Research on the performance of grid computing for distributed neural networks

Yang Bo, † *Wang Xun*

School of Computer Science and Technology, Harbin Institute of Technology, P.O.Box 319, 150001 Harbin, China

Summary
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.

Key words:
Artificial neural network, Grid computing, Cooperation strategy, Training algorithm, Performance evaluation

1. 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. While it's hard to make predictions, many expect the trend to quicken with continued advances in mobile computing [1], DNA computing [2], microelectronics and nanotechnology [3]. 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.

The computational bottleneck comes from the rapid increasing data set, however, there are more than 400 million PCs around the world, many of them are as powerful as the supercomputer of 1990s, and most are idle much of the time. So it's the reason of distributed computing with cooperation (grid computing) emerging [4].

In order to utilize the grid computing, applications must evolve through the distributed-memory parallel version of the application algorithms. The actual decisions fall into two major categories - those related to structure, and those related to efficiency.

For structural decisions in applications, the major decisions to be made include the choice of cooperation computing models to be used - Master-Slave computation [5] vs. Node-Node computation and data decomposition vs. function decomposition [6]. Decisions with respect to efficiency when computing service system for Internet environments are generally oriented toward minimizing the frequency and volume of communications. In the latter that the distributed computing environments based on networks, large granularity generally leads to better performance. But it is in a dilemma between the computing efficiency and the communication efficiency.

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 be adopted 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.

Manuscript revised January 2006.

2 Architecture of Grid Computing Based on Internet

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

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.

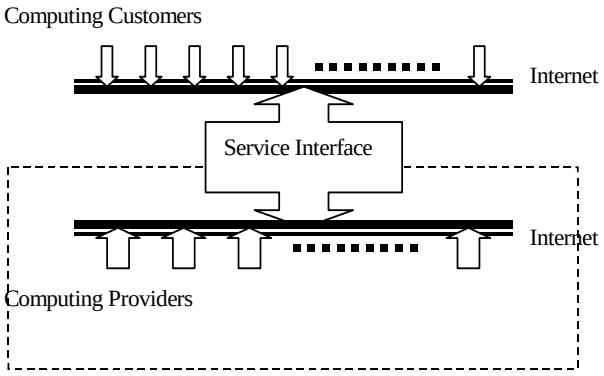

Fig. 1. Web service of grid computing based on multi agents

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. Workflow of the system can be described as Fig. 2.

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.

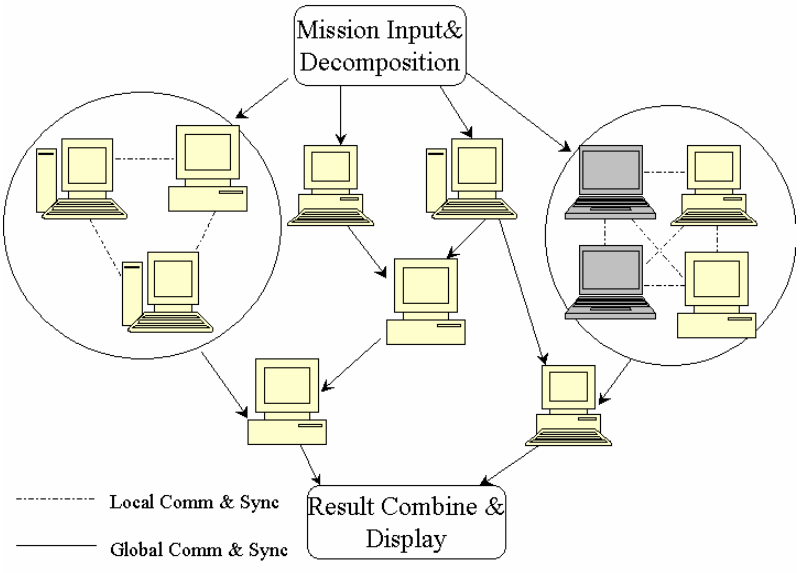

Fig 2. Computing Service Workflow

3. 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 feed-forward networks by gradient descent in weight space to minimize the output error. 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.

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 [7], Hierarchical Radial Basis Function (HiRBF)[8], Distributed-Structure-Based Neural Networks (DSBNN)[9] and so on. Among these, the Distributed-Structure-Based Neural Networks (DSBNN) is the representative of high-coupling ANNs. The content of communication among the distributed units is neuron response signal, as shown in Fig. 3.

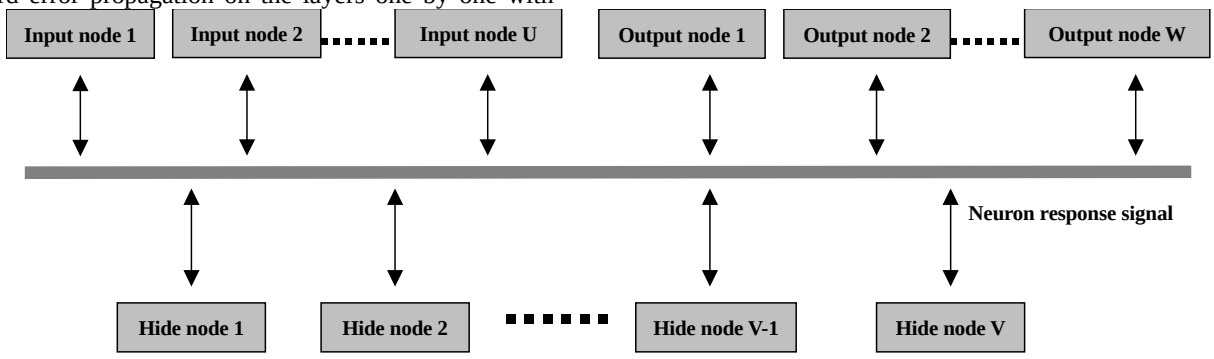

Fig. 3. High-coupling ANNs model.

Low-coupling ANNs refer to those methods that training many NNs model at the same time. That is, each neural network modules learn the same data using different initial weights. Many multi-module versions of ANNs can be regarded as low-coupling ANNs [10,11,12]. In which there are 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, as shown in Fig. 4.

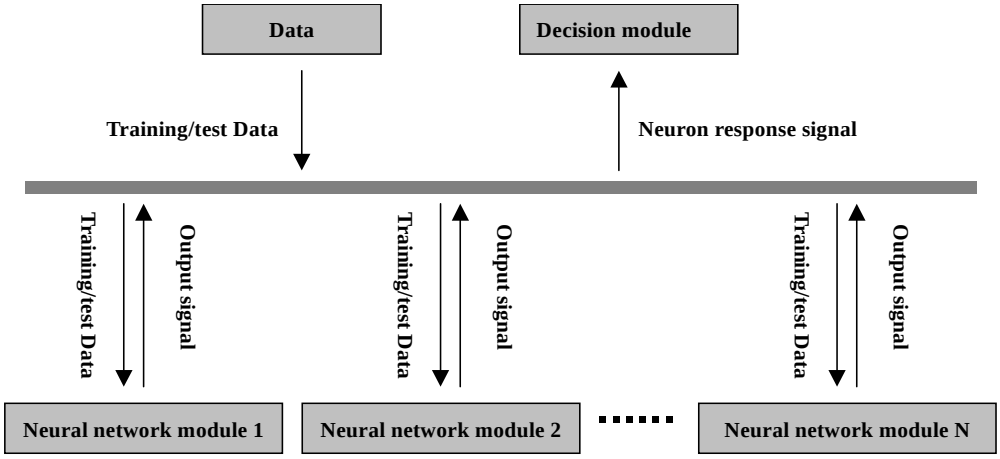

Fig. 4. Low-coupling ANNs model.

However, there is a dilemma between the computing efficiency and the stability, when design the model of classifier. 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 [13] 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. The content of communication among the distributed units is transferring neural networks, as shown in Fig. 5.

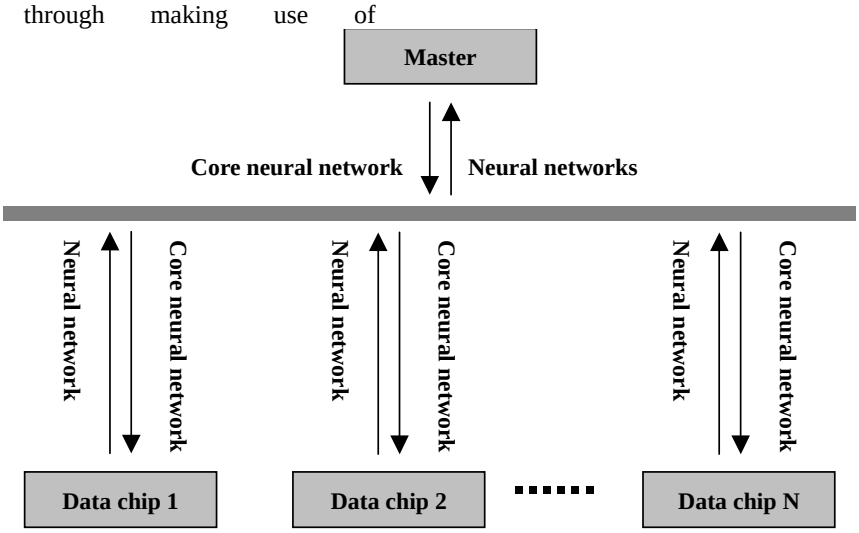

Fig. 5. Cooperative system based on distributed data-chip.

4. 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 Fig. 6.

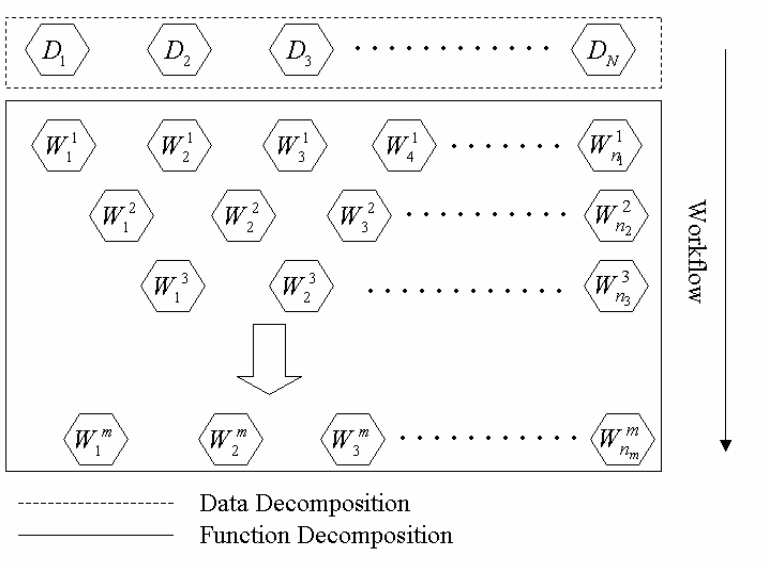

Fig 6. Computing Mission Decomposition

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

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

where D_i is the data-chip,

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

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_i is 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).

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

where C_j^i is the cost of processing one sample on the j^{th} node of i^{th} layer. |***S***| is the number of the whole sample set.

The communication efficiency $COMM_g$ is null, because there is no communication of computing works on a single computer. It represents the performance of traditional method without distributed-learning strategy.

4.1 Performance evaluation of learning strategy on distributed-structure-based neural networks (DSBNN)

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

$$CAL_s = |S| \times \sum_{i=1}^{m} Max\{C_j^i, j = 1, 2, ..., n_i\} \quad (6)$$

where ***m*** denotes the number of neural network's layers, n_i is the sum of nodes on the i^{th} layer, C_j^i is the cost of processing one sample on the j^{th} node of i^{th} 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.

4.2 Performance Evaluation of Learning strategy on Multi-Modal Neural Networks (MNNs)

The cost of low-coupling ANNs computing on ***N*** computing nodes is calculated by Eq.(8).

$$CAL_M = Max\{CAL_g^i, i = 1, 2, ..., N\} \quad (8)$$

where the cost of computing on i^{th} module is CAL_g^i. For the modules parallel processing, CAL_M relies on the maximal CAL_g^i.

The communication efficiency $COMM_M$ is calculated by Eq.(9).

$$COMM_M = 2 \times N \times CMM \quad (9)$$

where the coefficient '2' denotes the cost of data transfer is distributing sample set to ***N*** nodes when the data initializing and the result collection.

4.3 Performance evaluation of learning strategy on distributed data-chips neural networks (DDBNN)

The cost of chips computing on ***N*** computers is calculated by Eq.(10).

$$CAL_C = Max\{|D_k| \times \sum_{i=1}^{m} \sum_{j=1}^{n_i} C_j^i, k = 1, 2, ..., N\} \quad (10)$$

where $|D_k|$denotes the records of the k^{th} data-chip, n_i is the number of the i^{th} layer's nodes, C_j^i is the cost of processing one sample on the j^{th} node of i^{th} layer. Those indicate that the cost of chips computing depends on $|D_k|$ and the maximum cost on a computing node among those cooperators.

The communication efficiency $COMM_C$ is calculated by Eq.(11).

$$COMM_C = 2 \times N \times T_n \times CMM \quad (11)$$

where the coefficient '2' denotes the distribution of core neural network and result collection of each learning-chip, ***N*** is the number of cooperators, T_n is the number of learning-chips.

5. Performance Comparisons of Different Learning Strategies

Assume that those computing nodes have the same performance. These nodes have the same cost C_s of processing a sample.

For this case, we can refine (6), (8) to (12), (13).

$$\begin{aligned} CAL_S &= |S| \times \sum_{i=1}^{m} Max\{C_j^i,\ j = 1,2,...,n_i\} \\ &= |S| \times \sum_{i=1}^{m} C_s \\ &= |S| \times m \times C_s \end{aligned} \tag{12}$$

$$\begin{aligned} CAL_M &= Max\{CAL_g^i, i = 1,2,...,N\} \\ &= CAL_g \\ &= |S| \times \sum_{i=1}^{m} \sum_{j=1}^{n_i} C_s \\ &= |S| \times N \times C_s \end{aligned} \tag{13}$$

Because

$$m \le N \tag{14}$$

We can get the relationship Eq.(15) between CAL_S and CAL_M

$$CAL_S \le CAL_M \tag{15}$$

Considering that the data-chip is same with each other according to Eq.(3) when every computing node has the same performance P_S, we have

$$\begin{aligned} CAL_C &= Max\{|D_k| \sum_{i=1}^{m} \sum_{j=1}^{n_i} C_j^i, k = 1,2,...,N\} \\ &= |S| \frac{P_s}{\sum_{j=1}^{N} P_s} \sum_{i=1}^{m} \sum_{j=1}^{n_i} C_s \\ &= |S| \times C_s \end{aligned} \tag{16}$$

Then we can get the relationship Eq.(17) between CAL_C and CAL_S

$$CAL_C \le CAL_S \tag{17}$$

So we can derive Eq.(18) if each computing node has the same performance.

$$CAL_C \le CAL_S \le CAL_M \tag{18}$$

When the computing nodes have the different performance, we assume that C_w is cost of the computing node that have worst performance in the cooperators, and it is the maximum cost among those nodes when process a sample. Thus, we have the following conclusions:

The computing performance of DSBNN is easy to derive that

$$\begin{aligned} &|S| \times C_w \le CAL_S \le |S| \times \sum_{i=1}^{m} C_w \\ \Rightarrow &|S| \times C_w \le CAL_S \le |S| \times m \times C_w \end{aligned} \tag{19}$$

The computing performance of MNNs is easy to derive that

$$\begin{aligned} CAL_M &= Max\{CAL_P^i \mid i \in [1, N]\} \\ &= CAL_P^w \\ &= |S| \times \sum_{i=1}^{m} \sum_{j=1}^{n_i} C_w \\ \Rightarrow CAL_M &\ge |S| \times m \times C_w \end{aligned} \tag{20}$$

Assume that the size of each data-chip is same to the others, we have

$$|D_s| = \frac{|S|}{N} \tag{21}$$

Computing performance of DDBNN is easy to derive that

$$\begin{aligned} CAL_C &\le \frac{|S|}{N} \times \sum_{i=1}^{m} \sum_{j=1}^{n_i} C_w \\ \Rightarrow CAL_C &\le |S| \times C_w \end{aligned} \tag{22}$$

So we can get the relationship in Eq.(23)

$$CAL_C \le CAL_S \le CAL_M \tag{23}$$

According to (7), (9) and (11), we can derive the relationship (24) about communication efficiency when $N \le |S|$

$$COMM_M \le COMM_C \le COMM_S \quad (24)$$

Thus, overall we have the conclusion about the rate of improved performance of those distributed-learning strategies in Eq.(25)

$$\begin{aligned} Rate_{IP}^{C} &= \frac{CAL_C + COMM_C}{CAL_g} \times 100\% \\ &\le Rate_{IP}^{S} = \frac{CAL_S + COMM_S}{CAL_g} \times 100\% \end{aligned} \quad (25)$$

and

$$Rate_{IP}^{M} = \frac{CAL_M + COMM_M}{CAL_g} \times 100\% \quad (26)$$

The rank of $Rate_{IP}^{M}$ in those strategies relies on the performance of communication network.

From the comparisons mentioned above, different distributed learning strategy takes different computing efficiency. MNNs has the same computing efficiency to the traditional computing on single computer. The computing efficiency of MNNs is the worst among the three distributed learning strategies. But the multi-modules can train the neural networks from different initial state, which enhance the ability to approach the global optimum. And it has the lowest cost of communication among the distributed-learning strategies. DSBNN is proposed for a long time, and it is researched widely [9, 14]. It has the better performance of computing efficiency than the MNNs', but the communication efficiency is worse than MNNs'. Moreover, the stability of DSBNN is far from satisfactory, because the single initial state always make the learning fall into local minimum state. The DDBNN resulting from DLSBC in this paper has the best performance of computing efficiency than others. It improves the computing efficiency by using local small data-chip, which reduces the search space. The strategy of ANNs' transferring makes the local training to acquire the knowledge of the whole samples. And its cost of communication is acceptable than the DSBNN. The rates of improved performance for those distributed-learning strategies demonstrate that distributed-learning strategy based on data-chips (CLSBC) is more feasible than others, when the performance of communication network is not too bad.

6. Conclusion

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.

Acknowledgment

The authors would like to express their thanks to Prof. Su Xiao-hong and Prof. Wang Ya-dong for their valuable advice.

References

[1] G. H. Forman and J. Zahorjan. "The challenges of mobile computing". IEEE Comp., 27(4):38–47, Apr 1994.
[2] Lewin, D.I., "DNA computing". Computing in Science & Engineering [see also IEEE Computational Science and Engineering], Volume: 4 , Issue: 3 , May-June 2002, pp:5 - 8
[3] Imry, Y., Introduction to Mesoscopic Physics, Mesoscopic Physics and Nanotechnology, Oxford University, New York. 1997.
[4] Ian Foster. "Internet Computing and the emerging Grid". Nature, vol 408 issue 6815, 2000.
[5] G. Shao, R. Wolski, and F. Berman, "Performance effects of scheduling strategies for master/slave distributed applications", in: Proc. PDPTA'99, CSREA, Sunnyvale, Calif. 1999.
[6] A. Geist, A. Beguelin, J. Dongarra, J. Weicheng, R. Manchek, and V.Sunderam, "PVM: Parallel Virtual Machine". Cambridge, MA: MIT Press, 1994.
[7] O. M. Lucila, "Medical applications of artificial neural networks: connectionist model of survival," Ph.D. Thesis, Stanford University, 1996.
[8] N. A. Mat Isa, M. Y. Mashor, and N. H. Othman, "Diagnosis of Cervical Cancer using Hierarchical Radial Basis Function (HiRBF) Network," Proc. of Int. Conf. on Artificial Intelligence in Engineering and Technology, Kota Kinabalu, Sabah, Malaysia, Jun. 2002, pp.458-463.
[9] C. Milea, P. Svasta, "Using distributed neural networks in automated optical inspection," Concurrent Engineering in Electronic Packaging, 24rd Int. spring seminar on electronics technology, Calimanesti-Caciulata, Romania, May. 2001, pp.286-288.

[10] H. Cardot, M. Revenu, B. Victorri, M. J. Revillet, “A Static Signature Verification System Based on a Cooperating Neural Networks Architecture,” International Journal of Pattern Recognition and Artificial Intelligence, 8,1994:679--692.
[11] M. N. Ahmed, and A. A. Farag,, “Two-state neural network for volume segmentation of medical images,” Pattern Recognition Letters, 18,1997:1143-1151.
[12] Yoshihara, I., Kamimai, Y., and Yasunaga, M. (2001). “Feature Extraction from Genome Sequence using Multi-Model Neural Network”. Genome Informatics, 12, 420-422.
[13] Bo Yang, Ya-dong Wang, Xiao-hong Su: Research and Design of Distributed Neural Networks with Chip Training Algorithm. Lecture Notes in Computer Science 3610: 213-216, Springer Verlag, 2005.
[14] S. J. Russell and P. Norvig, {¥it Artificial Intelligence: A Modern Approach}, Englewood Cliffs, NJ: Prentice Hall, 1995.

Yang Bo received the B.S. degree from Guizhou Univ. in 2001. He received the Ph.D. degree from Harbin Institute of Technology in 2006. His research interest includes artificial neural networks, multi-agents system, genetic algorithm, pattern recognition, and their application to bioinformatics.

Wang Xun received the B.S. degree from Guizhou Normal Univ. in 2001. She is with Information Center, Organization Department of Guizhou since 2001. Her research interest includes artificial neural networks, statistics, and their application to data mining.