第33卷 第19期 计 算 机 工 程 2007年10月
Vol.33 No.19 Computer Engineering October 2007
·工程应用技术与实现· 文章编号：1000—3428(2007)19—0236—02 文献标识码：A 中图分类号：TP393

一种基于FPGA的SOM神经网络算法的并行实现

孔 超，李占才，王 沁，李 昂，钱 艺
(北京科技大学信息工程学院，北京 100083)

摘 要：分析了SOM神经网络算法在FPGA实现过程中要考虑的2个主要问题：并行性和有限字长效应。通过分析，提出了一种实现该算法的高并行体系结构并给出了该体系结构中关键模块的具体实现电路。根据计算机仿真以及在FPGA上的实现所得到的结果表明，该体系结构在保证神经网络性能的同时，可以使电路具有较高的处理速度。

关键词：SOM；FPGA；并行；有限字长效应

FPGA-based Parallel Implementation of SOM Neural Network Algorithm

KONG Chao, LI Zhan-cai, WANG Qin, LI Ang, QIAN Yi
(School of Information Engineering, University of Science & Technology Beijing, Beijing 100083)

【Abstract】 This paper analyzes 2 critical problems in FPGA-based implementation of SOM neural network algorithm: parallelizability and finite word-length effect. The analysis of the two problems, this paper presents a highly parallel architecture to implement this algorithm and shows the circuit of critical blocks. By the result of computer simulation and FPGA implementation, shows that the architecture can ensure the quality of the neural network and meanwhile make the circuit have a high computing speed.

【Key words】 SOM; FPGA; parallelizability; finite word-length effect

SOM(self-organized feature mapping)学习算法是一种无监督的神经网络学习算法，在智能控制、编码和聚类方面有着广泛的应用。其实现方式可分为软件方式和硬件方式。在某些对数据实时处理要求很高的场合（如工业控制领域），由于软件串行计算的特点，其处理速度往往不能满足实时性的要求，因此须用硬件方式来实现神经网络。FPGA内部含有丰富的硬件可编程逻辑资源，可以利用FPGA搭建专用的硬件电路通过并行计算的方式来实现SOM神经网络，从而加快数据处理速度，以满足计算实时性要求。

用FPGA实现SOM神经网络，一方面需要考虑设计何种电路结构来充分实现并行计算，加快数据处理速度；另一方面需要考虑有限字长效应，不同的字长会对电路的性能（面积、速度）和SOM神经网络算法的性能（收敛性与准确性）产生影响。对于第1个问题，本文对SOM神经网络算法进行并行性分析，充分挖掘算法内在的并行性，从而提出适于FPGA实现的并行计算体系结构；对于第2个问题，本文利用统计分析的方法对有限字长效应进行分析，在对电路性能与神经网络算法性能之间折衷后确定字长。

本文以输入层神经元个数为4，Kohonen层神经元个数为6×6的SOM神经网络为例，对其进行分析。

1 SOM算法并行性分析

在FPGA中引入并行处理方式可以大大提高FPGA的计算能力，在串行算法中检测并行性的方法就是看其中的操作是否能独立执行，当计算之间存在数据相关时将无法并行计算。为了在FPGA中并行实现SOM算法，需要对算法进行并行性分析。

SOM学习算法步骤如下：

(1)初始化。对初试权值$[W_{ji}]$赋予[0,1]区间随机值，确定学习速率$\eta(t)$ 的初值η_0。确定临域$Ng(t)$的初值$Ng(0)$及总学习次数T。

(2)取样。从输入空间取样本$\boldsymbol{U}_k$。

(3)计算连接权矢量$\boldsymbol{W}_j$与输入矢量$\boldsymbol{U}_k$之间的欧式距离。

$$d_j=[\sum_{i=1}^{n}\left(U_i^{k'}-W_{ji}^{'}\right)^2]$$

(4)找出最小的欧式距离d_g，并确定获胜神经元g。

$$d_g=\min\left[d_j\right]$$

(5)判断神经元是否在获胜神经元的临域内，如果$\left(x_{win}-x_i\right)^2+\left(y_{win}-y_i\right)^2\le Ng(t)^2$，则表明该神经元在获胜神经元临域之内，其中，$x_{win}$，$y_{win}$为Kohonen获胜层神经元坐标；$x_i$，$y_i$为Kohonen层神经元坐标。

(6)调整连接权，对竞争层临域$Ng(t)$内所有神经元与输入层间的连接权进行调整。

$$W_{ji}^{'}(t+1)=\begin{cases}W_{ji}^{'}(t+1)+\eta(t)\left[U_i^{k'}-W_{ji}^{'}(t)\right] & \mathrm{j}\in Ng(t)\\ W_{ji}^{'}(t) & \text{其他}\end{cases}$$

(7)更新学习速率$\eta(t)$及临域$Ng(t)$。

$$\eta(t)=\left[\eta(0)-\eta(T)\right](1-t/T)+\eta(T)$$

$$r(t)=r(0)+\left[1-r(0)\right]t/T$$

(8)令t=t+1，返回步骤(2)，直到t=T为止。

首先分析各步骤之间的可并行性。由于步骤(1)和步骤(2)所执行操作互不相干，因此这2步可并行执行。步骤(4)可以将每一个由步骤(3)计算得到的结果d_j立即与步骤(3)的上一计

作者简介：孔 超(1982－)，男，硕士研究生，主研方向：计算机体系结构，IC设计；李占才，副教授；王 沁，教授、博士生导师；李 昂、钱 艺，博士研究生

收稿日期：2006-10-15 **E-mail：**kc1982@163.com

算结果d_{j-1}进行比较，而不是完全遵从与原算法将所有的距离计算完成后再对数据进行比较，从而实现步骤(3)与步骤(4)的并行计算。步骤(5)和步骤(6)的并行计算可以通过在计算更新权值的同时，计算Kohonen神经元与获胜神经元之间的距离并判定该神经元是否在获胜神经元的临域。步骤(6)计算得到的更新权值是否代替原来的权值要受步骤(5)的计算结果控制。然而该算法必须在执行完步骤(3)、步骤(4)确定获胜神经元后，步骤(5)、步骤(6)才能确定哪些神经元的权值需要更新，因而步骤(3)、步骤(4)与步骤(5)、步骤(6)之间存有数据依赖关系，二者无法同时并行计算。对于步骤(7)，该步骤的计算量较小，而且与以前的步骤不存在数据相关性，因此，可以在执行其他步骤的同时执行步骤(7)。电路具体并行计算过程见表 1。

表 1　电路并行计算过程

时间步	执行步骤	
1	步骤(1)，步骤(2)	
2	步骤(3)，步骤(4)	
3	步骤(5)，步骤(6)	步骤(7)
4	步骤(2)	
5	步骤(3)，步骤(4)	
6	步骤(5)，步骤(6)	
…	…	

然后分析步骤之内运算的可并行性。步骤 3~骤 6 中存在着大量的向量与矩阵的减、加、乘以及比较运算，并且各算术运算之间不存有数据依赖关系，因此，可以在各算术运算之间加入寄存器从而实现流水计算，加快数据处理速度。

2　字长效应分析

由于在工程的应用中经常会涉及到中大规模的神经网络[1]，对于大规模的神经网络，不可避免地会使用到低精度带有误差的神经元。高的精度，即选取较长的字长意味着牺牲了集成电路的性能（面积、成本以及计算速度）而获得了较好的神经网络性能（准确性和收敛性）。低的精度则会对集成电路的性能以及神经网络的性能产生相反的影响。因此，在选取字长时需要对二者权衡考虑。

根据SOM神经网络学习算法的特点，输入量化误差以及乘法截尾误差是对SOM神经网络有限字长效应的来源。可以使用有限字长神经网络实现研究理论中常用的统计分析方法[2]，针对特定规模的SOM神经网络分析有限字长对其性能的影响，从而确定神经网络运算部件的字长。

本文通过对输入层神经元个数为 4，Kohonen层神经元个数为 6×6 的神经网络，分别进行浮点运算学习（32 位及其以上几乎可视为无限精度的，即忽略计算机本身在运算和存储中的有限字长效应[2]）和有限精度下不同字长的定点运算学习。仿真过程中先从输入空间选取 4 000 个随机向量对网络进行训练，训练结束后再从输入空间中随机选取 1 000 个样本对其进行聚类。仿真结果见表 2。该结果表明针对该神经网络，当字长到达 18 时会达到比较好的学习效果。

表 2　同字长对神经网络性能影响的仿真结果

字长	15	16	17	18	19
权值均方误差	0.359 38	0.381 18	0.359 89	0.0000 30	0.000 22
聚类正确率/%	73.7	64.6	75.2	100	100

3　电路结构

通过上述的分析，在本节中将以输入层神经元个数为 4，Kohonen 层神经元个数为 6×6 的 SOM 神经网络为例来设计实现该算法的电路。该电路的字长选为 18，整体结构如图 1。

电路完成一次学习迭代的过程如下：首先从外部输入空间中读取向量“ Input ”并依次从“ 权值 RAM1~权值 RAM4 ”中读取权值向量至“ 距离计算/权值更新 ”模块，“ 距离计算/权值更新 ”模块将计算所得输入与各 Kohonen 层神经元之间距离依次送入“ 最短距离判定/临域判定 ”模块。“ 最短距离判定/临域判定 ”模块在判定输入与各 Kohonen 层神经元之间最短距离完成后，“ 距离计算/权值更新 ”模块将计算更新权值，同时“ 与获胜神经元距离计算 ”模块开始工作，其计算结果送至“ 最短距离判定/临域判定 ”模块中，当该模块所得的结果判定神经元在获胜神经元临域内时，计算好的更新权值将被写入“ 权值 RAM1~权值 RAM4 ”，完成一次迭代过程。“ 学习速率与临域更新 ”模块在迭代的开始即执行，在迭代结束时完成学习速率与临域的更新。

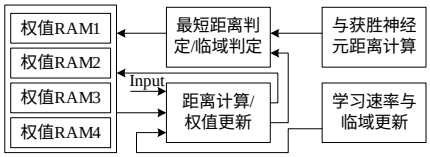

图 1　电路总体结构

这里只介绍其中比较重要的“ 距离计算/权值更新 ”模块和“ 最短距离判定/临域判定 ”模块。

3.1　距离计算与权值更新模块

由第 1 节可以得到，步骤(3)和步骤(6)都是由减法、乘法、加法 3 种运算构成的，而且步骤(3)和步骤(6)在时间上不会同时执行，因此，可以将减法器、乘法器和加法器 3 种运算部件复用。同时为了加快数据处理速度，在各运算部件之间加入寄存器从而引入流水线。

该部分数据通路如图 2 所示。在步骤(3)时用于计算输入向量与权值矩阵的距离；步骤(6)时用于计算更新权值，为保证流水效率，将更新权值的计算与判定是否更新权值运算并行执行，由判定结果决定该更新权值的计算结果是否写入权值 Ram 中。

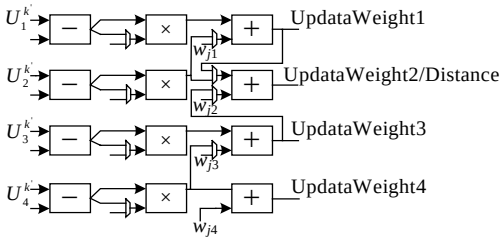

图 2　距离计算与权值更新模块

这种结构，通过复用运算部件，可以有效降低 FPGA 内部的资源使用率。同时，综合后结果表明选通器的引入不会改变整个 FPGA 电路的关键路径，即该模块的运算部件复用结构不会降低整个电路的最高时钟频率。

3.2　距离判定与临域判定模块

同样，依据第 1 节的分析，步骤(4)和步骤(5)的运算部件主要是由比较器组成，为了节省 FPGA 内部资源，将比较器复用。该模块具体电路见图 3。

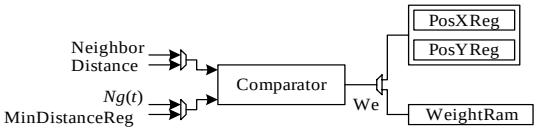

图 3　短距离判定与临域判定模块

（下转第 240 页）