第16卷 第7期
2006年7月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol.16 No.7
Jul. 2006

一种并行BP神经网络的动态负载平衡方案

赵 莉[1],程 荣[2]

(1.湖南信息职业技术学院 计算机工程系,湖南 望城 410200;
2.华中科技大学 集群和网格计算湖北省重点实验室,湖北 武汉 430074)

摘 要:为了加快在大规模神经网络训练下并行技术的训练速度问题,从BP算法的内部结构分析了BP神经网络算法的大规模行划分方法,提出了一种动态负载平衡方案。通过在PC集群环境下对并行算法的试验结果表明,这种并行划分提高了加速比,具有现实意义。

关键词:BP神经网络;并行;动态负载平衡

中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2006)07-0067-03

A Dynamic Load Balancing Scheme for Parallel Back-Propagation Neural Networks Algorithm

ZHAO Li[1],CHENG Rong[2]

(1.Department of Computer Engineering,Hunan College of Information,Wangcheng 410200,China;
2.Key Lab.of Cluster and Grid Computing,Huazhong University of Science & Technology, Wuhan 430074,China)

Abstract:In this paper,a parallel BP neural network algorithm is presented in order to speed up the training of large scale neural networks. And presents a dynamic load balancing scheme. To demonstrate the gain in performance provided, this algorithm is realized in MPI programming environment of the PC cluster. The result indicates that the main goal of speeding up the computation was achieved.

Key words:BP neural network;parallel;dynamic load balancing

0 引 言

近年来,神经网络理论的应用取得了令人瞩目的进展,特别是在人工智能、控制和优化、通信和空间科学等领域得到了广泛的应用。目前,在人工神经网络的实际应用中,BP网络是神经网络里应用最广泛的一种网络,它使用的是反向传播算法(back-propagation algorithm),该算法采用权值空间的随机梯度下降方法,使得收敛速度缓慢,导致训练时间长。在实际应用中,经常遇到大规模神经网络训练的情况,这时不得不考虑收敛速度和训练时间的问题。并行技术为解决大规模神经网络问题提供了一个可行的方案。近十年来随着并行分布式计算技术的发展和神经网络的广泛应用,国外许多人开始研究并行神经网络,早期的研究多集中在并行神经网络算法的设计和实现上,如文献[1]给出了基于节点并行的BP网络在Intel iPSC/860超立方体结构上的实现,文献[2]在MasPar MP-1216上实现了将训练集并行(数据并行)和节点并行结合BP网络并行算法等等,这些大多都没有考虑通信动态负载平衡的问题,文献[3]用Java实现了在异构环境下一个并行SOM(Self-Organizing Map)神经网络的应用,并提出了一个动态负载平衡方案,一定程度上改进了动态负载平衡问题,但并未给出具体的任务划分方法。

文中从反向传播算法的原理剖析了BP神经网络(Back-propagation NN)的并行划分方法,提出了在异构网络环境下或多类型处理机的PC集群环境下的一种动态负载平衡方案。

1 并行神经网络算法

BP神经网络是一种单向传播的多层前向网络,它采用误差反向传播算法训练网络。一次反向传播学习由两步组成:一次前向通过网络计算误差和一次反向通过网络计算权值。对于一个给定的训练集,反向传播学习有两种方式:串行的方式和集中的方式。

反向传播算法的串行方式也称为在线方式(on-line)、模式方式或随机方式。在这种方式里算法在每个训练模式(输入向量)呈现之后进行权值更新,网络的所有神经元的权值都是在一个模式(输入向量)接着一个模式的基础上调整的。而在反向传播学习的集中方式中,权值更新要在组成一个回合的所有训练模式呈现后才进行,所有的权值的调整是在一个回合接着一个回合的基础上进行的[4]。

收稿日期:2006-02-22

作者简介:赵 莉(1980-),女,湖北枝江人,助教,研究方向为并行分布式处理软件。

串行方式的随机性质使得要得到算法收敛的理论条件变得困难了，导致串行方式比集中方式慢得多，因此，集中方式比串行方式更容易并行化[4]。所以这里只考虑集中方式下的并行划分，权值更新策略采用集中方式(批处理方式)。神经网络的并行方式主要可分为数据并行和结构并行两种。

(1) 数据并行。

在前向计算部分，训练样本以流水线的形式逐个输入网络，通过各个节点的计算得到一个相应的输出值和误差值，一个回合结束后，根据误差调整整个网络权值。由于权值的改变是在一个回合的所有训练样本都呈现后才进行的，所以在一个回合中，不同的训练样本之间不存在数据依赖，因此它们的计算可以同时执行，这样就可以将训练集划分成多个子集，分别放在多个处理机上(这样就完成了前向计算的任务划分)，然后用一个主处理机汇总误差并执行权值更新。MPI 算法描述如下：

①所有处理机分配同一的初始网络权值 W(包含偏置 b)；

②获得机器 rank；

```
If( rank= =0)//主处理机
{ 更新网络权值 W;
Send(W);//向所有从处理发送更新后的权值
MPI_Recv(误差值 E);//接收所有从处理机发送来的误差值}
Else //本机非主处理
{ Receive(W);//接收主处理机发送来的更新后的权值
进行下一个回合的训练;
MPI_Send(误差 E); //向主处理发送误差值
}
```

(2)结构并行。

神经网络处于同一层的各节点之间存在着天然的并行处理能力[5]。由于同一层的各节点之间没有连接，因此没有通信的必要。它们可以在同一时刻并行执行各自的计算任务，因此可以将这些并行节点划分在不同的处理机上，从结构上将计算任务分解，这种结构并行也叫节点并行(node parallelism)。这种划分方法的关键是在节点和处理机之间建立一个合适的映射。多隐层的 BP 网络的节点的划分会带来频繁的通信，可以采用节点并行和数据并行混合的方式[4]。

2 动态负载平衡方案

在数据并行划分下由于所有处理机执行相同的主控程序，因此尽管计算是并行的，但处理机之间的通信是串行的，它们共享同一个通信信道。在同一时刻，主处理机只能与一个从处理机通信。一次权值的更新过程(一个回合)可以用如下的时序图 1 表示。

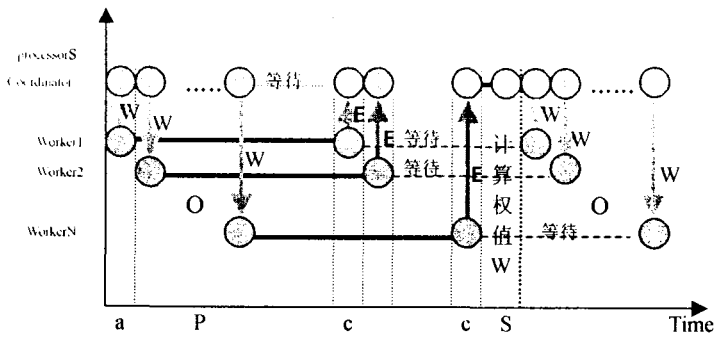

图 1 原时序图

用 Coordinator 表示主处理机，Worker 表示从处理机。假设主处理机向一个从处理机发送权值 W 的时间为都为 a，从处理机个数为 N，这里假定每个从处理机的处理速度都一样，且完成一次前向计算的时间都为 P，每个从处理机向主机发送误差信号所花的时间都为 c，主机汇总误差信号并更新权值 W 所花的时间为 S。则这种并行方式下，一个回合所用的时间为：

$$T_{para} = a + P + N \cdot c + S$$

串行执行时，一个回合所用时间为：

$$T_{serial} = N \cdot P + S$$

加速比为：$r = \dfrac{N \cdot P + S}{a + P + N \cdot c + S}$

从图 1 可以看出，在一个回合中，各个处理机都有闲置的时间段。主机在向最后一个从机发送完权值 W 起始，到第一个从机(最先收到 W)将误差 E 发送到主机时间段内主机处于等待状态，等待时间为 $a + P - (N-1)a$；第一个从机向主机发送误差信号 E 起始，到主机计算完 W 的时间内从机处于等待状态，等待时间为 $(N-1)c + S$。

显然，这种任务的分布下闲置资源未得到充分的利用。文献[3]针对并行 SOM(Self－Organizing Map)神经网络提出用处理速度最快的处理机处理最大负荷工作的方法平衡负载，这种方法在一定程度上确实减少了执行时间，但并未指出任务具体如何划分，并且相同的任务由一个块处理机执行的时间未必比相同的任务由多个慢处理机一起执行的时间短。

针对上述问题，文中提出了一种基于并行神经网络的动态负载平衡方案。一个处理机一次计算时间取决于两个因素：处理机的处理速度和任务的大小。假定每个处理机的速度都不一样，用 v_i 表示第 i 个处理机的速度，定义为该机每秒钟处理的样本个数，n_i 表示第 i 个处理机处理的样本个数，则处理机 i 的一次计算时间为 $t_i = v_i n_i$。现假设事先知道 v_i，主机在获得 N 个从处理机后，将训练集 N 等分，获得各机的速率，计算 t_i，对处理时间按照从小到大进行排序，然后主机最先向处理时间最长(速度最慢)的处理机发送消息，使之开始计算一等份任务，然后向处理时间次长的发送，最后向计算时间最短的处理机发送数据。见如下时序图 2。

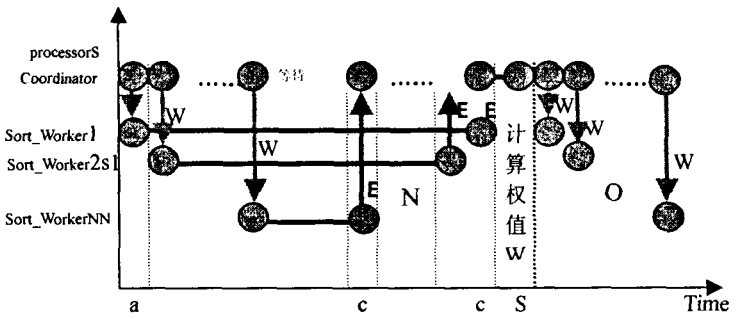

图 2　优化方案下的时序图

这样就不用考虑任务的分布问题了,关键是知道处理速度。由于各个数据是并行的,各个从处理机处理完全相同的网络、相同数量的训练数据,每次前向计算处理相同个数的神经元;每个相应的神经元所执行的相同的加法、乘法和指数操作,故速率取决于处理器的速度。因为不需要精确值,所以只需各处理机相对速率因子即可,假设每机速率因子为 k,这样就可以直接针对不同的 k 进行排序。

下面给出这种方案下的加速比。

设 P_i 为第 i 个处理机一个回合的计算时间,P_1 为第一个处理机的计算时间,则

$$P_i = P_1 - (i-1)(a+c) \tag{1}$$

设 v_i 表示第 i 个处理机的速度,定义为该机每秒处理的样本个数,n_i 表示第 i 个处理机处理的样本个数,则总样本数:

$$n = \sum_{i=1}^{N} P_i v_i = P_1 \sum_{i=1}^{N} v_i - (a+c)\sum_{i=1}^{N} v_i(i-1) \tag{2}$$

由式(1)和(2)得

$$P_1 = \frac{n + (a+c)\sum_{i=1}^{N}(i-1)v_i}{\sum_{i=1}^{N} v_i} \tag{3}$$

所以 $T_{para} = a + P_1 + c + S$,因此加速比 r 为:

$$r = \frac{N \cdot P + S}{a + P_1 + c + S} = \frac{N \cdot P + S}{a + \dfrac{n + (a+c)\sum_{i=1}^{N}(i-1)v_i}{\sum_{i=1}^{N} v_i} + c + S} = \frac{N \cdot P + S}{\dfrac{n + (a+c)\sum_{i=1}^{N}(i \cdot v_i)}{\sum_{i=1}^{N} v_i} + S}$$

结构并行的负载平衡也可以采用以上类似的方法,不同的是任务的划分是对节点的划分。

3　实验结果分析

在 MPI 环境下实现了数据并行的 BP 神经网络算法,采用集中式的训练方式,任务(训练集)平均划分给各个从处理机,主处理机负责协调和更新权值。试验平台是基于 Windows 的 PC 集群。

表 1 显示了随着处理机个数的增加得到的不同的训练时间。

表 1　不同处理机个数得到的训练时间

处理机个数	训练时间(s)	处理机个数	训练时间(s)
1	0.002248	4	0.003944
2	0.001839	5	0.004587
3	0.002651	6	0.004980

从图 3 训练时间随处理机个数变化曲线可知,当有两个处理机时,可以获得加速比为 0.002248/0.001839=1.2224,而当增加到更多的处理机时,加速比减小了,甚至并行时间超过了串行执行的时间,原因是处理机增加后主处理机要和更多的处理机通信,原来的计算虽然时间减少了但通信开销加大了,通信增加的时间超过了计算时间减少的部分,所以整个训练时间增大,加速比就减小了,但是从曲线的变化趋势看,随着处理机个数增加,总的训练时间增加的速度在减小。可以预知,当处理机增加到一定数目时,曲线会趋于平缓或者下跌,说明这种并行划分确实减少了计算时间,是有意义的。

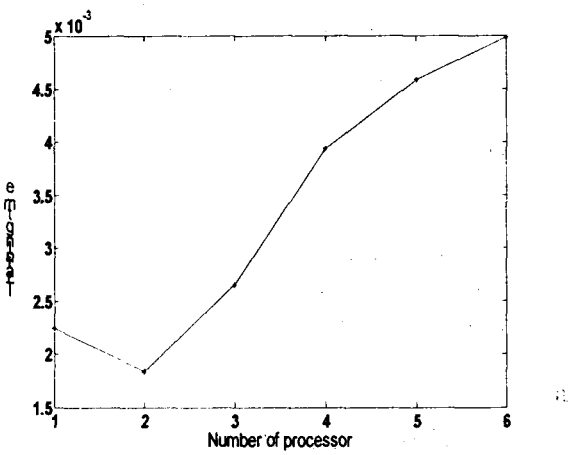

图 3　训练时间随处理机个数变化曲线

总的训练时间由计算和通信两部分组成,分别表示为 $t_{computation}$ 和 $t_{communication}$,则 $t_{total} = t_{computation} + t_{communication}$,所以 $\Delta t_{total} = \Delta t_{computation} + \Delta t_{communication}$。

当 $|\Delta t_{computation}| > |\Delta t_{communication}|$ 时加速比增加。可见,在这种并行算法下,要获得可观的加速比,取决于总的任务的大小和处理机个数。在大训练集的情况下,处理机个数增加时通信时间会远远小于计算机时间,加速比会增大。所以在大规模的神经网络的应用情况下,这种并行算法可以获得一定的加速比。

4　结束语

在大规模并行神经网络应用下,通信的负载平衡问题是很关键的一个问题,尤其是在异构网络环境下通信开销和延迟尤为突出。由实验结果及其分析可知,文中提出的动态平衡方案在一定程度上可以改善该问题。虽然文中

(下转第 72 页)

表 1 路径提取对比实验结果

方法		识别率(%)	目标错分比例(%)	背景错分比例(%)	其他原因错分比例(%)
Sobel 算子(t=0.09)		42.86	35.71	7	14.43
Laplacian 算子		71.42	21.43	4	3
Prewitt 算子	t=0.08	32.14	28.57	21.43	17.86
	t=0.06	53.57	7.14	25	14.29
	t=0.04	60.71	3.57	17.86	17.86
	最优情况	75	—	—	—
文中算法		78.57	3.57	10.71	7.15

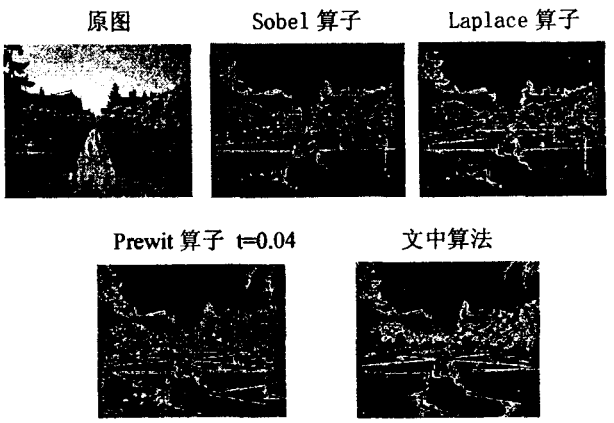

图 2 部分实验图像结果

通过实验,发现使用形态学的方法提取的路径特征经过细线化后拟合的直线较自然,便于在形状分析中拟合直线及进行进一步的长度、曲率等特征的判断,尤其在多路径环境下,提取路径的几何参数对于形状及连通性的分析具有极其重要的意义。总体上使用文中的算法对于近景、干扰较强的图像做路径的特征提取要优于传统方法,识别率与使用 Prewitt 算子相当,但抗噪性明显高于其他方法;在对航拍多路径图像的分析中,文中算法不如使用 Prewitt 算子,主要原因是因为在航拍图像中,路径均为远景,常常由于算法的去噪作用将有效的部分当作噪声消除。

5 结束语

将基于数学形态学的图像边缘算法应用于对含多路径的自然图像进行特征提取,通过实验及分析,使用形态学的方法对类直线的图像区域的对象边缘的特征提取具有很强的优势,主要表现在:a. 细线化的程度较高,仅为一个像素,便于存储及形状分析;b. 对细小背景对象的分割较彻底,易于通过滤波等方法消除;c. 有很强的抗噪性,便于复杂背景环境的使用。

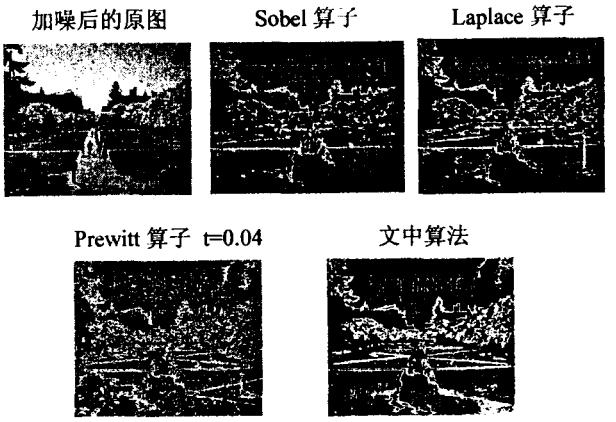

图 3 加入方差为 10 的高斯噪声后的部分实验图像

参考文献:

[1] 吴 谨,熊理良. 基于场扫描的 AGV 路径识别[J]. 武汉科技大学学报(自然科学版),2005(3):284-287.

[2] 王荣本,徐友春,李 兵,等. 基于线性模型的导航路径图像检测算法研究[J]. 公路交通科技,2001,18(2):47-51.

[3] 徐友春,王荣本,纪寿文. 智能车辆视野及其图像变形矫正的研究[J]. 公路交通科技,2000,17(5):76-80.

[4] Gonzalez R C. 数字图像处理(第 2 版)[M]. 阮秋琦等译. 北京:电子工业出版社,2003.

[5] Pratt W K. 数字图像处理(第 2 版)[M]. 邓鲁华等译. 北京:机械工业出版社,2005.

(上接第 69 页)

的动态平衡方案是基于 BP 网络提出的,但此方案同时也适用于其它神经网络。它不仅适合于有不同处理机的集群环境,也适合异构网络环境。

近年来,随着 Internet 的发展,PC 机群日益膨胀,尤其在高校和研究机构闲置资源随处可见,这为并行分布式计算提供了廉价的计算资源,随着神经网络的深入应用,基于神经网络的大型异构网络环境下的并行计算必将有可观的发展前景。

参考文献:

[1] Jackson D, Hammerstrom D. Distributing back propagation networks over the Intel iPSC/860 hypercube[A]. Proceedings of International Joint Conference on Neural Networks[C]. Seattle, USA:[s. n.],1991. 569-574.

[2] Chinn G. Systolic array implementations of neural nets on the MasPar MP-1 massively parallel processor[A]. Proceedings of International Joint Conference on Neural Networks[C]. San Diego, California, USA:[s. n.],1999. 169-173.

[3] Labonté G, Quintin M. Network Parallel Computing for SOM Neural Networks[A]. Proceedings of the High Performance Computing Symposium[C]. San Diego, California, USA:[s. n.],1999.

[4] Haykin S. 神经网络原理[M]. 叶世伟,史忠植,译. 北京:机械工业出版社,2004. 113-120.

[5] Valafar F, Ersoy O K. A Parallel Implementation of Backpropagation Neural Network on Maspar MP-1[A]. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications[C]. Athens:[s. n.], 1995.