site.xml" was modified.

The following command was implemented:

```
sudo vi /usr/hadoop/hadoop-1.1.2/conf / hdfs-site.xml.
<configuration>
<property>
<name>dfs.replication</name>
<value>4</value>
/* duplicate count of data block saved in HDFS */
</property>
</configuration>
```

The configuration file was modified: mapred-site.xml.

```
<configuration>
<property>
<name>mapred.job.tracker</name>
<value>localhost:9001</value> /* port number of JobTracker was set*/
</property>
</configuration>
```

The configuration file "masters" was modified; the name of the main node was set as "hadoop01".

The configuration file "slaves" was modified; list of the node names was set: hadoop01, hadoop02 ...

Test of Hadoop cluster

First, HDFS was formatted on Namenode; "hadoop Namenode format" was implemented; then, all the daemons were started, and "start - all.sh" was implemented. The starting status of the process was checked; "jsp" was implemented.

3. Experiment on Hadoop cloud computing platform

3.1 Experiment environment

The experiment was performed with Hadoop as the cloud computing platform. The machine cluster on the platform consisted of seven machines, one of which was the node and the others were data node machines. Within the Hadoop cluster, machine hardware configuration was: Intel (R) Core (TM) i3- 3240 CPU 3.40 GHz, 8.00 GB of internal memory, 512 GB of hard drive; software environment was Linux operating system, JDK 1.6, Hadoop 0.20.1.

3.2 Experiment results and analysis

The purposes of the experiment included: validation of the feasibility of neural network algorithm based on cloud platform, test of the speed of the algorithm and test of training precision of the algorithm. The experiment involved two data sets Breast Cancer Wisconsin Data Set and KDD Cup 1999 data set. The former was used to test the classification effect. In the data set, there were 699

sample data. Removing the 16 data whose record was missing, the first 450 data were used as the data of training samples, while the other 233 data were used as test data. There were 11 fields in each record. Data field 1 was used to record id; data fields 2 D were data input mode; data field 11 was used for classification of the records. Each record included two types: negative and positive. KDD Cup 1999 data set was used to detect abnormal network intrusion; each sample of the file contained 42 attribute values.

In the study, the training samples were arranged to multiple Mappers for training. In the experimental driving function, accuracy test on network classification was performed. Each experiment item was operated for eight times, and their average values were obtained. The results can be seen in Table 1 and Fig. 1.

	Amount of rounds	Error	Classification accuracy	Count of mappers
Serialback propagation(BP)	2	0.1701	0.7645	
Serial BP	8	0.1251	0.7813	
Serial BP	32	0.1302	0.9411	
MRBP	2	0.035	0.9926	3
MRBP	8	0.087	0.9803	3
MRBP	32	0.146	0.9827	3

Figure 1: Experimental results of multi-os remote booting protocol (MRBP)

Table 1 Experimental results of multi-os remote booting protocol (MRBP)

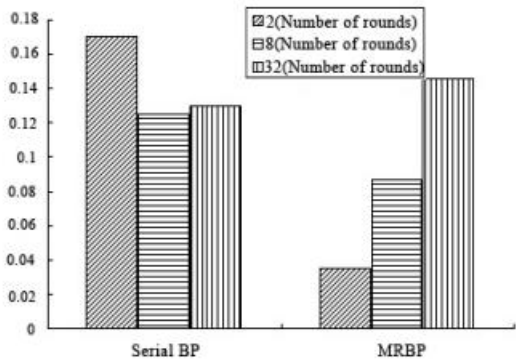

Figure 2: Comparison of errors of serial BP and MRBP

As shown in Table 1 and Figure 1, after two rounds of training on the training samples with MRBP, the effect was satisfactory and better than that of serial BP. The reason why MRBP was more effective was that the training samples were segmented properly and assigned to each Mapper; a reduction in sample size could accelerate the convergence rate on each Mapper, then the

entire training process was accelerated. However, as the segmented samples failed to cover the total samples effectively, multiple training was required.

Table 2 shows the running time of serial BP and MRBP in training samples when the amounts of samples were respectively: 240, 000; 600, 000; 1, 000, 000.

Sample amount	240,000	600, 000	1, 000, 000
Running time of serial BP (s)	2459.1	4013.8	12351.9
Running time of MRBP (s)	2563.9	3601.5	7213.4

Figure 3: Running time of serial BP and MRBP

As shown in Table 2, when sample amount was 240,000, the parallel processing speed of serial BP was faster than that of data-parallel MRBP pattern classification algorithm, which was due to the influence of factors such as system overhead of Hadoop [17,18]. However, as sample amount increased to more than 600,000, there was parallel efficiency; and as the data size continued to increase, the parallel processing speed became faster and faster, and the speed-up ratio started to increase gradually, which was because parallelization and Hadoop distributed file storage could reduce data network transmission overhead and memory overhead. As the samples amounted to one million, the growth trend of speed-up ratio was more obvious, proving that on Hadoop platform, parallel MRBP combination classification algorithm was effective, feasible and superior in processing large-scale data.

Suppose Ta as the running time of Hadoop cluster and Tb as the running time of uniprocessor, then Tb/Ta could be defined as the speed-up ratio which was an important measuring standard of the experiment. Separately, with two kinds of data sets, six experiments were performed, and the results can be seen in Table 3.

Testing dataset	Number of cluster machines	T_a (s)	T_b (s)	T_b / T_a
KDD Cup 1999	2	3926	7264	1.85
	4	1859		3.91
	6	1628		4.46
Breast Cancer Wisconsin (Original) Data Set	2	4126	8016	1.94
	4	2413		3.32
	6	2203		3.64

Figure 4: Performance contrast of uniprocessor and Hadoop cluster

According to Table 3, with the use of these datasets, while the amount of cluster machines increased, speed-up ratio increased correspondingly. When

there were two machines in the cluster, the speed-up ratios were respectively up to 1.85 and 1.94, which were close to parallel amount 2; when the amount of cluster machines was 4, the speed-up ratios respectively raised to 3.91 and 3.32; when the amount of cluster machines increased to 6, the increase in speed-up ratios was not significant, indicating that a greater amount of machines did not necessarily mean the speed-up ratio would be higher. The more the machines in the cluster, the greater the overhead of communication between nodes [19]; if the processing data size did not increase correspondingly, the loss would outweigh the gain.

Therefore, it was suggested that a proper coherence point between machine amount and the processing data size should be determined where the speed-up ratio was the closest to the amount of cluster machines, so that cloud computing platform could be used the most effectively, which could be achieved by setting the parameters of MapReduce tasks and performing multiple experiments. In the experiment, the optimal condition was that there were six machines in the cluster when the running time of Hadoop cluster was respectively 4.46 and 3.64 times that of uniprocessor; still, there was a certain gap between the speed-up ratios and the parallel amount, which indicated that the parallel process of the algorithm stilled needed to be improved. To enhance the efficiency of parallel computing system, the applications should be optimized and their concurrency should be enhanced.

The number of cluster machines was increased to 8. $T_a(s)$ of KDD Cup 1999 was 1608; T_b/T_a was 4.84. $T_a(s)$ of Breast Cancer Wisconsin (Original) Data Set was 2147; T_b/T_a was 3.39. Compared to the situation when there were 6 cluster machines, the speed increase of $T_a(s)$ was not obvious, and the value of T_b/T_a was significantly different with the parallel number. Therefore, it could be concluded that, 6 cluster machines was the most suitable.

4. Discussion

BP neural network, a kind of multi-layer forward artificial neural network model simulating the learning and memory processes of the brains of creatures, has been extensively applied in many fields. With the increase of the accumulated data in various fields, the problems faced by BP neural network in practical application become more and more complicated. Ren C. et al. [20] predicted the optimal wind speed based on the optimal parameter selection of the particle swarm optimized BP neural network in wind power generation. In the medical field, Samanta S. et al. [21] distinguished normal retina from the retina which was affected by glaucoma using Haralick function, processed the extracted features with BP neural network, and successfully realized the classification of eyes with glaucoma, with an accuracy of 90%. Based on the previous studies on BP neural network, this study parallelized multi-layer neural network based on cloud computing and compared it with the serial BP neural network in the perspectives of the speed and preciseness in processing data set. The results

demonstrated that, the preciseness of MRBP was superior to that of the serial BP; the processing speed of the serial BP was larger than that of MRBP when the size of data set was smaller than 0.25 million; but the processing speed of MRBP was higher when the size of data set was larger than 0.6 million, and the speed was higher if data size became larger. Thus the advantage of the cloud computing based multi-layer neural network was that it could decompose the large-scale input data and effectively shorten operation time on the premise of ensuring the accuracy of the algorithm.

5. Conclusion

In this study, multi-layer neural network was parallelized based on cloud computing platform, and it achieved favorable efffect in procesing large-scale data, which performed well in time and quality. But due to the limited experimental conditions and resources, there were many defects in the experiments and analysis, which remain to be corrected and perfected in the future studies.

Acknowledgments

This study was supported by The Design and Implementation of Examination Verification System Based On Face Recognition, the Key Project of 2014 Hunan Radio & TV University (No.XDK2014-A- 7).

References

[1] R.N. Calheiros, R. Ranjan, A. Beloglazov et al., *CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms*, Software Practice & Experience, 41(2011), 23-50.

[2] S. Abiteboul, *Object database support for digital libraries*, European Conference on Research & Advanced Technology for Digital Libraries, Springer Berlin Heidelberg, 2010, 11-23.

[3] X. Tao, Y. Li, J. Zhang et al., *Mapping semantic knowledge for unsupervised text categorisation*, Twenty-Fourth Australasian Database Conference, Australian Computer Society, Inc., 2013, 51-60.

[4] I. Aizenberg, A. Luchetta, S. Manetti, *A modified learning algorithm for the multilayer neural network with multi-valued neurons based on the complex QR decomposition*, Soft Computing, 16(2012), 563-575.

[5] J.C. Nunez-Perez, J.R. Cardenas-Valdez, J.A.G. Aguilar et al., *Measure-based modeling and FPGA implementation of RF Power Amplifier using a multi-layer perceptron neural network*, International Conference on Electronics, Communications and Computers, 2014, 237-242.

[6] M. Krawczak, S. Sotirov, E. Sotirova, *Generalized Net Model for Parallel Optimization of Multilayer Neural Network with Time Limit*, Intelligent Systems, International IEEE Conference, 2012, 173- 177.

[7] K. Gopalakrishnan, R.V. Uthariaraj, *Acknowledgment based Reputation Mechanism to mitigate the node misbehavior in mobile ad hoc networks*, Journal of Computer Science, 7(2011), 1157-1166.

[8] J. Chase, F. Amador, E. Lazowska et al., *The Amber system: parallel programming on a network of multiprocessors*, Sosp Proceedings of the Twelfth Acm Symposium on Operating Systems Principles, 23 (2010), 147-158.

[9] R.C. Taylor, *An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics*, Bmc Bioinformatics, 12 (2010), 3395-3407.

[10] Q. Li, T. Zhang, Y. Yu, *Using cloud computing to process intensive floating car data for urban traffic surveillance*, International Journal of Geographical Information Science, 25 (2011), 1303-1322.

[11] N. Dhingra, P. Jha, V.P. Sharma et al., *Adult and child malaria mortality in India: a nationally representative mortality survey*, Lancet, 376 (2010), 1768-74.

[12] S.Q. Li, S.X. Zhang, *A congeneric multi-sensor data fusion algorithm and its fault-tolerance*, International Conference on Computer Application and System Modeling, IEEE, 2010, 339-342.

[13] A. Martin, T. Knauth, S. Creutz et al., *Low-Overhead Fault Tolerance for High-Throughput Data Processing Systems*, Proceedings-International Conference on Distributed Computing Systems, 2011, 689-699.

[14] R. Alexandersson, J. Karlsson, *Fault injection-based assessment of aspect-oriented implementation of fault tolerance*, IEEE/IFIP International Conference on Dependable Systems & Networks. IEEE, 2011, 303-314.

[15] D. Seo, H. Lee, A. Perrig, *Secure and Efficient Capability-Based Power Management in the Smart Grid*, Ninth IEEE International Symposium on Parallel and Distributed Processing with Applications Workshops, IEEE, 2011, 119-126.

[16] D. Aguiar, S. Istrail, *HapCompass: a fast cycle basis algorithm for accurate haplotype assembly of sequence data*, Journal of Computational Biology A Journal of Computational Molecular Cell Biology, 19 (2012), 577-90.

[17] J.E. Gibson, R.L. Murray, R. Borland et al., *The impact of the United Kingdom's national smoking cessation strategy on quit attempts and use of cessation services*, Findings from the International Tobacco Control Four Country Survey, Nicotine & Tobacco Research, 12 (2010), 64-71.

[18] M. Diana, A. Gabriel, Bougé Luc, *Improving the Hadoop map/reduce framework to support concurrent appends through the BlobSeer BLOB management system*, ACM International Symposium on High Performance Distributed Computing, ACM, 2010, 834-840.

[19] M. Farmani, H. Moradi, M. Asadpour, *A hybrid localization approach in wireless sensor networks using a mobile beacon and inter-node communication*, IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, 2012, 269-274.

[20] C. Ren, N. An, J. Wang et al., *Optimal parameters selection for BP neural network based on particle swarm optimization: A case study of wind speed forecasting*, Knowledge-Based Systems, 56 (2014), 226-239.

[21] S. Samanta, S.S. Ahmed, A.M.M. Salem et al., *Haralick Features Based Automated Glaucoma Classification Using Back Propagation Neural Network*, International Conference on Frontiers of Intelligent Computing: Theory and Applications, 2014, 351-358.

Accepted: 21.03.2017