# 基于 FPGA 的人工神经网络实现方法的研究

杨银涛, 汪海波, 张　志, 周建华
(厦门大学　福建 厦门　361005)

摘　要: 基于 FPGA 的神经网络实现方法已成为实际实时应用神经网络的一种途径。本文就十多年来基于 FPGA 的 ANN 实现作一个系统的总结, 例举关键的技术问题, 给出详细的数据分析, 引用相关的最新研究成果, 对不同的实现方法和思想进行讨论分析, 并说明存在的问题以及改善方法, 强调神经网络 FPGA 实现的发展方向和潜力及提出自己的想法。另外, 还指出基于 FPGA 实现神经网络存在的瓶颈制约, 最后对今后的研究趋势作出估计。

关键词: FPGA; 可重构计算; BP 算法; 神经网络; FPNA
中图分类号: TP183　　文献标识码: A　　文章编号: 1004-373X(2009)18-170-05

## Research of Artificial Neural Network Arithmetic Based on FPGA

YANG Yintao, WANG Haibo, ZHANG Zhi, ZHOU Jianhua
(Xiamen University, Xiamen, 361005, China)

Abstract: FPGA implementation of Artificial Neural Networks(ANNs) has been an approach of actual real-time neural network. This paper systematically reviews the progress that has been made in this research area over a decade, and lists the typical technology issues, shows detailed data analysis and newest research results, Simultaneously different implementation techniques and design are discussed, then existing problems and realizition constrains can be presented as well as improvement solutions, the development direction and potential of FPGA implement of ANN are proposed. Eventually future developments are expected.

Keywords: FPGA; reconfigurable computing; neural network; ANN; FPNA

## 0　引　言

人工神经网络(Artificial Neural Network, ANN)是一种类似生物神经网络的信息处理结构, 它的提出是为了解决一些非线性, 非平稳, 复杂的实际问题。目前实现 ANN 还主要依靠软件程序, 但是依靠程序很难达到实时性的要求。

神经网络在 FPGA 上实现是独立于冯·诺依曼架构, 利用 FPGA 的并行性, 在一些实时性要求很强的领域应用。通用计算机虽然编程容易, 但是很多时间浪费在分析指令, 读出写入数据等。于是人们想利用 ASIC(专用计算芯片)完成神经网络的计算任务, 但是由于资源有限, 这种芯片只限于实现特定的算法结构和小规模网络, 而且专用芯片的制作成本很高, 只适合大批量生产。

可编程逻辑器件 FPGA 的出现给 IC 设计行业一个很强的工具, 它可以小成本的开发一些专用芯片, 如果开发是成功的可以考虑流片生产。用 FPGA 实现神经网络比 ASIC 神经计算单元更容易实现, 利用可编程逻辑, FPGA 可以实现像软件一样的设计灵活性, 特别是对于复杂网络, 设计周期大大缩短, 其内部的重构逻辑模块(Configurable Logic Blocks, CLBs)包含若干逻辑单元, 利用固有的可重构路径结构可以实现高效率的连接。此外, 现在正在开发中的一种神经计算芯片为神经网络的实现提出了一种新的有效方法。

## 1　FPGA 实现神经网络关键问题分析

(1) 选择合适的神经网络及其拓扑结构

不同的神经网络有不同的应用, 而且不同的网络完成知识表达的机理是不同的, 某一个神经网络不是万能的, 对于实际问题, 首先要做的就是选择针对性的神经网络, 如线性分类问题可以用简单的感知器, 对于复杂的分类问题, 函数逼近问题可以使用 BP 网络, 对于一些聚类问题可以使用径向基(RBF)网络等。以 BP 网络结构为例, 这种被广泛采用的架构由具有错误反向传播算法的多层感知器构成(Multilayer Perceptrons using Back-Propagation, MLP-BP)[1], 训练一个 BP 网络

收稿日期: 2009-02-18
基金项目: 大学生创新性实验项目(2007 年 78 号)

主要的问题就在于: 训练开始之前, 对于网络拓扑结构缺乏一种明确的确定方法。而进行各种拓扑结构的实验并不那么容易, 因为对于每一个训练周期都要消耗很长的时间, 特别是复杂的网络, 更是如此; 其次, 对于硬件而言, 最合适的网络运算法则不仅在于它达到收敛有多么快, 还要考虑是否容易在硬件上实现且这种实现代价和性能如何; 另外, 对于同一种 NN ( Neural Network) , 其拓扑结构对网络的收敛特性以及知识表达特性都有影响, 一般增加网络的神经元或者神经元的层数, 是可以增加网络的逼近能力, 但是可能会影响网络的学习收敛情况, 而且还可能会因为过适应( Overfit) 而失去泛化能力。

(2) 正确选择数值表达形式

精度的选择对处理密度( 与耗费的硬件资源成反比) 有直接影响[1]。其中浮点数可以在计算机中表达实数, 它有相对高的精度和大的动态范围, 使用浮点数使得计算更为精确, 但是在 FPGA 上实现浮点数运算是一个很大的挑战, 而且会耗费很多硬件资源。尽管如此, 加拿大研究人员 Medhat Moussa and Shawki Areibi 仍然实现了浮点数的运算, 并进行了详细的对比分析[1]。

对于 MLP-BP 而言, Holt and Baker[2] 凭借仿真和理论分析指出 16 为定点( 1 位标志位, 3 位整数位和 12 位小数位) 是最小可允许的精度表示( 指可以达到收敛) 。以逻辑 XOR 问题为例, 文献[1]中表格 2.5( 见表 1) 表明与基于 FPGA 的 MLP-BP 浮点法实现相比, 定点法实现在速度上高出 12 倍, 面积上是浮点实现的 1/13, 而且有更高的处理密度。

表 1 FPGA 与 MLP BP 相比

| XOR ANN Architecture | Precision | Total Area (CLBs, [Slices]) | % of Convergence in thirty trials | Max. Clock Rate/MHz | Processing Density (per Slice) |
|---|---|---|---|---|---|
| Xilinx Virtex-E xcv2000e FPGA | 16 b fixed-pt | 1 239 [2 478] | 100% | 10 | 25.33 |
| Xilinx Virtex-Ⅱ xc2v8000 FPGA | 32 b floating-pt | 8 334.75 [33 339] | 73.3% | 1.25 | 0.155 1 |

同时数据也说明基于 FPGA 的 16 位定点 MLP-BP 实现在处理密度上高于基于软件方法的 MLP-BP 实现, 这最好地证明可重构计算方法的处理密度优势。应该说, 在这种应用中浮点数远不如定点数合适, 但是定点数表示的缺点在于有限精度, 尽管如此, 对于不同的应用选择合适的字长精度, 仍然可以得到收敛。因此, 目前基于 FPGA 的 ANN 大多数是使用定点数进行计算的。

(3) 门限非线性激活函数( Nonr-linear activation Function) 的实现

ANN 的知识表达特性与非线性逼近能力, 有很大部分源自门限函数。在 MLP 网络中, 门限函数大部分是非线性函数( 少数是线性函数, 如输出层的门限函数) , 但是非线性传递函数的直接硬件实现太昂贵, 目前实现门限函数的方法主要有[3]: 查表法( look-up table) 、分段线性逼近、多项式近似法、有理近似法以及协调旋转数字计算机( Coordinated Rotation Digital Computer, CORDIC) 法则, CORDIC 法则实现函数的优点在于同一硬件资源能被若干个函数使用, 但是性能相当差, 因此较少使用。而高次多项式近似法尽管可以实现低误差近似, 但是实现需要耗费较高硬件资源。相对而言, 查找表法和分段线性逼近法( 注意: 查找表不易太大, 否则速度会慢且代价也大) 更适合 FPGA 技术实现[4]。其中分段线性近似法以 $y = c_1 + c_2x$ 的形式描述一种线性连接组合( 如图 1 所示) , 如果线性函数的系数值为 2 的幂次, 则激活函数可以由一系列移位和加法操作实现, 许多神经元的传递函数就是这样实现的[5], 而查找表法则是将事先计算的数值依次存储在需要查询的存储器中来实现。

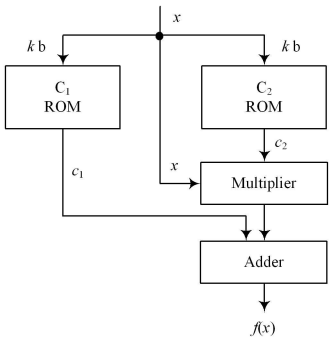


图 1 分段线性实现的硬件结构

(4) 面积节省及相关问题

为了最小化神经元实现的面积, 组成每个神经元的各个 HDL 算法模块的面积也应该最小。乘法器以及基本的传递函数( 例如, sigmoid 激活函数 tanh) 是最占用面积的, 这类问题非常依赖于所要求的精度, 尽管神经网络常并不要求很精确的计算, 但是不同的应用所要求的精度不同。一般来讲, 浮点运算要比定点运算需要更大的面积, 比如浮点运算中的并行加法器本质上是定点运算超前加法器加上必要的逻辑块, 减法器、乘法器

也类似如此, 这在激活函数实现方面更加突出, 文献[1]中面积优化对比显示, 32 位浮点运算要比 16 位定点运算大 250 倍。另外, 对于小型网络, 分布式存储器很适合权值存放, 但是对于大型网络, 权值存储器不应该被放置在 FPGA 中, 因此当 ANN 得到有效实现的时候, 就要认真考虑存储器的存取问题。其次, 神经网络应用有一个显著的缺陷: 在神经计算方面, 不同运算的计算时间和实现面积并不平衡。在许多标准神经模式中, 计算时间的大部分用在需要乘法器和加法器的矩阵向量运算中, 而很多耗费面积的运算如激活函数, 又必须被实现(它们占用很少的运算时间), 而 FPGA 的面积是严格一定的, 因此可将面积的相当一部分用来实现这些运算, 以至于 FPGA 仅剩的一小部分却实现几乎所有的运算时间。

(5) 资源和计算速度的平衡(Trade-off)

对于 FPGA, 科学的设计目标应该是在满足设计时序要求(包括对设计最高频率的要求)的前提下, 占用最少的芯片资源, 或者在所规定的占用资源下, 使设计的时序余量更大, 频率更高。这两种目标充分体现了资源和速度的平衡思想[6], 作为矛盾的两个组成部分, 资源和速度的地位是不一样的。相比之下, 满足时序、工作频率的要求更重要一些, 当两者冲突时, 采用速度优先的准则。

例如, ANN 的 FPGA 实现需要各种字长的乘法器, 如果可以提出一种新的运算法则, 从而用 FPGA 实现变字长的乘法器, 则可以根据需要调整字长, 从而提高运算速度的可能性, 其中, 基于 Booth Encoded optimized wallence tree 架构(见图 2)就可以得到快速高效的乘法器, 这种方式实现的乘法器比现在所用的基于 FPGA 的乘法器的处理速度快 20%[7])。

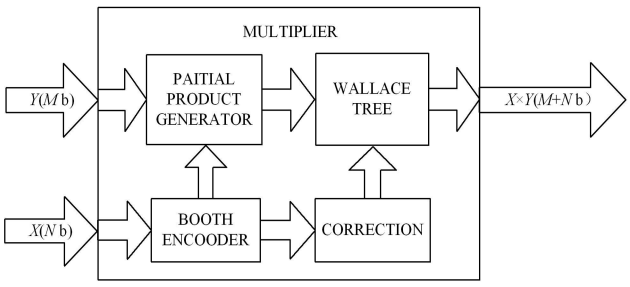


图 2　乘法器架构

(6) 亟待解决的问题

FPGA 凭借其如同软件实现一样的灵活性, 集合了硬件实现高效和并行性的优点, 好像非常适合神经实现的正常需要, 但是, FPGA 的二维拓扑结构不能处理标准神经网络规则但复杂的连线问题, 而且 FPGA 仍然实现很有限的逻辑门数目, 相反, 神经计算则需要相当耗费资源的模块(激活函数, 乘法器)。这样对于 FPGA, 可用的 CLBs 中部分将被用来增加路径容量(连线), 导致计算资源的丢失。一般的方法只能实现很小的低精度神经网络, 连线问题不能依靠几个具有比特序列算法的可重构 FPGA 以及小面积模块(随机比特流或者频率)解决。

## 2　基于 FPGA 的 ANN 实现方法

经典实现方法有:

(1) 可重构的 RNN 结构(Reconfigurable Neural Network)

可重构计算是一种增加处理密度(每单元硅片面积的性能)的有效方法, 且处理密度远大于用于通用计算方法, FPGA 作为可重构计算的平台, 可以提供如同软件一样的设计灵活性。该方法基于可扩展的脉动阵列结构、可重用的 IP(Intellectual Properties)核及 FPGA 器件, 即将要实现的神经网络算法分为几种基本运算, 这些基本运算由可重构单元(Reconfigurable Cell, RC)完成, RC 间以规则的方式相互连接, 当神经网络变化时, 只要增减 RC 的数量或替换不同功能的 RC 就可重构成新的神经网络硬件; 文献[8]中同时指出, 考虑到硬件实现要以最少的硬件资源满足特定应用的性能需求, 一般用神经元并行作为可重构部件的基本模式, 即神经网络的各层计算可复用相同的阵列结构。

(2) RENCO 结构

可重构网络计算机(Reconfigurable Network Computer, RENCO)是一种用于逻辑设计原型或可重构系统的平台, 所设计的可重构系统对于工作在比特级的算法实现特别有效, 比如模式匹配。RENCO 的基本架构包括处理器、可重构部分(多为 FPGA)以及存储器和总线部分[9], Altera 公司提供的最新的 RENCO 在可重构部分包括近 100 万逻辑门, 足够实现高复杂度的处理器。具体参见文献[9]。尽管如此, 得到的可重构系统并非对所有的硬件实现都是优化的方法, 比如不适合于浮点运算。

(3) 随机比特流方法

随机比特流(Stochastic Bit Streams)的方法是使用串行随机的方法实现一些运算操作, 目的是为了节约资源和充分利用神经网络的实时性。随机算法的提出源于它的简易性, 基本原理即首先将所有的输入转换成二进制随机比特流, 就是任意化; 然后, 由数字电路组成的随机算法实现取代正常的算法; 最后, 随机比特流转回到正常的数值(文献[10]中有详细总结)。随机算法提供一种方法, 用简单的硬件实现复杂的计算, 同时又不失灵活性, 而且随机实现又与现代 VLSI 设计和生产

技术兼容。

FPNA 实现方法:

凭借着简化的拓扑结构和独特的数据交换流图, FPNA(Field Programmable Neural Arrays) 成功地解决了以简单的硬件拓扑结构有效地实现复杂的神经架构问题, 是一种特别适合 FPGA 直接实现的神经计算范例。FPNA 基于一种类似 FPGA 的结构: 它包含一系列可以自由配置的资源, 这些神经资源被定义用来实现标准神经元的计算功能, 但是它们是一种自主的方式, 这样通过有限的连接可以创造出许多虚拟的连线。利用这种新的神经计算理念, 一个标准的但结构复杂的神经网络可以由一个简化的神经网络替代( 文献[ 11] 给出了详细的数学表示和说明) 。

为了有个直观的理解, 图 3( a) 表示一个简单的 MLP 结构; 图 3( b) 说明通过节点间的直接连接建立虚拟连接。

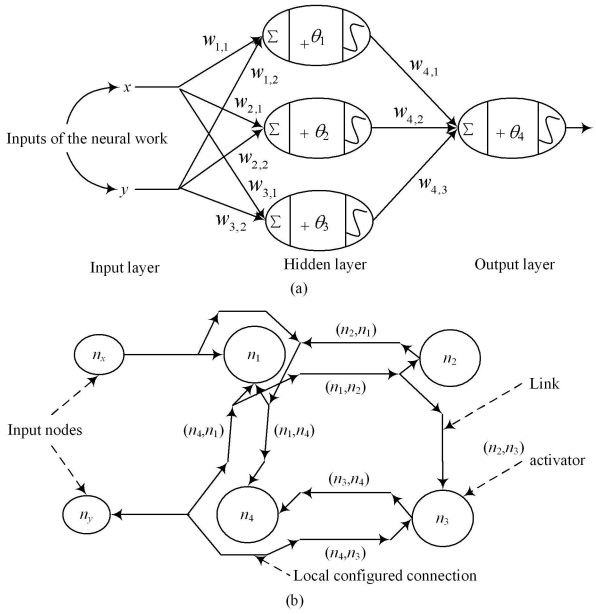


图 3 MLP 及虚拟连接

文献[ 11] 中的例证表明FPNA 计算范例确实允许一系列给定的神经资源代替具有不同架构的标准神经网络。然而, 从图 4 中可能并非如此, MLP 架构并没有得到简化, 原因在于如此简单的 MLP 完全没有必要, 也不可能再简化。文献[ 12] 描述了大型神经网络得到明显简化的实例。需要注意的是, FPNA 是一个适应神经计算的硬件框架, 而不是一种处理简化神经计算的实现方法[11] (Field Programmable Neural Network, FPNN)。要设计一个 FPNA, 首先要选择一个针对应用的合适的标准神经架构, 然后决定一个既适合于实现又在功能上等价于所选择神经网络的可配置 FPNA, FPNA 独特的计算方案在于在复杂神经网络和可用的硬件资源之间创造了一座桥梁, 它适用于许多实现选择; 最后, 得到的 FPNA 直接映射到硬件设备上, 这将得益于完整的模块式实现, 即对于每个神经资源, 预先给定可配置模块, 然后依照 FPNA 硬件友好的架构进行组合。

## 3 基于 FPGA 的神经网络的性能评估及局限性

对于 FPGA 实现的 ANN, 最普遍的性能评估方法是每秒神经元乘累加的次数(Connections-Per-Second, CPS) 和即每秒权值更新的次数( Connections-Updates-Per-Second, CPUS)。但是 CPS 和 CPUS 并不是适于所有的网络, 如 RBF 径向基网络, 另外, 更大的 CPS 和 CPUS 值并不一定意味着更好的性能。因此, 最好的性能测量方法是实际执行时间, 但是仍有些问题要讨论[13]。FPGA 实现神经网络存在的一些缺点(相对于计算机软件而言):

(1) FPGA 上实现的神经网络大多数是计算结构, 而不是认知结构(虽然现在有些人试图在 FPGA 上实现 BP 算法, 但是整个的结构和时序控制变得很复杂, 并且无法达到计算机软件那样的计算精度) ;

(2) 在 FPGA 上实现的神经网络通用性差。目前 FPGA 的使用者大多数都是在 RTL 级(寄存器传输级) 编写 VHDL/Verilog HDL 实现数字系统, 而正在兴起的 Handel-C & SystemC, 可以使硬件编程者站在算法级角度, 可能对以后的基于 FPGA 的神经网络的性能有所改善。

## 4 基于 FPGA 实现神经网络的发展方向

(1) 一种基于 REMAP-β 实现神经网络计算机的方法。REMAP-β 可重构架构基于 FPGA 技术, REMAP-β并行计算机应用在嵌入式实时系统中, 以有效提高 ANN 算法实现的效率, 目前它的进一步发展 REMAP-$r$ 正在探讨中[14]。

(2) 另一种基于 FPGA 实现神经网络的发展方向——系统 C 语言, 直接在可编程硬件平台支持 C/C++, 使得编程更加容易。但是这个转换并不容易, 因为 FPGA 不是程序, 而是电路[15]。

## 5 结 语

详细总结了 FPGA 实现神经网络的方法及相关问题, 这里要注意, 基于 FPGA 实现神经网络, 并不是要与基于计算机软件实现一比高低, 相反, 在很多情况下, 采用计算机软件测试神经网络的收敛情况, 计算出收敛时的权值, 然后通过数据口线与 FPGA 模块通信, 把权值交给 FPGA 中的神经网络, 使用 FPGA 完成现实的工作。直到现在, 软件方法仍然是实现神经网络的首

选。另外, 对于硬件设计者( 指利用 FPGA 或者全定制、半定制 ASIC 实现设计) 而言, mask ASICs 提供首选的方法以得到大规模、快速和完全的神经网络。现在它已经开发出了所有的新型可编程器件的嵌入式资源, 以得到可以实时训练的更有用的神经网络。

## 参 考 文 献


[1] Medhat Moussa, Shawki Areibi, Kristian Nichols. On the Arithmetic Precision for Impleentation Back-propagation Network on FPGA: A Case Study, University of Guelph, Canada, 2005.

[2] Jordan L Holt, Thomas E Baker. Back Propagation Simulations Using Limited Precision Calculations[A]. Proceedings International Joint Conference on Neural Networks[C]. Seattle, USA, 1991.

[3] Amos R Omondi, Jagath C Rajapakse, Mariusz Bajger. FPGA Implementations of Neural Networks[J]. IEEE Trans. on Neural Networks, 2007, 18(5): 1550-1550.

[4] Jihan Zhu, Peter Sutton. FPGA Implementations of Neural Networks: A Survey of a Decade of Progress[M]. Berlin: Springer, 2003.

[5] Wolf D F, Romero R A F, Marques E. Using Embedded Processors in Hardware Models of Artificial Neural Networks [A]. proceedings of SBAI[C]. 2001: 78-83.

[6] Pedro Ferreira, Pedro Ribeiro, Ana Antunes, *et al*. A High bit Resolution FPGA Implementation of a FNN with a New Algorithm for the Activation Function Science Direct[J]. Neurocomputing, 2007: 71-77.

[7] Suthikshn Kumar, Kevin Forward, Palaniswami M. A Fast-Multiplier Generator for FPGAs[A]. 8th International Conference on VLSI Design[C]. 1995.

[8] 李昂, 王沁, 李占才, 等. 基于 FPGA 的神经网络硬件实现方法[J]. 北京科技大学学报, 2007, 29(1): 90-95.

[9] Jearr Luc Beuchat, Jacques-Olivier Haenni. Hardware Reconfigurable Neural Networks[M]. Berlin: Springer, 1998.

[10] Kuno Kollmann, Karl-Ragmar Riemschneider, Hans Christoph Zeidler. On-chip Backpropagation Training Using Parallel Stochastic Bit Streams[A]. Proceedings of 5th International Microelectronics for Neural Networks [C]. 1996: 149-156.

[11] Bernard Girau. FPNA: Concepts and Properties[M]. US: Springer, 2006.

[12] Bernard Girau. FPNA: Application and Implementations [M]. US: Springer, 2006.

[13] Amos R Omondi, Jagath C Rajapakse, Mariusz Bajger. FPGA Neurocomputers[J]. FPGA Implementations of Neural Networks, 2003: 1-36.

[14] Lars Bengtsson, Arne Linde, Tomas Nordstrom, *et al*. The Remap Reconfigurable Architecture: A Retrospective[J]. FPGA Implementations of Neural Networks, 2005: 325-360.

[15] Tan W H, Thiagarajan P S, Wong W F, *et al*. Synthesizable System C Code from UML Models. National University of Singapore, 2004.

[16] Alanf Murray, Anthony V W Smith. Asynchronous VLSI Neural Networks Using Pulse-Stream Arithmetic[J]. IEEE Journal of Solid-state Circuits, 1998, 23(3): 688-697.



作者简介　杨银涛　男, 1986 年出生。主要研究方向为 FPGA 应用及 IC 设计。
汪海波　男, 1987 年出生。主要研究方向为智能信号处理与信息融合。
张　志　男, 1987 年出生。主要研究方向为嵌入式系统。
周建华　女, 1965 年出生, 副教授, 硕士生导师。主要研究方向光电子技术。


## 互联网电视符合产业发展方向

近日, 国家广电总局发布《关于加强以电视机为接收终端的互联网视听节目服务管理有关问题的通知》, 旨在进一步维护著作权的合法权益, 规范互联网视听节目传播秩序。政府部门在加强对互联网电视内容监管的同时, 应该为互联网电视的发展进一步创造有利的市场环境。

首先, 互联网电视可以为人民群众多样化的精神文化需求提供服务。党的十七大报告明确指出, 要在时代的高起点上推动文化内容形式、体制机制、传播手段创新, 解放和发展文化生产力; 要运用高新技术, 创新文化生产方式, 培育新的文化业态, 加快构建传输快捷、覆盖广泛的文化传播体系。互联网电视正是迎合了这一需要而逐渐发展起来的一种传播手段, 它能够为满足人民群众新时期多样化、多层次、个性化的精神文化需求服务。从市场反映情况来看, 近两年来我国彩电骨干企业推出的互联网电视越来越受到广大用户的欢迎。

(摘自 中国电子报)