

神经网络理论

[俄] A.И.加卢什金 著

阎平凡 译

010010101001000111001001010101100010010



清华大学出版社

神经网络理论

[俄] A.И.加卢什金 著

阎平凡 译

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书从建造高性能计算机的观点出发讨论了神经网络的作用,详细分析了多层前馈网络的结构设计、学习算法和故障诊断问题。全书共分 4 篇:第 1 篇重点讨论前馈网络的各种典型结构;第 2 篇讨论神经网络学习中的一些理论问题;第 3 篇研究具体的学习算法;第 4 篇讨论神经网络的可靠性及故障诊断。本书理论分析深入,并有不少具体实例,每章后面都附有俄罗斯学者在神经网络研究方面的大量文献。

本书可作为从事神经网络研究和应用,以及从事模式识别领域研究的科技工作者的自学参考书,也可作为相关专业研究生的教学参考书。

原书在俄罗斯被推荐为高年级大学生及研究生的教学用书。

Alexander I. Galushkin

Russian Control System Agency

Scientific Center of Neurocomputers Novaja Basmannaja str. 20 107066 MOSCOW.

Russian.

neuro@ac.orbita.ra

А. И. Галушкин

ТЕОРИЯ НЕЙРОННЫХ СЕТЕЙ

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

神经网络理论/(俄)А. И. 加卢什金著;阎平凡译. —北京:清华大学出版社,2002
ISBN 7-302-05856-3

I. 神… II. ①加… ②阎… III. 人工神经元网络—理论 IV. TP183

中国版本图书馆 CIP 数据核字(2002)第 068443 号

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

[http:// www.tup.tsinghua.edu.cn](http://www.tup.tsinghua.edu.cn)

责任编辑: 马幸兆

版式设计: 韩爱君

印 刷 者: 北京市国马印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 18.25 字数: 348 千字

版 次: 2002 年 12 月第 1 版 2002 年 12 月第 1 次印刷

书 号: ISBN 7-302-05856-3/TP·3466

印 数: 0001~3000

定 价: 28.00 元

译者的话



本书是系列丛书“神经计算机及其应用”中的第1本,作者加卢什金教授是俄罗斯神经计算机科学中心主任,也是这一领域的主要权威人士。

目前有关神经网络的中外文书籍很多,但尚未见到介绍俄罗斯学者在这一领域的研究成果的。本书从建造高性能计算机的观点出发研究了神经网络,强调了阈值逻辑的优点及其在建造高性能计算机中的作用。本书的主要内容:详细讨论多层前馈神经网络的结构设计、学习算法和故障诊断。为衡量训练样本的质量,引入了教师技术等级的概念,并分析了它对学习结果的影响。由于引入这一概念,还可在形式上把监督学习和非监督学习统一起来。作为网络学习的目标函数,采用输出分布的各阶矩阵,这在很多情况下比只用误差平方和更为合理。在学习公式中直接给出各种情况下目标函数的梯度计算公式,而不需用多次迭代计算的反向传播算法。这些都说明俄罗斯学者在这一领域的研究有自己的特点,对我们有很好的参考价值。

原书中有不少印刷错误(特别是公式),在翻译过程中尽可能作了改正。限于译者水平,错误和不当之处在所难免,恳请广大读者给予批评指正。

译者

2002.4.15

目录

引言	1
----------	---

第 1 篇 神经网络的结构

第 1 章 从布尔元件的逻辑基础向阈值逻辑基础的过渡	27
1.1 线性阈值单元(神经元)	27
1.2 多阈值逻辑	29
1.3 连续逻辑	29
1.4 激活函数的形式	30
参考文献	31
第 2 章 神经网络结构的定性分析	34
2.1 神经网络结构的几种形式	34
2.2 有顺序前向连接的多层网络	35
2.3 多层网络的结构及其符号表示	37
参考文献	40
第 3 章 有跨越连接的多层网络结构的优化	42
3.1 关于问题复杂性的准则	42
3.2 有跨越连接的一维输入网络的方案	43
3.3 类区数上限与下限的估计	44
3.4 结构优化问题的一些特例	45
3.5 根据某种拓扑特性进行的网络结构的优化	47
3.6 有 K_p 个输出值的网络结构的优化	50
参考文献	51

第 4 章 连续神经网络	52
4.1 输入特征为连续的神经元	52
4.2 取连续值的层中神经元	53
4.3 有离散特征的连续神经元层	53
4.4 神经元的连续模型的分类	54
参考文献	58

第 2 篇 神经网络的最优模型

第 5 章 神经网络输入信号特性的研究	60
5.1 问题的提出	60
5.2 有两类样本时输入信号的联合概率分布	61
5.3 有 K 类样本时输入信号的联合概率分布	66
参考文献	68
第 6 章 建造神经网络的最优模型	69
6.1 最优模型的一般结构	69
6.2 典型神经网络分界面的解析表达	70
6.3 多维 $\epsilon(n)$ 及 $y(n)$ 时的最优模型	85
6.4 自学习状态下神经网络输入信号的先验信息	87
6.5 自学习状态下网络的一次优化准则	88
6.6 在有任意技术等级的教师和自学习状态下网络的最优模型	90
参考文献	92
第 7 章 开环神经网络的分析	93
7.1 神经网络的模拟量和离散量误差的分布规律	93
7.2 二次优化泛函的选择	100
7.3 系统“Adline”中二次优化泛函的选择	101
7.4 对应给定一次优化准则二次优化准则的形成	102
7.5 连续型神经网络	105
7.6 在有任意技术等级的教师和自学习状态下的神经网络	109
参考文献	109
第 8 章 多变量函数极值的搜索算法	110
8.1 多层神经网络二次优化泛函极值的搜索过程	110

8.2 有关多变量函数迭代搜索法的分析	110
8.3 随机逼近法	112
8.4 对变量有等式约束的多变量函数极值进行搜索的迭代方法	112
8.5 变量有不等式约束时多变量函数寻优的迭代法	117
8.6 多变量函数的局部和全局最优点的随机搜索算法	118
8.7 使用二次优化泛函的二阶导数估计值的自适应算法	119
参考文献	122

第 3 篇 自适应神经网络

第 9 章 神经网络的调整算法	126
9.1 问题的提出	126
9.2 有二值及连续输出的神经元	127
9.3 两层网络	129
9.4 由连续输出神经元构成的多层网络	131
9.5 变量存在约束的闭环调整的神经网络的构造	132
9.6 有二值输出的网络的一次优化准则的实现	134
9.7 有连续输入和 K_p 种输出神经网络中平均风险最小化的实现	136
9.8 有 N^* 个输出通道的神经网络中平均风险最小化的实现	137
9.9 多层神经网络中平均风险最小化的实现	138
9.10 输入为非平稳样本时闭环神经网络的构造	140
9.11 带跨越或反馈连接的闭环调整的神经网络的构造	141
9.12 自学习与有任意技术等级教师的闭环神经网络	142
9.13 二次优化泛函的二阶导数的估计	143
参考文献	145
第 10 章 连续型神经网络的调整	146
10.1 有连续特征的网络的调整	146
10.2 层中神经元为连续时权值的调整	147
10.3 连续神经元层网络的学习过程中参数矩阵的选取	147
10.4 有连续特征并基于给定随机样本时参数 $K^*(i, j)$ 的选择	149
10.5 连续两层网络调整算法的特点	151
10.6 连续神经元层权函数的 3 种实现方案及相应的学习过程	151
10.7 两层连续神经网络中使用二次优化泛函 α_{2g} 的学习算法	153
10.8 有分段常数权函数的连续神经元层	154

10.9 带分段线性权函数的连续神经元层·····	156
10.10 带分段常数权函数的连续神经元层的网络 ·····	158
参考文献·····	159
第 11 章 调整神经网络时初值的选择及多层网络的 典型输入信号 ·····	160
11.1 初始条件的选择方法·····	160
11.2 确定性选择初值的算法·····	161
11.3 多层神经网络中初始条件的选择·····	164
11.4 多层神经网络的典型输入信号·····	167
参考文献·····	168
第 12 章 闭环多层神经网络的研究 ·····	169
12.1 闭环调整的多层神经网络设计问题的提出·····	169
12.2 输入信号是多峰分布时神经元特性的研究·····	170
12.3 识别非平稳样本的神经网络的动态研究·····	176
12.4 学习状态下三层神经网络的动态研究·····	181
12.5 有反馈网络的一些特例的研究·····	184
12.6 自学习状态下单层神经网络的动态研究·····	188
12.7 自学习状态下的两层神经网络·····	194
12.8 闭环多层神经网络调整算法中一些有关参数矩阵选择 的工程方法·····	202
12.9 用于解决矩阵变换问题的多层神经网络的构造·····	202
12.10 用于把二进制数转换为十进制数的多层神经网络 ·····	204
12.11 有任意教师等级的多层网络的研究 ·····	205
12.12 对闭环调整的神经网络进行研究的解析方法 ·····	206
参考文献·····	214
第 13 章 可变结构多层神经网络的设计 ·····	216
13.1 第 1 层神经元的顺序学习算法·····	216
13.2 使用随机法搜索局部及全局极值的多层神经网络中第 1 层 神经元的学习算法·····	219
13.3 超平面数增多时有关算法收敛性的分析·····	222
13.4 两层神经网络中第 2 层神经元的学习·····	224
13.5 3 层神经网络中第 2 层和第 3 层神经元的学习 ·····	230

13.6	对多层神经网络做依次调整的一般方法·····	231
13.7	有连续特征的多层神经网络中第1层神经元的学习方法·····	232
13.8	用调整变结构多层神经网络的方法解决初始条件的选择问题·····	232
13.9	变结构多层神经网络的自学习算法·····	233
	参考文献·····	233
第14章	多层神经网络中有效特征的选择 ·····	235
14.1	学习状态下特征选择问题的提出·····	235
14.2	固定结构的多层神经网络中特征提取的结构方法·····	237
14.3	用于选择有效特征的第1层神经元顺次调整的多层神经网络·····	238
14.4	神经元数的最少化·····	240
14.5	自学习状态下多层神经网络中有效特征的选取·····	241
	参考文献·····	241
 第4篇 神经网络的可靠性及故障诊断 		
第15章	神经网络的可靠性 ·····	244
15.1	神经网络功能可靠性的研究方法·····	244
15.2	用多层神经网络形式实现组织自恢复时功能可靠性的研究·····	246
15.3	多层神经网络功能可靠性的研究·····	247
15.4	神经网络参数可靠性的研究·····	248
15.5	灾难性故障发生时多层神经网络功能可靠性的研究·····	256
	参考文献·····	257
第16章	神经网络的故障诊断 ·····	260
16.1	神经网络的状态图、基本概念及定义·····	261
16.2	神经网络中故障定位的算法·····	262
16.3	神经元输出端有逻辑常数故障时构造最少测试点的算法·····	268
16.4	神经网络的自适应故障诊断方法·····	269
	参考文献·····	274
结论	·····	276
	参考文献·····	279

纪念我的老师

——控制理论、离散及自适应系统
领域中的俄罗斯经典科学家：

弗·维·索洛道弗尼科夫

列·吉·库金

雅·查·崔金

引 言

1. 神经计算机

神经计算机是新一类的计算机，它的出现是由于下述客观原因：一方面是对电子计算机结构起决定作用的元件技术的发展；另一方面是对更快更有效地解决实际问题的方法的需求。

发展神经计算机的基本因素是早在 20 世纪 50 年代发展起来的阈值逻辑，它和以与、或、非为基本运算的布尔逻辑不同。阈值逻辑的发展导致了 60 和 70 年代一系列专用和通用神经计算机的建造。“神经计算机”这个专用名词与人或生物的神经系统没有任何联系，它仅指具有类似神经元栅格间简单传递函数的权值可调或固定的阈值元件系统。70 年代大规模集成电路 (LSI) 技术的急剧发展以及用 LSI 实现的基于与、或、非逻辑的传统结构的微处理器，虽然延缓但并未阻止基于阈值逻辑的计算机的发展。80 年代神经计算机的复苏使人们用新的眼光看待它。超大规模集成电路 (VLSI) 的发展，使得在一个或几个芯片上不仅可以实现或仿真神经元的处理功能，而且能实现或仿真它们之间的相互联系，这在以前是不可能的。到 80 年代中，不仅可用电子元件也可用光学方法来实现上述功能。

建造专用或通用神经计算机的出发点是构造模拟-数字型计算机。其中“快速”-模拟计算部分基于阈值逻辑实现高维运算。权值调节算法可以用模拟方式快速实现，也可用专用数字电路或借助通用计算机以数字方式慢速实现。

建造神经计算机，要求从原理上构造一种新的可解决多维问题的算法，这种算法解决具体问题所用的时间一方面应与问题的规模成线性关系；另一方面又决定

于具体神经网络实现迭代算法的收敛速度。

构造神经计算机的算法的形式化基础就是神经网络理论^[B-1~B-7]。把使算法具体化的一些基本运算组称为逻辑基础,因而对大多数任务而言,逻辑基础应是 $\{\sum ax\}$ 。属于这类任务的有向量代数运算、傅氏变换、最优化计算等。类似任务还有解常微分方程、泊松方程、斯托克斯方程、椭圆方程等。

电子计算系统的逻辑基础决定于它的基本元件。对传统的计算机而言它是与、或、非运算,用它们可组成第一级的较复杂的运算(如与非门、多级与或非等)。再进一步可构成一些运算处理器。在这种情况下,计算系统的基本运算与待解问题的逻辑基础就难以对应,因而需要附加复杂的编程过程。

对于神经计算机来说,其逻辑运算基础在最简单的情况下是 $\{\sum ax, \text{sign}\}$,它可最大限度地对应于待解问题的逻辑基础。通常情况下无需附加人工改造即可适应任务的要求。

以下约定,计算系统的逻辑基础与问题是对应的,从而可达最大计算效率。这个条件对于为解决给定问题构造的专用计算机并不困难,而对于趋于通用化的神经计算机就不简单了。

用神经计算机以迭代方法解决具体问题时,搜索多极值目标函数这一困难仍然存在。仅在此时需要(通过冯·诺依曼机、SIMD、MIMD机)用程序或程序加硬件的方法来协助实现。简而言之,在神经计算机中,对基本的实际任务可分出一个能用硬件或软硬件的结合方式最大限度地快速实现的算法核。对此算法核,神经计算机是一个最大限度的并行系统。(在上述算法核框架下)解决问题时的循环次数(即对二次优化泛函调节的循环次数)不决定于主观的电路技术方面的直观能力,也不决定于编程者的想法,而是决定于任务的物理本质及困难程度。

单一的神经网络,其输出有逐级退化的性质,这一点罗津布拉特(Rosenblatt)在构造第1层为随机连接且第1层单元有冗余的3层感知器时已经指出过。因此神经网络所能实现的功能决定于网络的结构。

神经计算机是其计算机制可视为模拟计算方法的第1个例子。它不是从对任务和元件基本功能的某种主观设想以经验方式建造的。

实现神经计算机的方法基本上有3种:

① 在 SISD(单指令流,单数据流,如经典的个人计算机)、SIMD(单指令流,多数据流,如连接机制机器)或 MIMD(多指令流,多数据流,如在 Transputer 网络上)结构的计算机上进行仿真。

② 基于数字元件,用程序加硬件方式来仿真神经型部件,以加速阈值元件的一些大量运算,主要是乘与加运算(如 Weitek 公司的处理器, TMS32020 信号处理器等)。

③ 用全硬件实现神经型部件的建造(CMOS 神经芯片,光控 Transputer,全息摄影等等)。

解决问题的效能从方法①到方法③是递升的。

用硬件实现神经型部件的建造或用程序加硬件方法仿真神经计算机时,用的是经典结构的专用计算机,一般是 SIMD 结构。在复杂条件下可用多个 SIMD 组合,构成一个异步工作的计算系统,每个 SIMD 上连接有同步工作单元,以实现所需神经型部件的运算。

超级电子计算机如 CRAY, X-MP, CYBER-205, 有很强的计算能力,但价格昂贵,其结构与神经网络处理信息的方式也不相同。它们的高速处理能力来自向量处理、流水线处理等特种处理器和缩减的时间循环。通常它们基于现代技术,而进一步提高处理能力可能会受到信号在计算模块间的传递速度这一物理因素的限制。走出这个死胡同的出路是“细粒度化”处理、通过连接机制方式,如用 Intel Hypercube, Ncube, Meiko Computing Surface 等的 MIMD 机来实现,特别是用“神经型”的处理方式。

可用如下不同的方式实现并行处理:

- ① 子任务或事件级并行;
- ② 显式算法并行;
- ③ 几何并行。

第 1 种并行方式对有不同参数的多级程序是有效的,此时任务可分为多个独立的子任务(每个子任务有自身的参数),并用不同的处理器并行完成各子任务。

算法级并行的例子是带浮点的向量处理器,这里将并行处理的组织归结为对各子任务分配流水线处理器,每个子任务都可对数据做运算并传送结果。为保证各流水线处理器的有效性,必须进行负载均衡,并核对各阶段间交换数据的时间。

几何并行指在处理器间分配数据,使每个处理器分配有中间存储器,或者处理器直接包围着中间存储器,这是 SIMD 的标准形式。实践表明,许多模拟神经网络的计算方式都是局部性的,这使得神经计算机与几何并行方式相联系,因此如上面指出过的,硬件或程序-硬件实现的神经计算机可归属于 SIMD 或 MIMD 结构这一类。

对应于神经网络处理过程的一个明显的并行实现数学运算的例子是矩阵与向量相乘。这里向量是单层神经网络的输入,矩阵是神经网络的权值,输出则是二者相乘后经非线性变换后的结果。

神经计算机属于交叉学科研究的内容,因此要阐明神经计算机首先需要对一些相关学科加以说明,其中包括

数理统计 神经计算机是能对一些较复杂的常常是多峰或分布未知的随机过程及其复合效果做形式化描述的一种系统。

数理逻辑与自动机理论 神经计算机是一种由与布尔逻辑元件不同的特殊逻辑元件——神经元组成的、可用算法解决问题的逻辑网络。其中元件之间的与众不同的连接方式值得特别注意。

阈值逻辑 (20 世纪 50 及 60 年代的计算机是基于阈值逻辑的) 神经计算机解决问题的算法是由动态可调权值的阈值逻辑元件组成的网络构成的系统,其算法与网络规模和输入元件的个数无关。实际上所有与阈值逻辑有关的方法都具有神经网络的外部属性,与布尔元件类似,对其输入元件的个数和网络规模是有限制的。

控制理论 非线性动态控制系统的综合这一问题中的困难是众所周知的。对于神经计算机而言这一困难部分由于下述原因而被绕开:其控制对象——多层网络是已经很好形式化了的,而其动态调节过程正好是解决问题的过程。可见实际上这类自适应网络的综合只是控制问题的特殊情况。

计算数学 与经典方法不同,神经计算机是以神经网络的方式来解决问题的,因而从物理上说其算法更加并行化。神经网络算法是计算数学中的新的分支,可称为“神经系统数学”。

计算技术 从计算技术观点来看,神经计算机是按 MSIMD(多 SIMD)体系组成的计算机,它的实现基于以下 3 类技术:

- ① 简化到单一处理单元(神经元)组成的系统;
- ② 单元间的连接大大复杂化;
- ③ 编程过程变为单元连接权值的调节过程。

神经计算机的一般定义:神经计算机是一种有硬件和程序支持的计算系统,适用于完成以神经网络的逻辑基础表示的算法。

2. 神经计算机在高效计算系统中的位置

如果高效计算系统在一定条件下可分为两类,即“粗粒度”(处理器较少)和“细粒度”(有大量,如几百上千个处理器),则神经计算机属于后者。

新一类计算机可认为能解决很广泛的任务,与以前计算机不同,且使用新的技术、算法和编程方式来解决问題。

按照上述观点,与冯·诺依曼机相比,粗粒度计算机 CRAY、Эльбрус 可算作新型计算机,SIMD 体系的细粒度处理机与粗粒度机相比也可算作新型计算机。

以下所列各类机器,后面的与前面的相比,都可算作新一类计算机:

- ① 冯·诺依曼机;
- ② 粗粒度机 CRAY,Эльбрус;
- ③ SIMD 体系结构的细粒度机;

- ④ MIMD 体系结构的细粒度机;
- ⑤ 混合结构的 MSIMD 细粒度机;
- ⑥ 神经计算机。

图 B-1 按发展过程给出 4 种细粒度机, SIMD 体系在 20 世纪 70 年代中首次出现,当时的处理器是带一些局部存储器的单拍处理器,因此在解决复杂任务时,需要把它们组织起来同步工作。

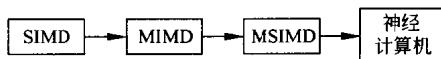


图 B-1 细粒度机发展过程

80 年代初 VLSI 技术的发展,使得出现了第一个 MIMD 体系结构的细粒度机的工业样本,它是由 16 或 32 个微计算机组成的异步网络,也包括 Transputer 网络。

1985—1988 年出现了某些混合结构的机器,它的核心是包括一些由异步工作处理器组成的核,这种核中的处理器又去控制一些由带局部存储器的简单处理器组成的同步网络(有时称这种结构为 SIMD-MSIMD 组合)。

MSIMD 是在新的水平(MIMD 体系)上重复,尝试在同步网络上执行并行算法,这在 SIMD 体系上已经实现过。

从整体上说,神经计算机可作为 MSIMD 体系结构的特例或其发展。每个“细粒度”异步工作,而各粒度之间又形成同步方式。从算法的观点看,神经计算机是有存储器的单处理器网络,用程序加硬件的方法实现同步工作,更好的方法是用硬件实现神经算法。在最简单的情形下实现高维向量相乘或向量与矩阵相乘,此时神经计算机可看作 MSIMD 结构的特例。其中单拍处理器的同步工作组具有特定的组织方式,可用程序加硬件方式实现。

在按同步方式构成的神经计算机中,当用程序加硬件方式实现同步工作时,还可执行下述两个附加任务:

- ① 使神经计算机中各异步工作的核间的信息交换量最小,有时达到无信息交换。这对大多数任务来说用 Transputer 网络或相似的系统是做不到的。
- ② 解决所谓的不良形式化的任务,例如,以最优识别为目的的学习和自学习(聚类)等。

必须指出,在所有的神经计算机的应用中,最重要的是其超级计算能力。在现有条件下缺乏有效计算工具的问题促使了神经计算机体系结构的出现,这好比 MIMD 体系的自然发展一样,希望在 MIMD 结构中增加各部件的效能以及部件个数以提高整机的效能。但是从效能/价格上看朝着这一方向的发展会因下述 2 个因素而受到限制:

- (1) 当增加部件数时,部件之间信息交换的损失会大为增加;

(2) 当提高部件效能时,其价格会大幅度上升。

用增加部件数和提高部件效能来提高 MIMD 结构的总效能这一方案遇到禁区(见图 B-2)是由于效能的增加比系统价格的上升慢(或慢得多),所以是不能让人接受的。

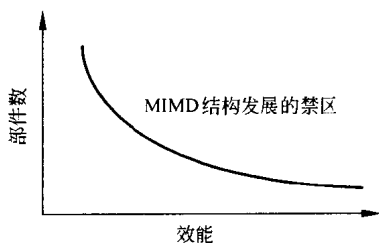


图 B-2 MIMD 中增加部件及各部件效能的限制

这也是造成在 MIMD 结构的各部件中引入同步工作的神经处理器的主要原因。神经处理器可用程序加硬件方式实现,将来可用硬件神经网络实现。

经济问题现在是将来也会是建造细粒度超级计算机的重要因素。特别是具有大量部件的那些计算机。在类 Transputer LSI 中价格的增加是由于其复杂性和高效能所致,这使得它们不宜于作为大型栅格的节点。而作为大的互相通信栅格的基本单元应使用单一类型的更廉价的由 LSI 形成的协处理器,按结构应使用带存储器的单拍处理器网络,或按功能使用神经计算器。

由此可见,由于结构复杂化或增加晶体管的数量,会使细粒度超级计算机在发展计算技术中的地位降低,因此 LSI Transputer 或类 Transputer 元件将在 PC、超级 PC 中大量使用,而在超级微型机(如 Meiko 与 Megaframe 的个人计算机)中较少使用。具有几千个 Transputer 或类 Transputer 元件的系统的出现是客观需要,而它在超级计算机中的使用则是暂时的,因为超级计算机中需要的是单一化的元件。

随着超级计算机的进一步发展,超过 10000 个节点的 Transputer 型结构看来将会陷入困境,因为对每个节点都需要在外部设备中考虑信息交换装置。根据以上情况,神经计算机才是发展超级计算机的有效途径。

3. 通用计算机的概念

各种计算机在解决不同问题时,其效率是不一样的,从这个意义上说它们在某种程度上都是专用机。当熟悉了各类计算机的应用情况后,就能扩展它们的应用范围,这种专用性也就减弱了。可把各类计算机的通用性定性地表示出来,如图 B-3 所示。

目前神经计算机是比 Transputer 型较为“专用”的计算机。而 Transputer 型又比单处理器型专用,这属于时间或资源的分配问题。在图 B-4 中定性给出了算法并行性的前景,随着并行可能性的增大,每一循环中的运算数也增加,从而缩短了解决问题的时间。

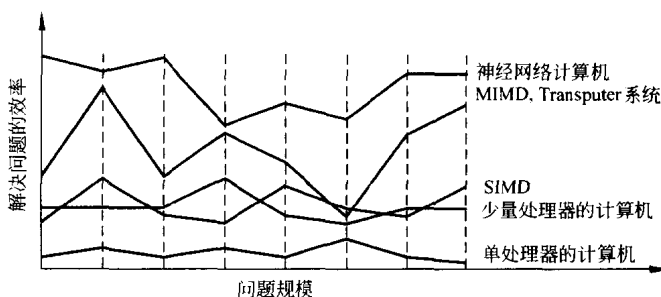


图 B-3 不同类计算机通用性的定性表示

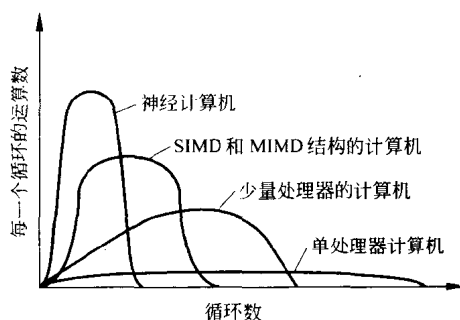


图 B-4 不同类型计算机并行化的可能性

4. 神经计算机的模块性

神经计算机的模块性(模块可扩展性)决定于对其中 Transputer 和类 Transputer 核的客观需要。

具体处理例子显示,考虑到现实情况,要增大神经计算机的效率,需要其中有 Transputer 或类 Transputer 核,以便组织各神经组块间的异步信息传递。这类核的结构可以是不同的,当异步工作的部件足够多(几百或上千)时,可能需要把核的结构由“栅格型”、“环型”改为“超立方体型”,以增大核中信息流量。

与 MIMD 机不同,神经计算机中对异步工作核的效率没有临界点(见图B-2),也无需提高其中的技术水平。

这里必须指出神经计算机异步核的特点,当用 Transputer 元件构成这种核时,保持了核之间硬件和程序上的相互兼容性,因而神经计算机并不是某种新奇的东西,而是细粒度计算机的自然进化结果。

正如 Transputer 系统一样,神经计算机可在不同水平上实现:

① 装在 PC 板上;

- ② 用 PC 控制几个组块;
- ③ 带可控 PC 的框架;
- ④ 带多个可控 PC 框架的总体(超级 PC 级)。

与 Transputer 系统相似,建造神经计算机的基本目的是建造超级计算机和进行超级计算。正因为如此,现在在 PC 上建造不同的神经计算电路板可分为 3 个阶段:纯技术阶段、仪器设备阶段以及实现阶段,其主要的战略任务是建造神经超级计算机。这可由神经计算机制造厂家 TRW 的发展路线(MARK I, II, III, IV, V 等)看出来。

神经计算机异步核的基本功能如下:

- ① 准备传送原始数据,并接收各神经组块间的处理结果;
- ② 在解决问题时,传送未完全处理好的信息,应力求使其中不正确的信息及其传送量最小化。

5. 适于用神经计算机解决的任务种类

能在计算机上解决的问题可分为 3 类:

- ① 可形式化的;
- ② 难以形式化的;
- ③ 非形式化的。

能用算法清楚地描述的问题属于可形式化的问题,对它们可清楚地判断出使用哪类计算机(SISD, SIMD, MIMD 或其他)对相应算法最合理。

算法解不惟一,或者难以估计算法解的质量的问题可归属于难以形式化的问题。例如某些规模很大的问题,由于“规模灾难”用迭代算法求解时其收敛性及收敛结果的精度都没有保证。某些可形式化的问题当算法非常繁杂费时(如大规模线性方程组)也可能出现上述问题。

非形式化问题指那些算法中的参数或函数与输入数据没有明显关系的问题,例如模式识别,聚类或自学习,特征提取等问题。

应当指出,上述各类问题与计算机型之间原则上是有联系的,但并不直接。可形式化问题适于用串行机和 SISD 结构,良好形式化的问题中有一些可通过并行算法来提高解题的效率。因而并行算法在当前是一个很大的研究领域,并行算法与计算机结构(SIMD, MIMD)也是有关的。

难以形式化的问题也可在串行机上解决。但它更适于用 SIMD 和 MIMD 结构。它也是促使这类计算机发展的原因。

可以说, SIMD, MIMD 结构的计算机适用于易于并行计算的可形式化和难以形式化的问题。