VOL. 12, NO. 21, NOVEMBER 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

GRID COMPUTING FOR DISTRIBUTED NEURAL NETWORKS: AN APPROACH

Abhishek Gudipalli[1] and M. Saravanan[2]
[1]School of Electrical Engineering, VIT University, Vellore, Tamil Nadu, India
[2]Department of Electronics and Instrumentation, SreeVidyanikethan Engineering College, Tirupati, India
E-Mail: abhishek.g@vit.ac.in

ABSTRACT

In this paper, we report the first evaluation of cooperation computing for artificial neural networks in distributed environment. Several performance-relevant factors are considered, including architecture of computing service, workflow and cooperation strategy. Evidence on basic processes and performance of such strategies of cooperation computing are reviewed. We also present a theoretical analysis of distributed-training strategies of neural networks for structure-distributed and data-distributed. We prove a strategy of distributed computing based on data-distributed is more feasibility for distributed neural networks, which makes training the neural networks more efficient. In the final, we concluded the evaluation by briefly considering selected open questions and emerging directions in construction of grid computing for distributed neural networks.

Keywords: grid computing, distributed neural networks, artificial neural networks, back propagation learning algorithm.

INTRODUCTION

Service system of computing is becoming increasingly important and ubiquitous in our lives - for organizations, financial institutions, professionals and individuals. It is emerging with the popularity of network in workstation and greatly accelerated with the development of inexpensive and powerful personal computers. It's blooming with the rapid deployment of the engineering applications and exploded with the unfolding of the web in the past five years [G. H. Forman., *et al* 1994]. While it's hard to make predictions, many expect the trend to quicken with continued advances in mobile computing, DNA computing, microelectronics and nanotechnology [Imry. Y., *et al* 1997]. Imagine a world with billions of people and agents who interact daily with billions of computational devices, each of which is consuming and producing services and communicating with scores of other devices on a constant and largely autonomous basis. This evolution provides many new challenges to our ability to design and deliver computing service systems. An important challenge of which is how to construct an efficient service system with the amount of distributed computational devices in the Internet, since many in the world of modern scientific calculations are relying on multiple, time-lapsed analyzed of a large amount of data.

In this paper, for simplicity, we will only discuss the performance of grid computing for distributed artificial neural networks (ANNs), although the same methodology could beadopted in the analysis of other cooperation computing Since the different strategies of cooperation will take different bias, the following chapters will analyze and evaluate the different strategies' performance from structure to computing efficiency.

ARCHITECTURE OF GRID COMPUTING BASED ON INTERNET

In this paper, architecture of grid computing is defined as shown inFigure-1. The structure partitions the computing service system into two sub services: service for the computing providers, and service for the computing consumers.

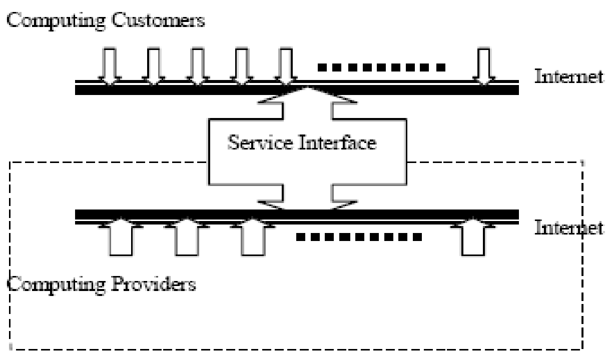

Figure-1. Web service of grid computing based on multi agents.

Each resource of grid computing provides local computing energy to cooperation system by the local agent, and become a member of the system. The agent control and manage the local resource. All of those constitute the computing providers. Agent of service manager is in charge of interface of the service between the providers and the consumers. It assigns a sub computing-grid for a mission and maintains the mission queue. For computing consumers, all they have to do is cast their problem to the service-manager in a form suitable for execution on terminal (Internet browser or submit terminal), and then waiting for the result come from the service system. As to the providers, how to organize the computing resources and distribute the computing mission for parallel performing is the key decision. Different strategy will take different efficiency for varying applications [G. H. Forman., *et al* 1994].

VOL. 12, NO. 21, NOVEMBER 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

REVIEWS ON TRAINING STRATEGIES OF DISTRIBUTED NEURAL NETWORK

Neural network is a computational model, which consists of many simple units working in parallel with no central control. The multilayer feed-forward networks can represent any function with enough units. And it is also an accumulated unit of knowledge that can get result directly from a trained one by responding to the input stimulation. Back Propagation (BP) learning algorithm is successfully in learning multilayer feedforward networks by gradient descent in weight space to minimize the output error [Lewin, D.I., 2002].As we all known, however, there is no guarantee that the global optimum is sure to be found, and its convergence speed is often very slow especially when the training data contains thousands and hundreds samples. Although the traditional neural network has the character of parallel computing in the neural units, the realization of learning algorithm such as BP algorithm is still serial. The process of learning carries out forward calculation and back forward error propagation on the layers one by one with the learning sample entrance. It is not suitable for the distributed environment; the memory bottleneck problem will be occurred when the learning sample set is very large, and the characteristic of parallel computation on the neural units is not fully reflected. Therefore, distributed-learning strategy for neural network is inducted to improve the performance of learning [Lewin, D.I., 2002].

To the best of our knowledge, there are three main kinds of distributed implementation for ANNs, high-coupling ANNs, low-coupling ANNs and data distributed ANNs. High-coupling

ANNs refer to those ANN classifiers that a neural network model is constructed by combining many sub-network-units. Many distributed structure-based versions of ANNs can be regarded as high-coupling ANNs, such as Hierarchical Neural Network, Hierarchical Radial Basis Function (HiRBF), and Distributed - Structure-Based Neural Networks (DSBNN) and so on. Among these, the Distributed - Structure-Based Neural Networks (DSBNN) is the representative of high-coupling ANNs [Ian Foster., 2000]. The content of communication among the distributed units is neuron response signal, as shown in Figure-2.

There is a decision module to incorporate different output of multi-modal into the final decision. The content of communication among the distributed units is data to each module and the output to decision module, however, there is a dilemma between the computing efficiency and the stability, when design the model of classifier.

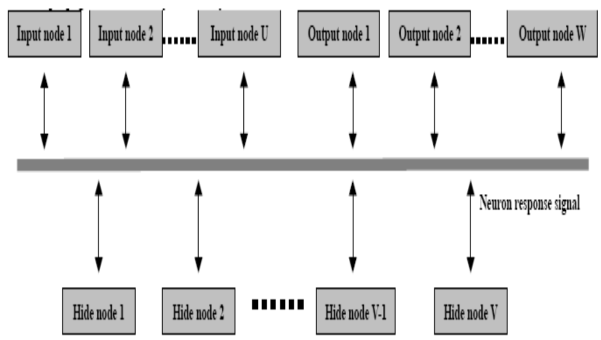

Figure-2. High-Coupling ANN's model.

For emphasizing particularly on parallel data processing, high-coupling ANNs model behave itself with good computing Efficiency. On the other hand, because of the high coupling among distributed united and relying on the initial weights of system the stability of high-coupling ANNs model is still far from satisfactory. Though low-coupling ANNs is robust to initial weights by training and incorporating redundant modules, it has no ability to improve the computing efficiency.

In order to avoid the limitations of structure based methods, in a distributed-learning strategy based on distributed data-chip (DLSBC) is proposed to balance the computing efficiency and the stability. It improves the convergent speed through making use of multi-computing-nodes with different dataset on network. Since the BP learning is relying on the initial weights, DLSBC trained more than one neural network in different computing node with different initial weights to improve the stability of model. It is a parallel climbing strategy to avoid local minimum. At the same time, it inducts evolutionary mechanism to optimize the neural network's weights, and exchange the knowledge among computing-nodes, which make DLSBC have the ability to learn a whole knowledge from local sample. All of these operations reduce the impact of the special initial weights that lead to fall into local minimum, and improved computing efficiency [A. Geist., *et al* 1994, O. M. Lucila., 1996]. The content of communication among the distributed units is transferring neural networks, as shown in Figure-3.

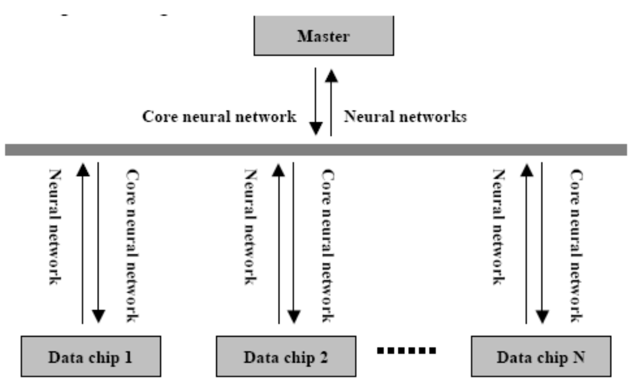

Figure-3. Cooperative system based on distributed data-chip.

VOL. 12, NO. 21, NOVEMBER 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

EVALUATION OF PERFORMANCE OF GRID COMPUTING

As mentioned above, performance of grid computing is the major decision of service system's feasibility, which can be partitioned into two sub performance evaluations, communication efficiency and computing efficiency. The costs of the service system rely on the communication efficiency and the computing efficiency. As we known, not all of us will perform a mission in a remote computing with 3 min computing efficiency and 3 min communication efficiency, and the local machine's efficiency is 5 min. In this paper, feasibility of service system is specified by rate of improved performance $Rate_{IP}$ as calculated by Eq. (1).

$$Rate_{IP} = \frac{Cost_{distributed}}{cost_{local}} \times 100\% \quad (1)$$

In our case, there are two general workload allocation methods are commonly influence on the communication efficiency and the computing efficiency. The one called data decomposition, assumes that the overall problem involves applying computational operations or transformations on one or more data structures and these data structures may be divided and operated upon, then identical tasks operate on different portions of the data. The other called function decomposition, divides the work based on different operations or functions, and fundamentally different tasks perform different operations. Assume that there is a training of feed-forward neural network can be divided into the decomposition as shown in Figure-4.

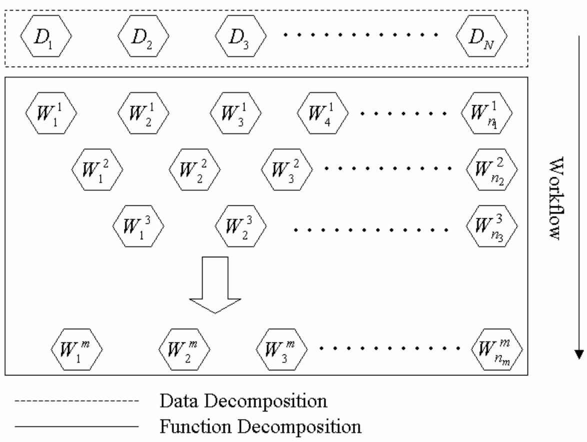

Figure-4. Computing mission decomposition.

The training deals with |S| samples of dataset S,

$$S = \coprod_{i=1}^{N} D_i \, , i = 1,2, \ldots, N. \quad (2)$$

$$|D_i| = |S| \times \frac{P_i}{\sum_{j=1}^{N} P_j} \quad (3)$$

where D_i is the data-chip, where |S| is the record number of sample set, P_i is the computing power of i_{th} cooperator. And there are N neurons

$$N = \sum_{i=1}^{m} n_i \; , n_i \geq 1, m \geq 1 \quad (4)$$

Where m is the layers of neural network, n iis the number of nodes on i_{th} layer. For simplicity, we assume that the number of computing-nodes is N in the distributed environment for discussion on the DSBNN. CMM denote the cost of communication between two computing nodes for processing a sample in an ideal speedy network, in which CMM could be regarded as the cost of constructing a connection. The cost of serial computing on a single computer (CAL_g) is calculated by Eq.(5).

$$CAL_g = |S| \times \sum_{i=1}^{m} \sum_{j=1}^{n_i} C_j^i \quad (5)$$

The cost of high-coupling ANNs computing on N computers is calculated by Eq.(6).

$$CAL_s = |s| \times \sum_{i=1}^{m} Max \left\{ C_j^i \, , \; j = 1,2, \ldots, n_i \right\} \quad (6)$$

Where m denotes the number of neural network's layers, n is the sum of nodes on the i_{th} layer, C_j^i is the cost of processing one sample on the jth node of ith layer. Those indicate that the cost of structure parallel computing depends on the number of layers, the maximum cost of the nodes in a layer and the scale of the sample set S. The communication efficiency $COMM_S$ is calculated by Eq. (7).

$$COMM_s = 2 \times |S| \times T_n \times T_c \sum_{i=2}^{m} CMM \quad (7)$$

Where coefficient '2' denotes the neuron response signals have feed-forward and backward propagations in BP method processing, $T_n \times T_c$ is the sum of learning iterations.

CONCLUSIONS

As the concept of ubiquitous and pervasive computing developing, the computing service system is emerging importance in the process. To utilize this method, its effectiveness would be considered when it's been constructed. A review on the performance of computing service system could contribute to the work, and provide a possible direction of the research in future. This paper analyzed and evaluated the performance of computing service system when it is used in the distributed neural networks. From the comparisons of the structure-distributed model with data-distributed model, we can conclude that the performance of distributed computing is rely on those factors and depends on the problem, which has better character on data decomposition fit to the data parallel structure, or else the structure parallel may be considered. Moreover, for those more complex problems, the hybrid method may be suitable. Evidently, the grid computing is more complex when it is applied in different cases. This paper just takes a brief review, and the more details in depth would be considerate in the future.

REFERENCES

G. H. Forman and J. Zahorjan. 1994. The challenges of mobile computing. IEEE Comp. 27(4): 38-47.

Lewin D.I. 2002. DNA computing. IEEE computational Science and Engineering. 4(3): 5-8.

Imry Y. 1997. Introduction to Mesoscopic Physics, Mesoscopic Physics and Nanotechnology, Oxford University, New York.

Ian Foster. 2000. Internet Computing and the emerging Grid. Nature. 408(6815).

G. Shao, R. Wolski and F. Berman. 1999. Performance effects of scheduling strategies for master/slave distributed applications. in Proc. PDPTA'99, CSREA, Sunnyvale, Calif.

A. Geist, A. Beguelin, J. Dongarra, J. Weicheng, R. Manchek and V. Sunderam. 1994. PVM: Parallel Virtual Machine. Cambridge, MA: MIT Press.

O. M. Lucila. 1996. Medical applications of artificial neural networks: connectionist model of survival. Ph.D. Thesis, Stanford University.