3 Conclusion

The parallel distributed processing feature of ANN is explored in the mapping process, and more accurate definitions of Load Balance and Communication Overhead are presented. Absorb Algorithm gets rid of the back modification of the existant algorithms, and improve the quality of mapping result and the speed of mapping. Absorb Algorithm finds a new way to resolve the mapping problem in the ANN virtual implementation field. It can map all kinds of neural network models. All kinds of experiments show that Absorb Algorithm is a real time algorithm.

References

1 Fogelman-souli F, Herault J. Neurocomputing: Algorithm, Architectures and Applications. Springer-Verlay Berlin, 1990

2 Ghosh J, Hwang K. Mapping neural networks onto message-passing multicomputers. J Parallel Distributed Computing, 1989, 6(2): 291～330

3 Kohonen T. Introduction to neural computing. Neural Network, 1988, 10(1): 3～16

4 Kung S y, Hwang J N. Parallel architectures for ANNs. IEEE Int. Conf. NNs, ICNN'88, 1988, 2: 165～172

5 Kung S Y, Hwang J N. A unified systolic architecture for ANNs. J. Parallel Distributed Computing, 1989, 6 : 357～387

6 Wasserman P. Neural Computing, Theory and Practice. Van Nostrand Reinhold, New York, 1989

7 Hwang J N. Algorithms/Application/Architecture of ANNs: [Ph. D. Dissertation]. Dept. of EE, Uni. of Southern California, 1988

8 Schwartz(ED) E. Computational Neuroscience. MIT press, 1988

一种优化神经网络映射算法——吸收算法

王意洁　胡守仁

(国防科技大学计算机系　长沙　410073)

摘　要　本文揭示了神经计算的本质——并行分布处理，并以此为基础提出了时间步的概念。分析了映射算法的两个重要概念——负载均衡和通讯开销，并提出了映射分配准则。在神经网络的映射分配中引入图论的有关思想，提出了一种优化的神经网络映射算法——吸收算法。最后给出了重要的试验结果，这些数据表明吸收算法是一种有效的映射算法。

关键词　神经网络，映射算法，负载均衡，通讯开销，时间步

分类号　TP301　　(责任编辑　张　静)

国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第18卷第3期 1996年9月 Vol. 18 No. 3

An Optimal Neural Network Mapping Algorithm —— Absorb Algorithm*

Wang Yijie Hu Shouren

(Department of Computer Science, NUDT, Changsha, 410073)

Abstract In this paper the important feature of artificial neural networks —— parallel distributed processing is discussed thoroughly, so the essential features of neural computing are revealed, and the important conception of timestep is introduced. At the same time, two critical concepts of mapping algorithm —— balancing the working load of PEs and the communicatin overhead among PEs are explained more thoroughly, and the main allocation criterions are presented. Adopting the ideas of graph theory, an optimal neural network mapping algorithm —— Absorb Algorithm is advanced. This mapping algorithm shows remarkable efficiency. Some interesting experimental results are presented.

Key words neural network, mapping algorithm, load balance, communication overhead, timestep

The virtual implementation of neural network is an important part of artificial neural network research. A neural network model consisting of N neurons is simulated by P processing elements (PEs), if $N>P$, this is called virtual implementation. At present, the general purpose neurocomputer is an important research subject of the neural network virtual implementation. In order to explore the important feature of ANN, we developed a general purpose parallel neurocomputer —— Neuro_I. In this neurocomputer, several neurons are placed onto a PE, how to map the neural network model onto PEs will affect the parallel processing capacity of Neuro_I directly. Therefore, the neural network mapping algorithm is an important research subject in the neural network virtual implementation field. We present an optimal neural network mapping algorithm —— Absorb Algorithm, which is applied in Neuro_I.

* 本文受863计划，国家自然科学基金资助。
1995年12月25日收稿

In the following, firstly we present the theory of neural network and the neural network mapping criterions. Then we discuss the main idea of the Absorb Algorithm and discribe it in detail. Finally we present and discuss some experimental results.

1 The Theory of Neural Network and Neural Network Mapping Criterions

The design of neural network mapping algorithm is based on the theory of neural network and the neural network mapping criterions.

1.1 The Computing Model of Neural Network

From our research work, we realize that the computing model of neural network is an information flow driven model. Neural computing combines parallel computation with serial computation, and has evident periodicity. The parallel computation in neural computing means that there are many computations of neurons (the computations inside the neurons and the computations of the connections) in one timestep, and a computing period consists of several timesteps. The serial computation in neural computing means that the computations of some neurons are interrelated with those of some other neurons, so they are processed in serial mode. According to the neural computing model and the topology structure of neural network, we can analyse a period of information flow driven process, and mark the cmputations of neurons which can be processed parallelly in one timestep.

The following conclusions about neural computing are based on the studies of all kinds of neural network models.

Conclusion 1 The computations with the same time mark can be processed parallelly.

Conclusion 2 Two different computations of a neuron don't exist in one timestep of one period.

Conclusion 3 The computations with small time mark should be accomplished before the computations with big time mark.

Conclusion 4 The neuron with time mark t_{i+1} is interrelated with at least one neuron whose time mark is t_i

1.2 The Basic Concepts

The following concepts are very important in the neural network mapping algorithm.

Definition 1 Load Balance

In a system, balancing the working load of PEs means that the working load of PEs is balanced in every timestep. Load Balance of a system can be measured as the following formula:

$$Balance\_sys = \frac{1}{steps} \sum_{i=1}^{steps} Balance_i$$

Balance_sys: the Load Balance factor of a system;

steps: the number of timesteps in a neural computing period;

Balance$_{t}$: the Load Balance factor of timestep$_i$.

Definition 2

The load of PE$_j$ in timestep t_i is the weighted sum of the amount of neural computation and the amount of related information with time mark t_i on PE$_j$.

$Load_j^{t_i} = \alpha\ Comp_j^{t_i} + \beta \cdot Info_j^{t_i}$

$Load_j^{t_i}$: the load of PE$_j$ in timestep t_i;

$Comp_j^{t_i}$: the amount of neural computation with time mark t_i on PE$_j$;

$Info_j^{t_i}$: the amount of related information with time mark t_i on PE$_j$;

α, β: the weighted factors.

Definition 3

Balancing the working load of PEs in timestep t_i means that in timestep t_i the load of every PE is equal to the average load of PE. Load Balance of timestep t_i can be measured as the following formula:

$$Balance_{t_i} = \frac{1 - \sum_{j=1}^{p} |load_j^{t_i} - load_{ave}^{t_i}|}{p \cdot load_{ave}^{t_i}}$$

$Balance_{t_i}$: the Load Balance factor of timestep t_i;

p: the number of PE, in the system;

$Load_j^{t_i}$: the load of PE$_j$ timestep t_i;

$Load_{ave}^{t_i}$: the average load of PE in timestep t_i;

$$Load_{ave}^{t_i} = \frac{\sum_{j=1}^{p} Load_j^{t_i}}{p}$$

Definition 4 Communication Overhead

The Communication Overhead of a system is measured by the sum of the amount of data communication between the two neurons which are placed on different PE$_s$. Communication Overhead of a system can be measured as the following formula:

$$Comm\_sys = \frac{\sum_{i=1}^{p}\sum_{j=1}^{p} C_{ij}}{C_{NN}}$$

Comm_sys: the Communication Overhead factor of the system;

C_{ij}: the amount of communication between the neurons on PE$_i$ and the neurons on PE$_j$;

C_{NN}: the amount of communication between neurons of the neural network model;

p: the number of PEs in the system.

1.3 The Mapping Criterions

In order to apply all kinds of neural network models to resolve specific problems efficiently on the general purpose parallel neurocomputer, we map the neurons of the neural network model onto PEs according to the following mapping criterions.

1) Balancing the working load of PEs

2) Reducing the communication overhead among PEs

2 Absorb Algorithm

At present, many researchers have advanced many mapping algorithms, which all have a back modification process. The back modification will affect the mapping result and mapping speed directly, so these mapping algorithms can not realize real time assignment.

2.1 The Overview of Absorb Algorithm

After the research of the neural network model and mapping criterions, we adopt the idea of the graph theory, and convert the neural network mapping process into the evolution of a series of directed graphs, from the initial one to the stable one. The initial directed graph represents the status of the neural network model and PEs before the mapping process. The stable directed graph represents the mapping result. Directed graph $G=\langle V,E\rangle$ stands for a neural network model consisting of N neurons and P PE_s, V is the node set of G, which includes N subnodes and P homenodes, a subnode stands for a neuron, a homenode stands for a PE, and it has a status set, recording the neurons mapped onto the PE; E is the edge set of G, the edge is weighted, the weight represents the amount of data communication between two nodes.

Now we discuss some different directed graphs:

1)Initial Directed Graph $G_s=\langle V_s,E_s\rangle$(see Fig. 1)

V_s includes N subnodes and P homenodes, the status set of each homenode is empty;

E_s only includes the edges between the subnodes (the connections between the neurons of the neural network model), the weight of edge is the amount of the communication between the corresponding neurons. There are no edges between the homenodes, and there are no edges between the homenodes and the subnodes.

2) Temporary Directed Graph $G_t=\langle V_t,E_t\rangle$(see Fig. 2)

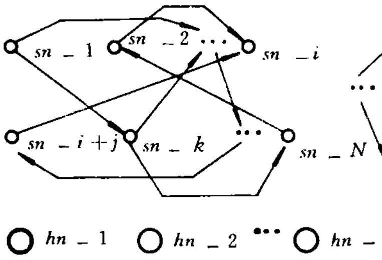
G_t represents the status of the neural network model and the PEs during the mapping process, the status set of each homenodes records the neurons mapped onto the corresponding PE.

E_t includes the edges between the nodes of V_t.

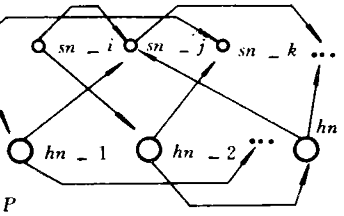
3) Stable Directed Graph $G_d=\langle V_d,E_d\rangle$(see Fig. 3)

V_d only includes p homenodes, the status set of each homenodes records all the neurons mapped onto the corresponding PE.

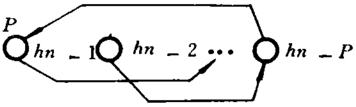
E_d includes the edges between the P homenodes, the weight of the edge represents the amount of communication between the corresponding PEs.



$sn-i$: subnode i;
$hn-j$: homenode j
Fig. 1 Initial directed graph

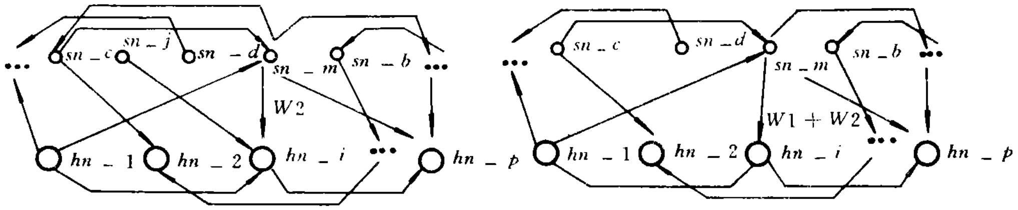


$sn-i$: subnode i;
$hn-j$: homenode j
Fig. 2 Temporary directed graph



$hn-j$: homenode j
Fig. 3 Stable directed graph

The key of Absorb Algorithm is how to map the neurons onto the PEs. We will describe the process of mapping the neurons onto the PEs ——absorbing process(see Fig. 4). Assuming that k timesteps of mapping process have been finished,the current status



$sn-i$: subnode i;
$hn-j$: homenode j
(a)

$sn-i$: subnode i;
$hn-j$: homenode j
(b)

Fig. 4 Absorbing process

of neural network model and PEs can be represented as directd graph $G_k=\langle V_k,E_k\rangle$(see Fig. 4a). Under the consideration of Load Balance and Communication Overhead,neuron n_j will be mapped onto PE_i. In Absorb Algorithm,it is called that homenode i absorbs subnode j, the absorbing process can be realized by several operations on directed graph. Firstly,the subnode j is put into the status set of homenode i. Secondly,search all the nodes connected with subnode j in V_k,except for homenode i. Assuming that node m is one of them, if the direction of the edge between node m and node i is the same as that of the edge between node m and subnode j(node m is the source node or the destination node of the two edges), the sum of weights of two edges is the new weight of the edge between node m and homenode i,remove the edge between node m and subnode j and the edge between subnode j and homenode i from E_k, otherwise,generate one edge between node m and homenode i with the same direciton as that of the edge between node m and subnode j,and its weight is equal to that of the edge between node m and subnode j,then remove the edge between node m and subnode j and the edge between subnode j and homenode i from E_k, now we get a new edge set E_{k+1}. This is to say, if neuron n_j is mapped onto PE$_i$, the communication between neuron n_j and other neurons which are not mapped onto PE$_i$ will be converted into the communication between PE$_i$ and other neurons which are not mapped onto PE$_i$.

Finally, remove subnode j from V_k, we get a new node set V_{k+1}. Now, the process of mapping one neuron onto one PE is accomplished (see Fig. 4b).

2.2 Description of Absorb Algorithm

Step_1: Build the initial directed graph $G_s = \langle V_s, E_s \rangle$ representing the neural network model (constructed by N neurons and P PEs).

Step_2: Group the subnodes of G_s according to the timesteps, each subnode must be in only one group. Set *sum_group* to the number of groups.

Step_3: Set the code of current mapping group to 1, *c_group*=1,
Calculate the maximum number of subnodes which can be absorbed by a homenode in this timestep, $max = \left\lceil \frac{num\_c\_group}{p} \right\rceil$
num_c_group is the number of subnodes of the current group.

Step_4: Set the code of current mapping neuron to 1, *c_neuron*=1.

Step_5: Sort P homenodes in descending order according to the sum of weights of edges between the homenode and the current subnode. If the number of subnodes in the current group mapped onto the first homenode in less than *max*, the homenode absorbs the current subnode; otherwise, consider the second homenode, the third homenode…until one homenode can absorb the current subnode.

Step_6: If *c_neuron*<*num_c_group* then *c_neuron*=*c neuron*+1; goto step_5.

Step_7: If *c_group*<*sun_group* then *c_group*=*c_group*+1; goto step_3.

Step_8: Output the mapping result, calculate *Balance_sys* and *Comm_sys*.

Step_9: End.

2.3 Experimental results

neural network model	the number of PEs	mapping results		*Balance_sys*	*Comm_sys*
		code of PE	codes of the neurons mapped onto the PE		
Parallel Hopfield Net (constructed by 5000 neurons coded from 1 to 5000, fully connected between the neurons)	4	1	1~1250	1.0	0.75
		2	1251~2500		
		3	2501~3750		
		4	3751~5000		
BP Net (constructed by 4500 neurons, coded rom 1 to 1000 in the input layer, from 1001 to 4000 in the hidden layer, from 4001 to 4500 in the output layer, fully connected between neurons of adjoint layers)	5	1	1~200 1001~1600 4001~4200	1.0	0.80
		2	201~400 1601~2200 4001~4100		
		3	401~600 2201~2800 4101~4200		
		4	601~800 2801~3400 4301~4400		
		5	801~1000 3401~4000 4401~4500		

3 Conclusion

The parallel distributed processing feature of ANN is explored in the mapping process, and more accurate definitions of Load Balance and Communication Overhead are presented. Absorb Algorithm gets rid of the back modification of the existant algorithms, and improve the quality of mapping result and the speed of mapping. Absorb Algorithm finds a new way to resolve the mapping problem in the ANN virtual implementation field. It can map all kinds of neural network models. All kinds of experiments show that Absorb Algorithm is a real time algorithm.

References

1 Fogelman-souli F, Herault J. Neurocomputing: Algorithm, Architectures and Applications. Springer-Verlay Berlin, 1990

2 Ghosh J, Hwang K. Mapping neural networks onto message-passing multicomputers. J Parallel Distributed Computing, 1989, 6(2): 291～330

3 Kohonen T. Introduction to neural computing. Neural Network, 1988, 10(1): 3～16

4 Kung S y, Hwang J N. Parallel architectures for ANNs. IEEE Int. Conf. NNs, ICNN'88, 1988, 2: 165～172

5 Kung S Y, Hwang J N. A unified systolic architecture for ANNs. J. Parallel Distributed Computing, 1989, 6 : 357～387

6 Wasserman P. Neural Computing, Theory and Practice. Van Nostrand Reinhold, New York, 1989

7 Hwang J N. Algorithms/Application/Architecture of ANNs: [Ph. D. Dissertation]. Dept. of EE, Uni. of Southern California, 1988

8 Schwartz(ED) E. Computational Neuroscience. MIT press, 1988

一种优化神经网络映射算法——吸收算法

王意洁　胡守仁

（国防科技大学计算机系　长沙　410073）

摘　要　本文揭示了神经计算的本质——并行分布处理，并以此为基础提出了时间步的概念。分析了映射算法的两个重要概念——负载均衡和通讯开销，并提出了映射分配准则。在神经网络的映射分配中引入图论的有关思想，提出了一种优化的神经网络映射算法——吸收算法。最后给出了重要的试验结果，这些数据表明吸收算法是一种有效的映射算法。

关键词　神经网络，映射算法，负载均衡，通讯开销，时间步

分类号　TP301　　（责任编辑　张　静）