计算机科学2000Vol. 27№. 3

全连接和随机连接神经网络并行实现的性能分析

Performance Analysis on Parallel Implementation of Fully-and Random-Connected ANN

孙亚军[1]　刘志勇[2]　曹　磊[3]

（暨南大学电子工程系　广州510632）[1]　（西南工学院计算机系）[2]　（广东省微波通信局）[3]

Abstract　The performance of parallel implementation for fully-and-random-connected artificial neural networks on multiprocessor architecture is analyzed in this paper. It concludes that the topology of the multiprocessor architecture and the fan-in of processor have little effective. The iteration learning time of neural networks is discussed, and the maximum speedup and best size of multiprocessor system are calculated in this paper too.

Keywords　Neural network, Fully connected, Random connected, Parallel implementation, Performance analysis

直觉告诉我们：当人工神经网络算法在多处理器系统上并行实现时，处理器网络的拓扑结构和处理器节点的扇入尺寸（即输入输出规模）会影响并行算法的效率，但是对全连接和随机连接神经网络，上述结论并不成立。在神经网络的并行实现中，处理器的通信开销是一个主要的限制因素，本文将对全连接和随机连接神经网络并行实现的几个相关问题进行讨论。

1　学习时间的分解

本文的研究范围限制在这样一种情况：在每一个重复学习周期，每一个处理器都必须同时处理神经计算和处理器之间的数据传送。一个处理器用于数据传递的时间可以分解为两个部分：定量部分 t_a 和变量部分 t_v。如果所讨论的神经网络在单处理器上实现时一次学习所需要的时间为 c，那么在 n 个处理器组成的多处理器系统上实现时，完成一次学习所需要的时间为 t_i：

$$t_i=t_a+t_v+c/n \tag{1}$$

启动一个通信进程要占用一定的时间（t_a 中的一个参数）；处理器传送一个数据块所需要的时间 t_v 与数据块的大小（如字节数）成正比，所以数据块的大小也是 t_v 的一个参数。

为了讨论处理器通信时间中的定量部分和变量部分，必须考虑在处理器网络中传递数据的寻径算法。对于全连接和随机连接神经网络，每一个神经元的数据必须送给其他所有的神经元。从而，每一个处理器上的神经网络分区的数据也必须送给其他所有的处理器。可以使用这样一种标准的寻找路径的算法：假设每一个处理器有 k 个双向通信接口，与 k 个处理器直接连接，能同时与相邻的 k 个处理器通信、交换数据。在处理器中设置一个路由表，指出了通过哪一条通信链路可以最快到达一个指定的处理器。寻径算法通过处理器的双向链路，把数据传遍整个处理器网络。从而，处理器网络被看作一棵树，某一个处理器被看作根。数据传送从根开始，依次到树的每个分支。这个算法可以保证，只要有一个缓冲区，就不会发生死锁。

每个处理器运行 $n-1$ 个通信进程，在每一步，任意两个直接相连的处理器通过它们的双向链路交换一个分区的数据。因而，式(1)中通信时间的常量部分为：

$$t_a=k(n-1) \tag{2}$$

其中，k 是启动（并结束）一个通信进程的常量时间。

因为一个处理器中的所有激活值必须送到其他所有的处理器，所以通过网络传送的激活值的总数是 $m(n-1)$，这时 m 是神经网络中的结点数。因而，式(1)中通信时间的变量部分为：

$$t_v=rdm(n-1)/n=rdm(1-1/n) \tag{3}$$

其中，r 是单个激活值的字节数，而 d 是通过处理器的通信链路传送一个字节占用的时间。如果在式(1)和式(2)中，忽略处理器与自己通信的项（即，对很大的 n，$n-1$ 变为 n），从而得到一个学习周期的时间为：

$$t_i=kn+rdm+c/n \tag{4}$$

其中，c 是神经网络在单个处理器上的一次学习周期

孙亚军　博士，主要研究领域：计算机网络安全性，人工神经网络的软件并行实现，实时操作系统设计技术。

时间。

2 处理器的最优数目

通过把式(4)对 n 一次求导并把它设为0，可得到处理器的优化数目：

$$\tilde{n}=\sqrt{c/k} \tag{5}$$

对大多数类型的神经网络，在单个处理器上的重复学习时间主要取决于被处理的权值数。对权值的处理经常涉及到计算加权的输入激活值的总和，对学习型神经网络，还涉及到对单个权值应用学习规则。在没有专门化的神经网络硬件(如大规模向量处理器)的情况下，处理权值的时间线性依赖于权值数。处理一个激活值(或一个误差项、或增量)经常局限于应用激活原则，只占用很少时间。权值处理会经常涉及到乘法，偶尔还有除法运算。因为处理单个权值至少会涉及到一次乘法运算，所以处理单个权值需要的时间将等于处理少量激活值的时间。当然，精确的数字依赖于所使用的激活函数和传输规则类型，以及是否应用学习规则。

设网络中的权值数为 Fm，其中 F 是通过结点的扇入(例如，输入连接数)，而 m 仍是结点数。在全连接网络中 $F=m$。在使用常量扇入的网络类型中，F 等于某一常数 f，对常量扇入：

$$c=afm \tag{6}$$

其中，a 是处理一个连接需要的时间。在处理一个连接的时间中，包含计算它对总的输入激活值的贡献(例如，把权值与输入激活值相乘)和规则的应用。如果将式(6)带入式(5)，则得到：

$$\tilde{n}=\sqrt{afm/k}=\beta\sqrt{fm} \tag{7}$$

$$\beta=\sqrt{a/k} \tag{8}$$

式(8)显示，如果通信时间的常量部分 k 增加，则能够有效使用的处理器数目将减少。但只有在全连接神经网络中，$F=m$，才有线性关系：

$$\tilde{n}=\beta m \tag{9}$$

a 的值高度依赖于所使用的特定的学习和激活规则。估计它在10到250微秒之间。这对应于大约每秒4，000到10，000个连接的处理器速度。准确的值依赖于神经网络的类型。例如，在某一个多处理器网络上，测试 k 为250微秒，则 β 在0.2到1.0之间，用这些数字，一个有1，000个结点的全连接神经网络将最多在200得到1，000个处理器上运行，用有限的扇入，如 $f=80$，多处理器网络的最优规模将是57到289。注意，最优规模的定义排除了最小的重复学习时间，它不包括对系统代价的考虑。实际上，最后增加的处理器对减少总的重复学习时间只有很少的贡献。

3 最大加速比

在一个固定规模的多处理器网络上，通过考虑在单个处理器上的重复学习时间 c 与在整个处理器网络上的重复学习时间 t_c 之间的比例(如 c/t_c)，以及让结点数 m 变得非常大，可得到最大加速比。首先考虑通常的情况：(变量)扇入 F 依赖于结点数：$F=m^b$，$0<b\leqslant 1$，换句话说，考虑情况从全连接($b=1$)到近常数扇入(接近于 $b=0$)，从而比率变为：

$$\frac{c}{t_c}=\frac{am^{b+1}}{kn+rdm+am^{b+1}/n} \tag{10}$$

并且 m 非常大时

$$\lim_{m\to\infty}\frac{nam^{b-1}}{kn^2+rdmn-am^{b+1}}=n \tag{11}$$

那么，在相对于处理器网络来说很大的随机或全连接神经网络中，可得到的加速比与处理器网络的大小成正比。

对固定的有限扇入 $F=f(f>0)$的情况，极限变为：

$$\lim_{m\to\infty}\frac{nafm}{kn^2+rdmn+afm}=\frac{n}{1+(\frac{rd}{df})} \tag{12}$$

那么，对固定的扇入，如果 n 变得很大(但是 $n\ll m$)，加速比将趋向于常数分数 af/rd。例如，如果取 $a=120$微秒(例如一个处理器速度每秒10，000连接)，固定扇入 $f=80$，双精度 $r=8$(例如8字节表示)，传输一个字节的时间 $d=1$微秒，比例 af/rd 将会是1000，在一个包含1000个处理器的大型多处理器网络中，通过替代式(12)中的所有数字，得到一个加速比为1000/(1+1)=500。对小的处理器网络(例如小的 n)加速比将几乎线性缩小，因为分数 nrd/af 保持很小，正如在式(11)中 $F=m$ 的情况。但是在大规模多处理器网络中，一半以上的处理能力丢失给通信开销。

总结 一般认为，在多处理器网络上并行实现人工神经网络时，减少处理器之间的距离(即改变多处理器网络的拓扑结构)和处理器结点的扇入(即输入输出大小)很重要，但是这些假设对全连接和随机连接神经网络并不适用。对全连接或随机连接神经网络，处理器网络的拓扑结构对通信开销的减少只有很小的作用。对随机连接神经网络，即使扇入限制在常数 fn(这里 f 是一个大于2的小常数，而 n 是网络中的处理器的个数)，限制结点扇入也不能使通信开销大量减少。

参考文献

1 Fukuda N, et al. A transputer implementation of toroidal lattice architecture for parallel neurocomputing. Proc. IJCNN-90, Washington, DC, 1990, 2: 43～46

2 Kung S L, Hwang J N. Parallel architectures for artificial neural nets. Proc. IEEE Int. Conf. Neural Networks, San Diego, CA, July 1988, 2: 165～172

3 Huang S L, Adeli H. Parallel backpropagation learning algorithm on Cray YMP8/864 supercomputer. Neurocomputing, 1993, 5: 287～302