

一种基于 FPGA 的新的 SVM 硬件实现方法^{*}

刘 涵, 王 博, 尹 嵩, 刘 丁

(西安理工大学自动化与信息工程学院 西安 710048)

摘 要: 提出了一种基于递归神经网络的实现最小二乘支持向量机的 FPGA 串行计算方法, 与已有的并行计算方法相比, 该方法利用了递归神经网络的并行性及最小二乘支持向量机简化的约束条件的优点, 在保证计算速度的同时, 明显提高了 FPGA 的硬件资源利用效率, 能够适应大规模训练样本的情况。实验结果表明, 由于该方法具有灵活的串行计算、并行传输的特点, 在较少使用 FPGA 硬件资源的同时, 计算速度不会有明显变化, 可有效地用硬件实现支持向量机。

关键词: 串行计算方法; 最小二乘支持向量机; 递归神经网络

中图分类号: TN911.7 **文献标识码:** A **国家标准学科分类代码:** 520.2040

New serial implementation method for support vector machines using FPGA

Liu Han, Wang Bo, Yin Song, Liu Ding

(School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China)

Abstract: A new FPGA serial computational method for implementing least squares support vector machines based on recurrent neural network is presented in this paper. Compared with existing parallel computational method, the new method combines the parallel character of recurrent neural network with simplicity of least squares support vector machine, can get a good performance with less hardware resources while maintaining the computational speed, and also can adapt to large scale training samples. Experiment results show that due to the flexible serial computational and parallel transmission characteristics, the consumption of FPGA space could be reduced effectively while the computing speed will not drop obviously. In conclusion, the function of support vector machines could be implemented on FPGA platform using the proposed method effectively.

Key words: serial computational method; LS-SVM; recurrent neural network

1 引 言

支持向量机 (support vector machines, SVM) 作为 20 世纪 80 年代初由 Vapnik 提出的统计学习理论的主要内容^[1], 由于其良好的推广能力以及在模式识别领域成功的应用^[2], 越来越被人们所接受, 已成为机器学习研究的热点领域。

对 SVM 的求解需要求解二次凸优化问题, 其计算复杂度会随着样本集合的增大迅速增加。在许多工程应用

领域, 往往需要用 SVM 对数据进行实时处理, 因此实现 SVM 的并行和分布式计算显得十分必要。文献 [3] 将支持向量机的训练问题转化为递归神经网络的计算问题, 在此基础上文献 [4-12] 分别完成了基于硬件电路的支持向量机和最小二乘支持向量机 (least squares support vector machines, LS-SVM) 的并行计算结构。随着 FPGA (field programmable gate array) 技术的发展, FPGA 的并行性和快速重构的特点也被应用于实现神经网络结构, FPGA 已逐渐成为实现人工神经网络的一种有效方式^[5-7]。在 FPGA 中采用 VHDL 语言来描述神经网络结构, 具有较

收稿日期: 2010-04 Received Date: 2010-04

*基金项目: 陕西省教育厅专项基金 (09JK632) 资助项目

©1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. <http://www.cnki.net>

强的灵活性,并且可以较好地保留神经网络的并行性,既保证了处理速度,又适合于实现大规模复杂的网络结构。

本文提出了基于递归神经网络的 LS-SVM的 FPGA 串行计算方法,相比已有的并行实现方法^[8],该方法具有灵活的串行计算、并行传输的特点,可以在大幅减少 FPGA使用资源的同时,保证计算速度,可有效地硬件实现 LS-SVM

2 基于递归神经网络的 LS-SVM计算方法

在 SVM求解过程中的凸优化问题需要通过二次规划方法来解决,而求解二次规划问题需要计算核函数矩阵,其计算量与训练样本数的立方成正比。当处理大规模数据时,计算量比较大,内存占用量较多。LS-SVM可以在一定程度上解决 SVM的计算复杂性的问题,它将 SVM优化问题的不等式约束变为等式约束,训练 LS-SVM仅仅要求解一个线性方程组,因此 LS-SVM具有更简单的结构,更易于实现^[9-14],在使用 FPGA实现时,能更有效地利用片内资源。

对于 LS-SVM的分类问题,设给定分类训练集 (x_i, y_i) $i=1, 2, \dots, N$ 其中 $x_i \in R^n$ 为训练样本,而 $y_i \in \{-1, +1\}$ 为样本相对应的类别,其分类决策面可以表示为:

$$f(x) = w^T \cdot \varphi(x) + b$$

式中: w 为权值矩阵, b 为偏移量, $\varphi(\cdot)$ 表示从输入空间到特征空间的非线性映射。LS-SVM分类学习即为求解下面的受约束的最优化问题:

$$\min_{w, b, \xi} J(w, \xi) = \frac{1}{2} w^T \cdot w + \gamma \frac{1}{2} \sum_{i=1}^N \xi_i$$

$$s.t. \quad y_i [w^T \varphi(x_i) + b] = 1 - \xi_i \quad i=1, 2, \dots, N$$

式中: ξ_i 为误差,常数 $\gamma > 0$,它控制对超出误差的样本的惩罚的程度。具有分类功能的 LS-SVM的神经网络结构如图 1 所示^[4]。

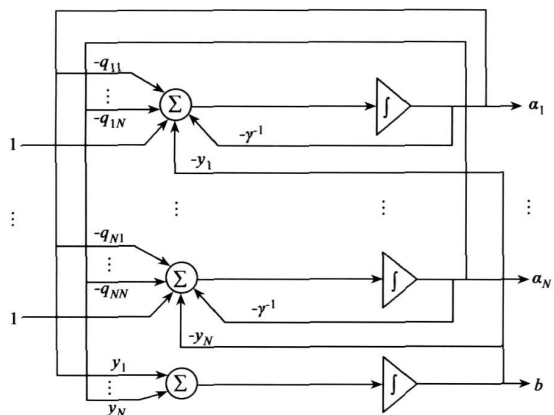


图 1 实现最小二乘支持向量机功能的递归神经网络结构图
Fig. 1 Scheme of the proposed recurrent neural network

for LS-SVM classification learning

该网络的连续动态方程如下:

$$\dot{b} = \frac{\partial J}{\partial b} = \sum_{i=1}^N \alpha_i y_i \quad (1)$$

$$\dot{\alpha}_i = -\frac{\partial J}{\partial \alpha_i} = 1 - b y_i - \sum_{j=1}^N \alpha_j q_{ij} - \gamma^{-1} \alpha_i \quad (2)$$

式中: α_i 为 Lagrange 乘子; $q_{ij} = y_i y_j K_{ij}$, $K_{ij} = \Phi(x_i)^T \Phi(x_j)$ 定义为核函数。可以看出,与 SVM的计算结构相比较^[3],该 LS-SVM结构没有非线性单元。由于 FPGA不能直接实现一个连续的微分方程,所以必须对式(1)、(2)进行离散化处理:

$$b(t + \Delta T) = \Delta T \sum_{i=1}^N \alpha_i(t) y_i(t) + b(t) \quad (3)$$

$$\alpha_i(t + \Delta T) = \Delta T [1 - b(t) y_i - \sum_{j=1}^N \alpha_j(t) q_{ij} - \gamma^{-1} \alpha_i(t)] + \alpha_i(t) \quad (4)$$

式中: ΔT 为采样周期。图 1 所示递归神经网络的渐进稳定性的证明详见文献[4-11]。

3 SVM的 FPGA硬件实现方法

3.1 串行计算结构的实现方法

在实现带约束的二次优化问题(3)、(4)时,文献[15]提出了 FPGA环境下的并行计算结构与实现方法,其核心计算部分如图 2 所示,图 3 为图 2 中的 neuro 单元结构,其中 $Q = [q_{ij}]$ 。

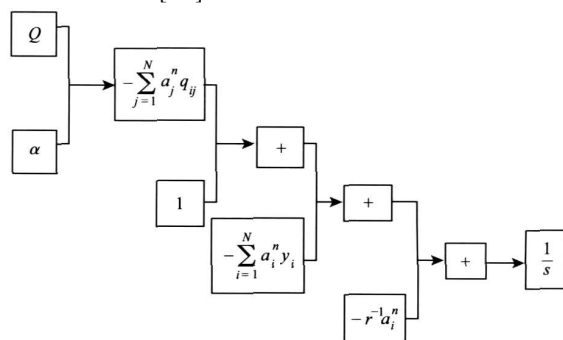


图 2 并行计算结构图

Fig. 2 The parallel computing structure chart for SVM learning

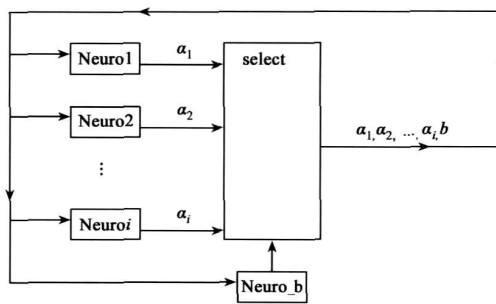


图 3 并行计算结构各个神经元内部结构

Fig. 3 The internal structure of neurons

此算法的思想是并行计算 m 个 α , 同时依据 α 值计算 b 值, 这时计算中所占用的硬件资源与样本数 m 有着直接的关系。当样本数增加时, 该结构所实现的 LS-SVM 将大量消耗 FPGA 的资源, 并且使结构的复杂程度增加。假设需要的样本数为 m 个, 对每个样本进行训练需要占用的 FPGA 资源为 k_0 , 那么 m 个样本总共要占用 FPGA 里 $m \times k_0$ 的资源, 这样很容易造成资源利用超出实际资源总量的问题, 将大大限制其在实际中的应用。

因此, 本文对图 1 结构进行优化和改变, 使其以串行结构的方式计算, 如图 4 所示。该串行结构避免了实际电路与样本数的直接关联, 可以在不改变实际电路的情况下完成实际样本数未定的训练工作, 其核心算法结构如图 5 所示。由图 5 可看出, 每次先循环计算 n 个 α 值后再计算 b 值, 避免了原来建立多个单元训练 α 值造成的硬件资源浪费, 因为本次计算的 α 值只与上个周期计算的 n 个 α 值有关。当然, 需要根据 FPGA 的资源来设定一个最大的训练样本数, 只要训练样本数不超过这个设定的范围, 通过串行的计算方法, 反复利用同一单元, 使得 FPGA 硬件资源的使用减少。

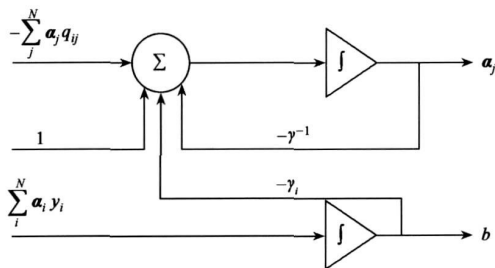


图 4 实现最小二乘支持向量机功能的串行计算结构图
Fig. 4 Serial computational structure scheme for LS-SVM classification learning

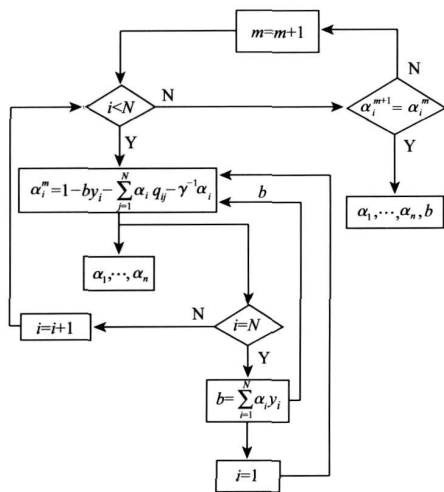


图 5 串行计算结构流程图
Fig. 5 Serial computation flowchart

图 4 的串行计算结构和并行计算结构相比, 学习速率会有所降低。因此需要将数据的传输与计算分开设计, 将计算与传输流水化, 在计算的同时将上一步的计算结果传递给其它模块, 这时在总体上采用串行计算结构的速度并不会受明显影响。

3.2 串行计算结构的 FPGA 实现

3.2.1 总体框图

图 6 为 LS-SVM 的 FPGA 实现串行计算框图, 整个框图的结构可以分为 3 个基础的功能单元。

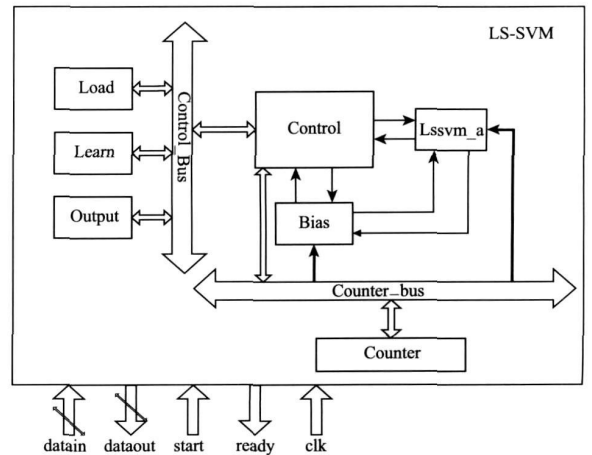


图 6 LS-SVM 的 FPGA 实现串行计算框图

Fig 6 Serial computational structure for LS-SVM based on FPGA

1) Load 单元: 在整个 LS-SVM 开始学习时, Start 端口的信号变成高电平, 首先通过 datain 端口输入矩阵 Q 和样本目标值 y , LS-SVM 将它们保存在 2 个 ROM 中。实际上, 为了操作方便并具有针对性, 直接把矩阵 Q 保存在一个 ROM 中, 方便随时读取。这里矩阵 Q 的数据必须严格按照行或按列寄存在 ROM 里。

2) Learn 单元: 在 Load 单元执行完毕后, 开始执行 Learn 单元。Learn 单元是整个 LS-SVM 的核心部分, 完成计算 α 和 b 具体细节在下节中将给予介绍。在学习训练完毕后, ready 端口变为高电平, 进入下一个 Output 单元。

3) Output 单元: 在 Lssvm_a 和 bias 模块训练完毕后, 将训练好的 $b, \alpha_1, \dots, \alpha_n$ 输出。

以上 3 个基本功能单元主要由图 6 中的 Control, Counter, Lssvm_a 和 Bias 模块 4 个模块组成, 其中 Lssvm_a 完成 α 的计算, Bias 完成 b 值的计算。所有控制信号由 Control 模块通过 Control_bus 来传递给其他模块, 时钟信号通过 Counter_bus 来传递, 模块间数据由 Data_bus 传递, Address_bus 主要负责读取所有 ROM 及 RAM 的数据。

3.2.2 模块介绍

1) Lssvm_a 模块

该模块执行图 5 中算法的 α 部分, 其硬件结构如图 7 所示。首先矩阵 Q 保存在 ROM 中, MAC 为乘累加单元,

用来计算 $\sum_{j=1}^N \alpha_j(\gamma) q_j$, 其中权值要与 $\alpha_j(\gamma)$ 相对应。3 个乘法器 MUL 分别计算 $b_j \gamma$ 、 $\gamma^{-1} \alpha_i(\gamma)$ 和最后的 ΔT 与 $1 - b(\gamma) \gamma_i - \sum_{j=1}^N \alpha_j(\gamma) q_j - \gamma^{-1} \alpha_i(\gamma)$ 的乘积; ADD 为另一个累加单元, 由于对式 (3) 进行离散化处理后, 最后的积分单元变为了一个累加单元, 因此 ADD 用来将本次得出的 α_i^{n+1} 值及上一组得到的 α_i^n 值进行累加, 并将更新的值存入 RAM 单元中。由于计算和比较过程需要用到上一步的向量 α^n , 所以利用 RAM 和 RAM1 来保存本次计算的一组 α^{n+1} 值和上次计算的一组 α^n 值。为了方便比较和累加运算, 根据计算每个 α_i 值的周期, 将上一组的 α_i 值临时保存在 A_in 中。由于计算采用的是定点运算, 因此 shift 单元用来将数据限制在指定的范围内, 本文限制的范围为 $(-15, 15)$ 。在计算完第 $n+1$ 步 α 值后, 下来计算第 $n+1$ 步 α_{i+1} , 直到计算完 k 个 α 值 (k 为训练样本数), 这就完成了串行计算方式。

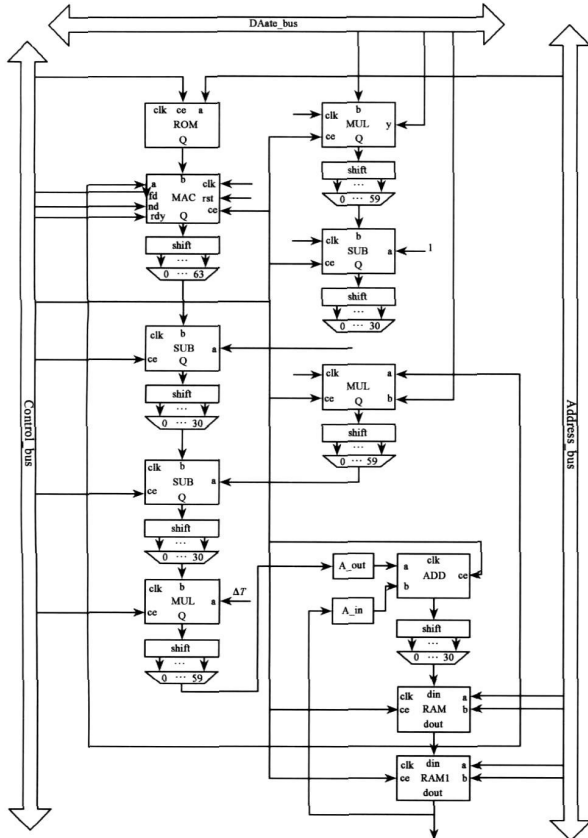


图 7 lssvm_a 模块结构图
Fig 7 The lssvm_a module

2) Bias 模块

Bias 模块用来计算 b 值, 并根据 Contro 模块的信号, 将更新的 b 值送入下一循环阶段的 Lssvm_a 模块中参与计算。

3) Contro 模块

Contro 模块是整个 LS-SVM 的控制模块, 它将控制各个模块并产生各模块所需的信号, 其控制信号并行传输给各个模块, 图 8 为 Contro 的内部状态机。

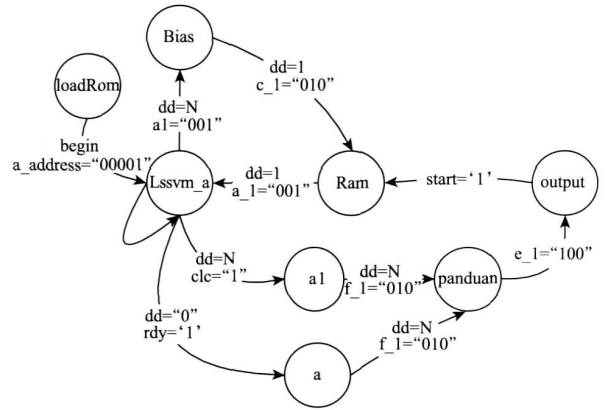


图 8 contro 模块状态机
Fig 8 FSM of the contro module

整个学习过程中, 需要输入运算矩阵 Q 以及样本目标值, 随后进入正式的训练学习过程, 先使用初始的 α 值和 b 值, 其分别存放于 Ram 和 Bias 中, 之后通过 Lssvm_a 对 α 值进行循环计算训练, 以计算完 n 个 α 值为一个周期, 在每个周期结束时计算 b 值, 之后对上次循环的 α 值和本次所得 α 值进行比较, 当 2 个周期的所有 α 值的误差小于允许的范围时, 得出所要的 α 值和 b 值。

4) Counte 模块

Counte 模块是整个运算的计数模块并产生时序, 其中的行列计数器分别产生读取行列的时序序列, 并通过 Contro 模块对 ROM 及 RAM 进行读取数据。Contro 模块也将根据 Counte 模块产生的选择信号进行各个功能单元之间的切换及工作。

4 实验结果及分析

根据上述提出的串行计算的 LS-SVM 结构, 本文通过 2 个实验来测试其性能, 并与文献 [15] 的并行计算结构进行比较。实验所用到的计算机配置为 Pentium(R) 4 CPU 3.20 GHz, 2.0 GHz, 1 Gb 内存。硬件平台为 XILINX 的 VirtexII Pro 系列 FPGA, 软件平台是 ISE 编译环境及 VHDL 硬件编程语言。

实验 1 在 R^2 空间上有 7 个线性可分的样本点:

$Z_1 = (3, 1)$, $Z_2 = (3, 2)$, $Z_3 = (3, 3)$, $Z_4 = (5, 1)$, $Z_5 = (5, 2)$, $Z_6 = (5, 3)$, $Z_7 = (5, 1.5)$, 相对应的类别为 $(+1, +1, +1, -1, -1, -1, -1)$ 。从直观上看, 显然最优超平面应该为 $x=4$ 。本例中选择 $\gamma^{-1} = 0.5$, 核函数选择高斯函数, 其中 $\sigma = 1$ 。

图 9 为参数 α 的 FPGA 串行结构实现结果, ram_initia₀ ~ ram_initia₆ 分别代表参数 $\alpha_0 \sim \alpha_6$ 。从图中可以

看出参数 α 的收敛时间为 3.182×10^{-4} s。图 10 为由 FPGA 串行结构得到参数 α 绘制的分类超平面。图 11 为参数 α 在应用并行结构时的结果, $\alpha_{1f} \sim \alpha_{16}$ 分别代表参数 $\alpha_0 \sim \alpha_6$, 从图中可以看出参数 α 的收敛时间为 2.346×10^{-4} s。图 12 为由 FPGA 并行结构得到参数 α 绘制的分类超平面。从图 10 和图 12 的结果比较可以看出, 串行结构的 LS-SVM 计算方法同样可以很好的解决实验 1 的分类问题, 从参数 α 的收敛时间看, 计算速度没有收到明显的影响, 因此这种方法对解决线性可分分类问题是可行的。

1.8 1.9 2.1 2.2 2.3 2.0 1.8 2.0], 相对应的类别为 (-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, +1, +1, +1, +1, +1, +1, +1, +1, +1, +1, +1, +1)。该样本点全部来自植物数据集 (iris data set) 中的不可分两类样本^[3]。

Current Simulation Time: 2.2899e-04 ms		318 200	318 400	318 600	318 800	319 000
■ Ram_mbr02 号	1	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr03 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr04 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr05 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr06 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr07 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr08 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr09 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr10 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr11 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr12 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr13 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr14 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr15 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr16 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F

图 9 实验 1 FPGA 串行计算结构计算结果

Fig. 9 The computation result of serial structure with FPGA in experiment 1

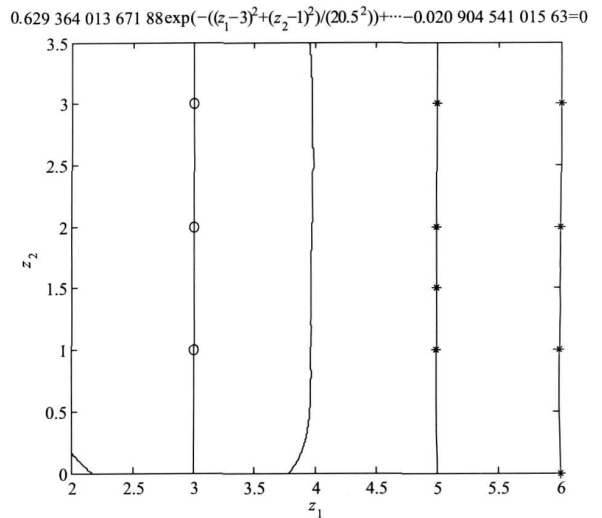


图 12 实验 1 的并行计算结构的分类情况
Fig. 12 Decision surface for parallel structure

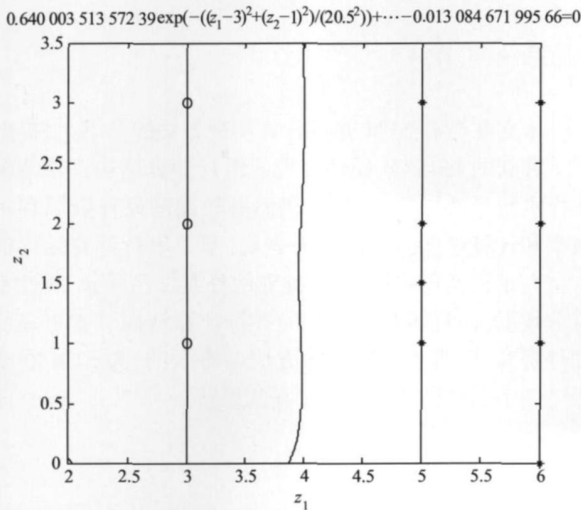


图 10 实验 1 的串行计算结构的分类情况

Fig. 10 Decision surface for serial structure

图 13 为参数 α 的串行计算结果, 参数 α 的收敛时间为 1.2948×10^{-3} s, 图 14 为参数 α 绘制的分类超平面。从图中可以看出两线性不可分类点已经被决策超平面分开。图 15 为参数 α 的并行计算结果, 参数 α 的收敛时间为 2.366×10^{-4} s, 图 16 为相应的参数 α 绘制的分类超平面。从图 14 和图 16 的结果比较可以看出, 对于线性不可分问题, 串行结构的计算方法也同样可以实现 LS-SVM 的分类功能。从参数 α 的收敛时间看, 计算速度没有受到明显的影响, 因此对于解决线性不可分分类问题也是可行的。

Current Simulation Time: 59.1927 ms		234 600	234 800	235 000	235 200	235 400
■ Ram_mbr02 号	1	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr03 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr04 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr05 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr06 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr07 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr08 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr09 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr10 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr11 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr12 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr13 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr14 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr15 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr16 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F

图 11 实验 1 FPGA 并行计算结构计算结果

Fig. 11 The computation result of parallel structure with FPGA in experiment 1

Current Simulation Time: 2.9601e-04 ms		1 294 800	1 295 000	1 295 200	1 295 400	1 295 600
■ Ram_mbr02 号	1	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr03 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr04 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr05 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr06 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr07 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr08 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr09 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr10 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr11 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr12 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr13 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr14 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr15 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F
■ Ram_mbr16 号	3	3F70000134B	3F70000134C	3F70000134D	3F70000134E	3F70000134F

图 13 实验 2 FPGA 串行计算结构计算结果

Fig. 13 The computation result of serial structure with FPGA in experiment 2

实验 2 选取 20 个线性不可分的样本点, $z = [x \ y]^T = [4.7 \ 4.5 \ 4.9 \ 4.0 \ 4.6 \ 3.3 \ 3.9 \ 3.5 \ 4.2 \ 5.1 \ 4.5 \ 6.3 \ 5.8 \ 5.3 \ 5.5 \ 6.7 \ 6.9 \ 4.9 \ 4.8 \ 6.4 \ 1.6 \ 1.5 \ 1.6 \ 1.3 \ 1.8 \ 1.0 \ 1.4 \ 1.0 \ 1.5 \ 1.8 \ 1.7 \ 1.8$

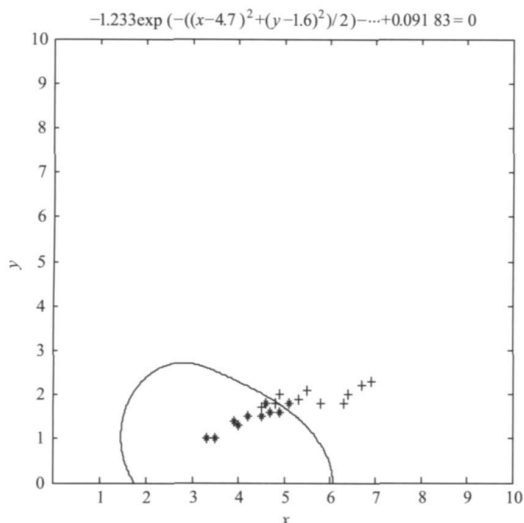


图 14 实验 2 的串行计算结构分类情况

Fig. 14 Decision surface for serial structure in experiment 2

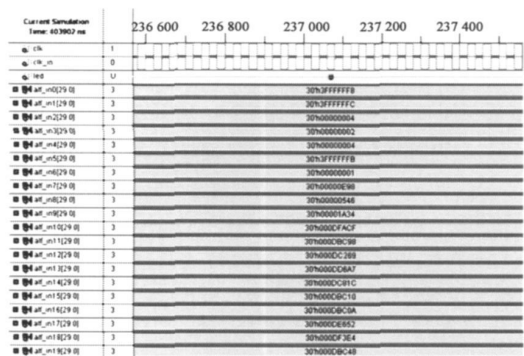


图 15 实验 2 FPGA 并行计算结构计算结果

Fig. 15 The computation result of parallel structure with FPGA in experiment 2

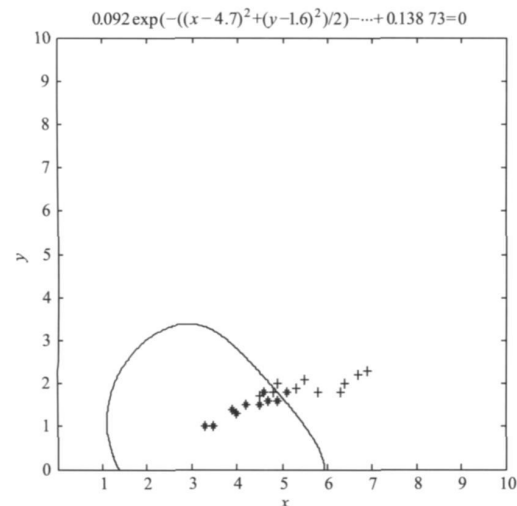


图 16 实验 2 的并行结构计算分类情况

Fig. 16 Decision surface for parallel structure in experiment 2

由表 1 可以看出,在实验 1 中 7 个训练样本时,串行计算结构的资源占用率很低(25%),而并行计算结构的资源利用已经超出了资源使用范围(103% WTBZ)。实验 2 中使用了 20 个训练样本,虽然样本数增加了很多,但由于串行计算结构的优势,其资源占用仍维持在较低水平(37%),而对于并行结构,其资源已远远超出了硬件适用范围,达到了 284% WTBZ。虽然串行结构在时间上要比并行结构稍慢,但其有效的提高了硬件资源利用效率。

表 1 两个实验的收敛时间及资源占用率

Table 1 Convergence time and resource allocation ratio in two experiments

实验手段		实验 1	实验 2
串行计算	收敛时间 / s	3.182×10^{-4}	1.2948×10^{-3}
	资源占用 (%)	25	37
并行计算	收敛时间 / s	2.346×10^{-4}	2.366×10^{-4}
	资源占用 (%)	103	284

5 结 论

本文在已有 SVM 并行计算实现方法的基础上,提出了一种新的 LS-SVM 硬件实现的串行计算结构。该结构在样本数量增加的情况下,仍然有较高的硬件资源利用效率和计算速度。实验结果表明,基于串行计算结构的支持向量机实现方法,能够准确地分类线性可分、线性不可分数据,而且能够比已有的并行计算结构节省更多的硬件资源,证明了所提出的方法是有效的,为 SVM 在实际应用中的硬件实现提供了新的思路。

参考文献

- [1] VAPNIK V. Statistical learning theory [M]. New York: Wiley Press, 1998.
- [2] CRISTIANINI N, SHAPE-TAYLOR J. An introduction to support vector machines [M]. Cambridge: Cambridge University Press, 2000.
- [3] PERFETTIERI R, RICCI E. Analog neural network for support vector machine learning [J]. IEEE Transactions on Neural Networks, 2006, 17(4): 1085-1091.
- [4] LU H, LU D. A neural network approach for least squares support vector machines learning [C]. 48th IEEE Conference on Decision and Control, Shanghai, China, 2009, 7297-7302.
- [5] COX C E, BLANZ E, CAGLIONE. A fast field programmable gate array implementation of a connectionist classifier [J]. IEEE Journal of Solid State Circuits, 1992, 28(3): 288-299.

- [6] BERTHELOT F NOUVEL F HOUSED D Partial and dynamic reconfiguration of FPGAs: A top down design methodology for an automatic implementation [J]. IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures Karlsruhe Germany 2006 436-437
- [7] 李海森, 陈宝伟, 么彬, 等. 多子阵高分辨海底地形探测算法及其 FPGA 和 DSP 阵列实现 [J]. 仪器仪表学报, 2010 31(2): 281-286
LI H S CHEN B W YAO B et al Implementation of high resolution sea bottom terrain detection method based on FPGA and DSP array [J]. Chinese Journal of Scientific Instrument 2010 31(2): 281-286
- [8] ANGUIA D BONIA R IDELLA S A digital architecture for support vector machines: Theory algorithm and FPGA implementation [J]. IEEE Transactions on Neural Networks 2003 14(5): 993-1009
- [9] MATHIAS M CHERIET M Model selection for the LS-SVM application to handwriting recognition [J]. Pattern Recognition 2009 42(12): 3264-3270
- [10] SUYKENS J A K VANDEWALLE J Recurrent least squares support vector machines [J]. IEEE Transactions on Circuits and Systems 2000 47(7): 1109-1114
- [11] 刘涵, 叶平. 基于递归神经网络的 LS-SVM 硬件实现与实验研究 [J]. 仪器仪表学报, 2009 30(8): 1745-1751
LIU H YE P Hardware implementation and experiment research of least square support vector machine based on recurrent neural network [J]. Chinese Journal of Scientific Instrument 2009 30(8): 1745-1751
- [12] SHEN M F LN L X CHEN J L et al A Prediction approach for multichannel EEG signals modeling using local wavelet SVM [J]. IEEE Transactions on Instrumentation and Measurement 2010 59(5): 1485-1492
- [13] ADANKON M M CHERIET M BIEM A Semi supervised least squares support vector machine [J]. IEEE Transactions on Neural Networks 2009 20(12): 1858-1870
- [14] 杨福刚, 孙同景. 基于最小二乘支持向量机的输液异物检测系统设计 [J]. 电子测量与仪器学报, 2008 22(14): 104-109
YANG F G SUN T J Design of Particle detection system for Pharmaceutical fluid based on least square support vector machines [J]. Journal of Electronic Measurement and Instrument 2008 22(14): 104-109
- [15] 尹嵩. 基于动态神经网络的支持向量机的 FPGA 实现 [D]. 西安: 西安理工大学, 2009
YIN S FPGA implementation of dynamic neural network for support vector machines [D]. Xi'an: Xi'an University of Technology 2009

作者简介



刘涵, 1993 年于西安理工大学获得学士学位, 1996 年和 2003 年在西安理工大学分别获得工学硕士和工学博士学位, 现为西安理工大学教授, 主要研究方向为复杂系统建模与控制。

E-mail: liuhan@xaut.edu.cn

Liu Han got B Sc in 1993, M Sc in 1996 and Ph D in 2003 all from Xi'an University of Technology. He is a professor in Xi'an University of Technology now. His main research field includes complex system identification and control.