文章编号:1001 - 9081(2006)01 - 0202 - 02

一种神经网络硬件实现的可重构设计

万 勇,王 沁,李占才,李 昂
(北京科技大学 信息工程学院,北京 100083)
(wanyong2003@sohu.com)

摘 要:以 BP 网络为例,提出了一种可重构神经网络硬件实现方法。通过可重构体系结构、可重构部件的设计,可以灵活地实现不同规模、传递函数及学习方法的神经网络,从而搭建起神经网络快速硬件实现的平台。经过对一个模式识别问题的实现和测试,证明了这种设计方法的可行性。

关键词:神经网络;可重构;硬件实现;体系结构
中图分类号: TP18 **文献标识码**:A

Reconfigurable design of neural networks hardware implementation

WAN Yong, WANG Qin, LI Zhan-cai, LI Ang
(*School of Information Engineering, University of Science & Technology Beijing, Beijing* 100083, *China*)

Abstract: The BP networks was taken as an example and a reconfigurable method of neural networks(NN) hardware implementation was proposed. Based on this reconfigurable architecture and components, NN with different scales, transfer functions or learning algorithms could be implemented flexibly and fast. The implementation and test of a pattern recognition problem prove the feasibility of this method.

Key words: neural networks; reconfigurable; hardware implementation; architecture

0 引言

人工神经网络在智能控制、模式识别等领域中应用广泛。但是传统的基于通用处理器的软件实现方法存在两个主要问题:一是无法实现并行计算,因为 CPU 在一个指令周期内只能执行一条指令,导致计算速度无法满足现场的实时性需求;二是在某些嵌入式应用中(例如手机的语音识别)及对稳定性要求很高或环境恶劣的应用中(例如工业现场的控制),神经网络软件并不适用。为此,研究人员提出了多种神经网络专用硬件实现的方法和技术[1,2]。

尽管有诸多优点,神经网络数字 VLSI 实现仍面临着设计周期长的问题。因为根据应用对象的不同(例如不同的控制对象),导致所需要的神经网络在规模结构、传递函数或学习算法上都会有所差别。所以每当上述某个因素发生变化时,就需要重新进行硬件设计,这将使硬件设计人员进行许多繁琐的重复性工作。针对这个问题,本文提出了一种可重构的设计方法,从而搭建起神经网络快速硬件实现的平台,其基本思路为:将神经网络算法划分为几种基本运算,这些基本运算由可重构单元(Reconfigurable Cell,RC)完成,RC 间以规则的方式互相连接,当神经网络发生变化时,只要增减 RC 的数量或替换不同功能的 RC 就重构成新的神经网络硬件。

1 BP 网络的可重构性分析

可重构计算系统提供了一种介于通用计算机和专用计算系统之间的计算手段。如果某一计算系统能够利用可重用的硬件(包括硬核)资源,根据不同的应用需求,灵活地改变自身的体系结构,以便为每个特定的应用需求提供与之相匹配的体系结构,那么这一计算系统就称为可重构的计算系统,其体系结构称为可重构的体系结构。

由神经网络构造及其数学模型可知,神经网络可以表示为一个四元组[V, E, f, s]。其中 V 是神经元结点的集合,E 是神经元之间联接的集合,f 是激活函数,s 是学习算法。可见,神经网络的可重构性表现为结构可重构、激活函数可重构和学习算法可重构。

从计算的角度考虑,BP 网络的算法可以划分成三个步骤:

前向传播:
$$net_i = \sum_j w_{ij} a_j + \theta \tag{1}$$
$$o_i = f(net_i) \tag{2}$$

误差反向传播:对于输出层
$$\delta_i = f'(net_i)(y_i - o_i) \tag{3}$$
对于隐含层
$$\delta_j = f'(net_j) \sum_k \delta_k w_{kj} \tag{4}$$

权值更新:
$$w_{ij}(t+1) = w_{ij}(t) + \alpha \delta_i o_j \tag{5}$$

式(1)~式(5)共包含加、乘、乘累加和函数这 4 种运算。也就是说,无论是何种规模或是具有何种激活函数的 BP 网络,它所涉及的运算都包含于上述 4 种运算之内。通过在实际设计中进行分析,运算单元可归结为 3 种:乘累加部件 MAC、函数映射部件 F 及权值更新部件 WU。对于一个具体的应用,需要调整的只是这些运算单元的个数而已,从而实现不同的并行程度以满足不同的目标处理速度。当然,如何设计体系结构和可重构单元才能实现方便、快捷的重构也是需要分析考虑的,这些内容将在文章的下面两个部分分别给予介绍。

收稿日期:2005 - 07 - 09;**修订日期**:2005 - 08 - 29

作者简介:万勇,男,硕士研究生,主要研究方向:计算机体系结构、IC 设计; 王沁,女,教授、博士生导师,主要研究方向:计算机体系结构、网络与通信 SOC、嵌入式系统; 李占才,男,副教授,主要研究方向:计算机体系结构、IC 设计、信息安全; 李昂,男,博士研究生,主要研究方向:计算机体系结构、IC 设计.

2　可重构目标下的 Systolic 体系结构设计

BP 网络的计算是逐层进行的,各层间的计算有数据依赖关系,而层内的计算可并行执行。层内并行有两种方式:联接并行(Synapse Parallelism)的并行度最高,同一层的每个连接都对应一个处理单元,可同时进行计算;而神经元并行(Neuron Parallelism)则是每个处理单元与同一层的各神经元相对应,对神经元的各个连接依次计算,其并行度较低,但资源消耗也小。从成本角度看,神经网络硬件实现并不意味着不计硬件代价去实现最高的并行度,而是要以最少的硬件资源满足特定应用的性能需求。基于这样的出发点,我们使用神经元并行作为可重构部件的基本计算模式。

首先分析一下可重构部件之一乘累加部件 MAC 在前向工作时所需数据的特点,由于采用的是神经元并行计算模式,即一个 MAC 对应一个神经元,所以对应同一层内的两个相邻神经元的两个 MAC 所需计算数据可分为两类。如图 1 所示,一类为相关数据,即前层神经元的输出;一类为不相关数据,即各自与前层神经元的连接权值。

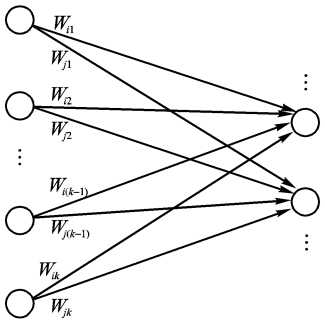

图 1　BP 网络连接示意图

如前所述,神经网络硬件实现相对于软件的一个突出优点在于硬件可以实现并行计算,所以可达到很快的处理速度。为了真正实现并行运算,对于不相关数据要采取分布存储的方式,即权值需要分布存储。而对于相关数据,本文采取 Systolic 结构[3]进行处理。

Systolic 结构也称为脉动阵列结构,它是一种有节奏地计算并通过系统传输数据的处理单元网络。图 2 显示了乘累加模块的 Systolic 结构。

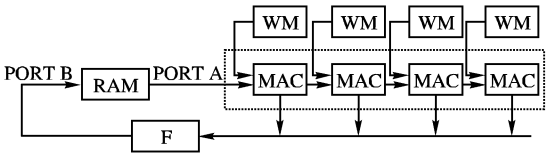

图 2　MAC 模块的 Systolic 结构示意图

在第一个并行计算部件 MAC 与 RAM 之间有一条数据总线将其连接,源操作数进入第一个并行计算部件后,依次流水传递给其他并行计算部件,各计算部件顺序产生乘累加和,该值经过 F 部件计算后将结果送入 RAM 存储。这个结构中,RAM 相当于心脏,数据相当于血液,通过流水传递的方式数据一级一级地向后传输,最后又流回 RAM,这也是脉动阵列的名称来由。

这种结构之所以适用于神经网络的实现,是因为它具有如下的优点:

1)方便实现可重构。由图 2 可知,为了达到目标处理速度而确定好所需 MAC 个数之后,只需要将各 MAC 部件依次首尾相连并把必要的数据通路接通,从而便可生成完整的局部硬件电路。

2)若图 2 中所示 RAM 的输出端口与各个 MAC 都直连的话,要求其输出端口要有很大的扇出能力,尤其当网络规模很大的时候,其扇出能力便无法得到满足,而采用 Systolic 结构则对 RAM 输出端口的扇出能力没有特殊要求。

3)在对速度与面积进行权衡之后,我们发现各 MAC 的前向计算结果需要依次流水输出,而不需要同步输出。这样在整个电路中,我们只需要用一个 F 模块即可。而 Systolic 结构恰恰可以满足我们的要求。

因为 WU 部件进行运算所需数据类型与 MAC 前向运算所需数据类型具有相似性,所以也适用这种 Systolic 结构。将本文上一节分析总结的三种基本可重构单元用这种阵列结构连接起来,就形成了下面这种可重构 Systolic 体系结构:

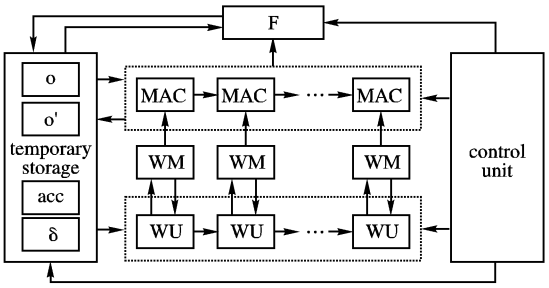

图 3　重构 Systolic 体系结构

3　可重构目标下的单元库设计

在实际设计过程中,我们总结出 BP 网络可重构单元库应该包括下面 8 种 RC:MAC、F、WU、状态机、双口 RAM、选择器、计数器,以及误差比较单元。在此只对其中两种重要的可重构单元设计进行介绍。

3.1　MAC 模块的复用设计

由于权值的分布存储和 Systolic 体系结构的确定,使得如何在前反向都能有效地利用 MAC 的资源是一个需要解决的问题。如前所述,一个 MAC 要求对应一个神经元,现以一个神经元为例来加以说明。

如图 1 所示,设网络中第 N 层含有 k 个神经元,取第 $N+1$ 层中的第 i 个神经元作为 MAC 对应的神经元来举例说明,它与前层各神经元的连接权值分别为 W_{i1}、$W_{i2}\cdots W_{i(k-1)}$、W_{ik}。权值分布存储决定了上面各个连接权值要共同存储于唯一的 RAM 中,并且此 RAM 只供第 $N+1$ 层中的第 i 个神经元对应的 MAC 使用。前向传播时,MAC 流水读 RAM 取权值进行与 out 值的乘累加运算,并最终得到 net 值。而如何在误差反向传播阶段也能利用上 MAC 来进行数据的运算是值得考虑的。根据第 N 层中第 m 个神经元 *error* 的计算公式

$$error = \sum_{n} \delta_n w_{nm} \tag{6}$$

其中 n 为第 $N+1$ 层所含神经元的个数,采取方法为让 MAC 在此阶段负责计算 $\delta_i w_{im}$(其中 $m \in [1,k]$),并加上前一 MAC 传递下来的 $\sum_{p=1}^{p<i} \delta_p w_{pm}$,并把相加结果作为新的部分积之和 $\sum_{p=1}^{p<i+1} \delta_p w_{pm}$ 传递下去。依次往下,对应第 $N+1$ 层最后一个神经元的 MAC 会流水输出第 N 层所有神经元的 *error* 值。

(下转第 219 页)

4　结语

从 Web-Logs 中挖掘出连续频繁访问路径,对于改进网站结构、进行网站智能导航等有着重要的意义。在论文中,笔者提出了一种无需产生候选集和多次扫描数据库的 OB-Mine 算法,该算法的执行效率要高于 WAP 算法,且相对于 CAP 算法而言,OB-Mine 算法在性能和它比较接近,但是 OB-Mine 算法能够应用于连续可重复频繁访问路径的挖掘。

参考文献:

[1] AGRAWAL R, SRIKANT R. Mining Sequential Patterns[A]. Proceedings International Conference on Data Engineering(ICDE 95)[C], 1995.3-14.

[2] CHEN MS, PARK JS, YU PS. Efficient Data Mining for Path Traversal Patterns[A]. IEEE Transactions on Knowledge and Data Engineering[C], 1998, 10(2):209-221.

[3] PEI J. HAN J, MORTAZAVI-ASL B, *et al.* Mining Acccess Patterns Efficiently from Web Logs[A]. Proceedings Pacific - Asia Conference on Know ledge Discovery and Data Mining(PaKDD)[C]. Kyoto, Japan, 2000.396-407.

[4] 战立强, 刘大昕. 一种在连续 MFR 中快速挖掘频繁访问路径的新算法[J]. 计算机工程与应用, 2005, (9):180-182.

[5] 何炎祥, 孔维强, 向剑文, 等. WebLog 访问序列模式挖掘[J]. 计算机工程与应用, 2003, (27):206-209.

(上接第 203 页)

MAC 组的前反向流水工作示意图如图 4 所示。

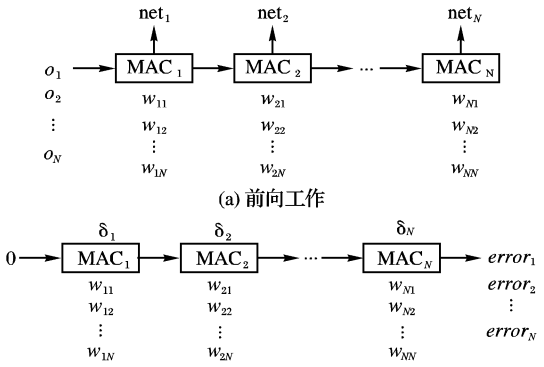

图 4　MAC 流水工作示意图

为了实现 MAC 的上述功能,其电路示意图如图 5 所示。

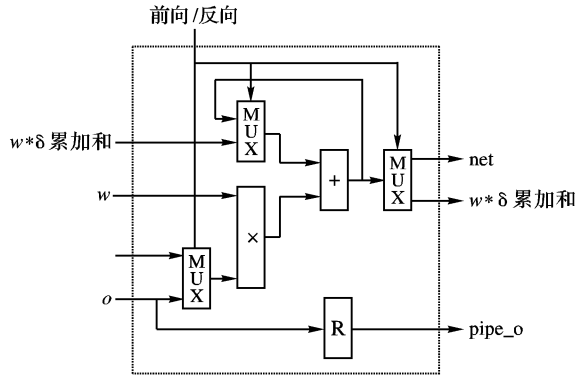

图 5　MAC 电路示意图

3.2　FSM 模块的设计

FSM 的作用是向电路中发送状态指示信息、各运算部件的初始值及使能信号。为了能够可重构进行硬件实现,基本上都是要由 FSM 发送各部件的使能信号,控制其工作,相互间基本上不使用握手信号。另外为了能够自动生成硬件代码,所以 FSM 要进行参数化设计,即各运算部件使能信号的发送时刻、各运算部件初始值的发送时刻及初始值本身、状态指示信息的发送时刻及信息本身都是通过含有参数的计算公式计算得出的。这些参数就是 BP 网络的结构参数,即神经网络层数和每层所含神经元的个数。

4　实现及结果

4.1　实现方法

本文使用上述体系结构,在可重构单元数设置为 5 和 10 的两种情况下,对一个字符(5×5 点阵表示)识别问题进行了实现。该例使用 25×10×10 的三层前向网络,激活函数为 sigmoid 函数,用基本 BP 算法更新权值。

文献[4]通过对一些典型应用的研究和分析,认为 16 位定点数是不削弱神经网络能力的最小精度要求。在本例中我们也使用 16 位有符号定点数表示数据,通过软件仿真分析各数据的变化范围,确定 16 位定点数的定标为 10 位小数。

4.2　实现结果

使用 xilinx 公司的 virtex2 pro 系列 FPGA 作为目标器件,综合结果如表 1。

表 1　综合结果

xcv2p4-6	nMAC=10	nMAC=5
Slices	1600	1056
Slice Flip Flops	1320	840
4 input LUTs	2820	1904
BRAM(18kb)	17	20
MULT(18×18bit)	22	12
Max. Freq. (MHz)	104.150	104.150

nMAC=10 时,完成一对训练向量的迭代计算需要 81 个周期,在 100MHz 运行频率下,每秒更新权值数可达 432MCUPS。速度比 PC(Pentium4 处理器,Windows2000 操作系统)软件实现提高了 2 个数量级。

5　结语

本文以 BP 网络为例讨论了可重构 systolic 体系结构设计及可重构单元库的设计问题。应该看到,这种快速神经网络硬件实现方法并非只适合于 BP 网络,可以将其推广至更多类型的神经网络。我们下一步研究工作的重点和方向是以这种体系结构为基础,进一步扩充可重构单元库并完善映射算法,使之能够完成多种类型网络的可重构实现,从而将神经网络硬件设计从 RTL 级提高到算法描述级,这也将推动神经网络硬件在相关应用领域中的实用化。

参考文献:

[1] FAIEDH H, GAFSI Z. Digital hardware implementation of a neural network used for classification[A]. The International Conference on Microelectronics[C], 2004.551-554.

[2] KIM CM, CHOI KH. Hardware design of CMAC neural network for control applications[A]. The International Joint Conference on Neural Networks[C], 2003.953-958.

[3] KUNG SY, HWANG JN. A unifying algorithm/architecture for artificial neural networks[A]. The International Conference on Acoustics, Speech, and Signal Processing[C], 1989.2505-2508.

[4] HIKAWA H. A new digital pulse-mode neuron with adjustable activation function[J]. IEEE Transactions on Neural Networks, 2003, 14(1):236-242.