

基于 GPU 的图像监督分类算法的研究

李思瑶 周海芳 方民权

(国防科技大学计算机学院 长沙 410073)

摘要 文中介绍了 3 种经典的图像分类算法在 GPU 上的实现,分别是简单贝叶斯分类、KNN、SNN 分类。GPU 与 CPU 协同处理是目前使用得较多的结构模式。一般在 GPU 上执行计算量比较大的程序,CPU 负责指挥协调。文中对这 3 种算法进行了测试,通过实验分析,3 种算法的 GPU 并行程序分别获得了平均 72.472,149.536,125.39 倍的加速效果。使用的 GPU 架构是 Tesla k20c。贝叶斯、KNN 和 SNN 算法是监督分类算法。实验给出了 3 种算法图像处理的结果和时间,其均符合要求。

关键词 监督学习算法,KNN,SNN,简单贝叶斯算法

中图分类号 TP391 文献标识码 A

Research of Image Classification Algorithm Based on GPU

LI Si-yao ZHOU Hai-fang FANG Min-quan

(School of Computer Science,National University of Defense Technology,Changsha 410073,China)

Abstract This paper introduced three classical image classification algorithms based on GPU, which are Bayes, KNN and SNN. Coprocessing of GPU and CPU is a structure pattern used frequently. Programs with large amount of calculation are working on the GPU, and CPU is used to control. This paper tested the programs. Through testing, the times of the acceleration effect of working on the GPU are 72.472, 149.536, 125.39. The used framework is Tesla k20c. Bayesian, KNN and SNN algorithms are based on supervised classification. The experiment shows the image process results and the times, which meet the requirements.

Keywords Supervised learning algorithm, KNN, SNN, Simple Bayesian algorithm

1 引言

本文介绍了简单贝叶斯算法、KNN、SNN 算法在 GPU 上的实现。第 2 节介绍了 3 种经典分类算法的基本思想;第 3 节介绍了贝叶斯分类算法在 GPU 上的实现方法;第 4 节展示了 3 个图像分类算法的图像处理结果;最后总结全文。

2 3 种经典的图像分类算法

2.1 最近邻算法——KNN

KNN 算法,即最近邻算法,是由 Cover 和 Hartzai 于 1968 年提出的一个较好的分类算法。该方法的思路非常简单直观:在一个大的样本库中,如果某一个样本的某些特征符合并属于大部分样本的特征,那么该样本即属于这个样本库,换句话说就是物以类聚。该算法在分类的过程中只依靠它最近的一个或者多个样本。

KNN 最近邻算法属于监督学习算法,主要应用于未知事物的分类。其判别思想为:样本属于距离最近的一类,同时具有该类所具有的特征。

K-近邻(K-Nearest Neighbors, KNN)算法是一种基于实例的分类方法。即输入一个未知的样本,然后找出距离该样本最近的 K 个样本,判断这 K 个样本所属类别,该未知样本

即属于该类。K 近邻分类其实是一种较为懒惰的算法^[21],其先存储这些样本,当需要时再进行分类。该算法较为明显的一个缺点就是如果样本较为复杂,则会导致计算过程复杂,从而消耗大量的时间,因此在需要满足实时性的情况下,该算法不适用。

KNN 算法的核心思想是:如果一个样本中的相邻 K 个样本的 N 个特征大部分属于某一个类别,那么该样本则属于该类别,并具有该类别所具有的特征。该算法所做的决定完全取决于它相邻的 K 个样本,因此被称作最近邻,故在做最后的决策时,只有最近邻的 K 个样本对该样本起决定性的作用。K 近邻算法只与最近的样本有关系,因此对于较为复杂且交叉的样本,具有较好的分类效果。

2.2 SNN 图像处理算法

BP 神经网络的代表者是 Rumelhart 和 McClland,“反向传播(backpropagation)”一词的使用出现在 1985 年以后,它的广泛使用是在 1986 年 Rumelhart 和 McClland 所著的 *Parallel Distributed Processing* 出版以后。BP 神经网络是一种根据误差逆传播的前馈网络,是目前应用研究领域中使用得比较广泛的网络模型之一。BP 神经网络具有很强的存储能力,能够存储大量的输入和输出数据,而这种存储不需要知道其映射关系模型。BP 神经网络使用了梯度下降法,并通

本文受国家自然科学基金:CPU/GPU 异构系统下高光谱遥感影像降维多级协同并行计算方法及优化策略(61272146)资助。

李思瑶 硕士,主要研究方向为高性能图像处理;周海芳 教授,主要研究方向为高性能图像处理、并行算法;方民权 博士,主要研究方向为高性能图像处理。

过反向传递的方式传递网络的权值和参数,从而使得整个网络的误差达到最小。BP神经网络包含输入层、隐藏层及输出层。

BP神经网络具有许多优点,但也有很多缺点,为了改进算法并弥补神经网络的缺点,本文提出了SNN算法。

SNN结构型神经网络包含两个阶段:1)训练阶段,根据已知的输入样本进行反复训练,从而得到一个较好的权重值;2)识别阶段,根据确定的权重值确定待测样本的类别属性。

训练样本时,分别对每一个子网络进行训练,此时如果输入的样本 x 属于第 i 类,则 SNN_i 最优的输出值为 1,其他的网络输出值为 0。在识别过程中,对每一个子网络 SNN_i 的输出结果 $p(w_i|x)$ 进行判断,根据判断结果划分不同的类别。在此过程中,DA 是一个决策算法,它根据子网络的输出结果决定 X 所属的类别,从而确定对子网络的需求情况。

2.3 贝叶斯分类算法

贝叶斯决策算法是贝叶斯派系中的一个重要组成部分。所谓贝叶斯算法,即是通过某些样本的先验概念,利用贝叶斯公式计算出样本的后验概率,对于某一个样本的概率,通常选择具有最大后验概率的类别作为该样本的类别。

所谓贝叶斯决策,就是在样本不是完全已知的情况下,对未知的状态进行估计,用贝叶斯公式进行计算,最终得出最后的决策。

假设 X 和 Y 分别为一对随机变量,它们的联合概率 $P(X=x, Y=y)$ 是指当变量 X 取值 x , 变量 Y 取值 y 时的概率,条件概率指的是其中一个变量是在已知的情况下的概率。因此, X 和 Y 的联合概率可以表示为:

$$P(X, Y) = P(Y|X)P(X) = P(X|Y)P(Y)$$

简单的调换一下顺序,就可以得到 $P(Y|X)$,称之为贝叶斯定理:

$$P(Y|X) = \frac{P(Y|X)P(X)}{P(X)}$$

$P(Y)$ 是先验概率 (Prior probability), 或称 Y 的先验概率。 $P(X|Y)$ 代表假设 Y 成立的情况下,观察到 X 的概率。 $P(Y|X)$ 是后验概率 (posterior probability), 或称条件 X 下 Y 的后验概率。

贝叶斯决策算法的思路如下:1)已知条件概率参数以及先验概率;2)通过贝叶斯公式求后验概率;3)根据后验概率大小进行决策分类。

3 简单贝叶斯图像分类算法的 GPU 设计与实现

对于大规模矩阵图像处理,本文将图像计算函数放到 GPU 上运行以获得 GPU 加速,同时 GPU 和 CPU 协同工作、合理分工。

GPU 和 CPU 协同工作的流程如图 1 所示。

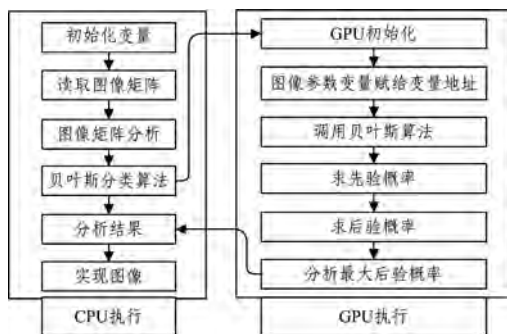


图 1 GPU 和 CPU 协同工作

该算法的流程如下:

- 1)对图像地址变量,初始化参数值,以便进行后验概率计算;
- 2)在 GPU 上进行先验概率计算,根据先验概率求得后验概率;
- 3)计算出最大后验概率值;
- 4)进行分析,将最大后验概率图像作为结果返回。

本文仅介绍 GPU 上运行的 3 个函数,核为 1。GPU 上的函数主要有 3 个: bayes_single, bayes_gpu_kernel 和 bayes_gpu。

以下是贝叶斯算法的并行化步骤。

分析串行程序的数据结构,如图 2 所示,在图 2 中与贝叶斯算法并行化相关的结构是 BAYES,其定义如下:

```
/* structure for bayes method with normal distribution */
typedef struct bayes
{
    int classnum;
    int attrnum;
    double * * mean; //mean[classnum][attrnum]
    double * * variance; //varinace[classnum][attrnum]
}BAYES;
```

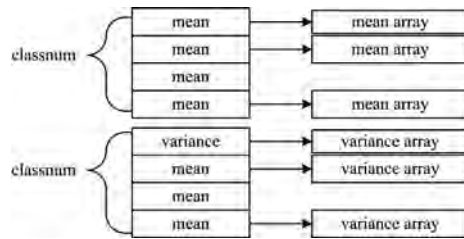


图 2 串行版本贝叶斯数据结构的内存排列

在串行版本中,读入的 BMP 分类图像的原始数据内存排列顺序如图 3 所示,图像之间的内存块是不连续的。



图 3 串行图像数据结构内存排列

(1)对以上数据结构和图像进行内存排列顺序的修改,对于 bayes 来说,每个 mean 和 variance 的访问是通过 bayes->mean[i] 和 bayes->variance[i] 获取要操作数据的首地址,然后通过首地址访问具体的数据。相对这种频繁寻址操作, CUDA 对连续内存的访问效率比较高,因此将原数据结构修改为利于 CUDA 加速的连续的内存结构,即将 bayes 中的 mean 和 variance 变成连续内存的一维数据,如图 4 所示。将 bayes 中的 mean 和 variance 数据脱离出来,形成一维连续的内存结构,相当于将原来的 classnum 个 mean array 或 variance array 中的数据连接起来,形成连续的内存布局,这样利于核函数的参数传递和运行效率。对于串行版本,将图像之间的内存块修改为连续的内存排列,如图 5 所示。

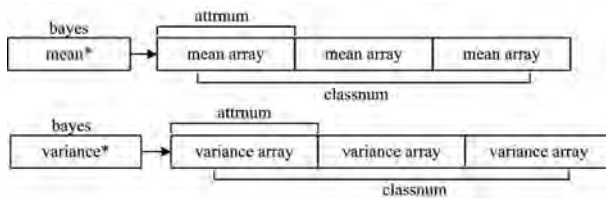


图 4 并行数据结构内存排列

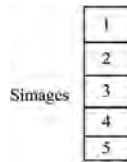


图 5 串行数据结构内存排列

(2)确保改造后串行程序和改造前程序的运行结果一样。基于改造后的串行程序,并行算法的思路是循环计算目标图像的一个点,且循环间没有依赖关系,因此启动 $imagewidth * imageheight$ 个 CUDA 线程,即每个线程完成对应像素点的计算,当 kernel 执行完毕后即生成了最后的目标区域,如图 6 所示。图 6 中的 CUDA thread 代表当前线程处理的像素点,目标图像的宽度和高度分别是 $imagewidth$ 和 $imageheight$ 。

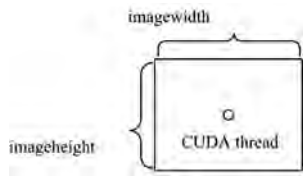


图 6 并行计算模式

(3)基于并行思想,编写 GPU 的主函数 `bayes_gpu`,在该函数中完成以下操作:

1)申请 `bayes_gpu_kernel` CUDA 核函数所需要的 GPU 显存,包括输入数据和输出图像。

2)将 kernel 运行所需要的数据(包括 BAYES, `simages` 和其他相关参数)从 CPU 端内存移动到 GPU 端的显存,供 kernel 使用。

3)设置 kernel 启动的线程配置,即每个 block 包括的线程数,该数值可以由用户输入,而每个 grid 包括的 block 数量等于 $imagewidth * imageheight /$ 每个 block 包括的线程数。

4)启动 `bayes_gpu_kernel` 进行目标图像像素值的计算,在 `bayes_gpu_kernel` 的开始先判断当前线程要处理的像素值是否在目标图像范围内,如果不在,则直接返回,即不计算。否则利用 bayes 算法计算对应的目标值。

5)计算完成后,将目标图像从 GPU 端移动到 CPU 端的内存上。

6)释放申请的 GPU 显存和临时内存。

(4)将 CPU 端内存上的图像保存。此次使用一个 GPU 核进行计算,在计算判断图像属于哪个类别时,我们进行了图像先验概率的计算,并且利用先验概率求得后验概率,其中后验概率是根据贝叶斯公式求得的。在程序中我们定义了像素和图像类别的变量,并且根据贝叶斯学习算法进行运算得到后验概率。在求取后验概率的过程中,使用了正态变量求值,对所有类别的后验概率的求取进行了遍历。因为正态变量都是正态分布的,所以使用正态分布的概率函数进行 n 个变量概率的相乘。

本文使用贝叶斯分类中所用到的特征变量,其符合正态变化规律并且根据其方差、标准差和均值进行预测。本算法在遍历计算所有的图像类别后得到每个像素的类别,然后将其重新组成一幅更为真实、所取得的各个像素值点的概率最大的类别图像,即为所求的图像。

$Argv[i]$ 是第 i 个输入的数据,在本算法中共有 4 个输入

数据: $Argv[0], Argv[1], Argv[2], Argv[3]$,分别代表函数名称、remote、输入文件名字、输出名字。

贝叶斯图像分类的思想是后验概率 = 先验概率 * 标准似然度。标准似然度和先验概率都可以通过统计方式和数学方法计算得出,因此用后验概率求取贝叶斯分类算法得出图像是可行的。

程序把贝叶斯学习和贝叶斯分类放到 GPU 设备端上运行,使得计算所花费的时间越来越短。例如,计算先验概率和数值统计都是大规模计算,将计算量大的程序放到设备端运行能节省时间。

在程序中, GPU 上运行的函数有 `bayes_single`, `bayes_gpu_kernel`, `bayes_gpu`, 为方便识别计算,将函数互相调用。其中 `bayes_gpu` 是在设备端进行程序的初始化并且协调各个函数进行综合计算的函数。`bayes_gpu_kernel` 是在最后生成一幅图片时给图片的所有像素点赋值的函数。因此,图像分类方法在 GPU 上运行的方式是按照像素值进行操作的。

简单贝叶斯统计分类算法的关键在于利用先验概率求得后验概率,这样就能判断哪个概率值较大,从而判断这个像素值点所属类别。本函数的使用方法是 `Bayes remote TMBand out. bmp threadnum`。基于 GPU 的 KNN 和 SNN 算法的计算原理大致与基于贝叶斯的 GPU 的实现相同,在此不再赘述。

4 图像处理结果

首先确定电脑配置,机器 GTX480,操作系统 win7,CPU Intel Xeon W5305,内存 60GB,显卡 Tesla k20c。

我们在 GTX480 机器上运行 Bayes, KNN, SNN 程序,得到运行时间如表 1—表 3 所列。

表 1 简单贝叶斯分类算法在 GPU 上的时间与加速比

图像尺寸	CPU 串行时间/s	GPU 并行时间/s	加速比
50×50	0.006487	0.001187	5.467
512×512	0.635857	0.005071	125.39

表 2 KNN 算法在 GPU 上的时间与加速比

图像尺寸	CPU 串行时间/s	GPU 并行时间/s	加速比
50×50	1.381	0.788	1.7525
512×512	136.524	1.8838	72.472

表 3 SNN 图像分类方式在 GPU 上的时间与加速比

图像尺寸	CPU 串行时间/s	GPU 并行时间/s	加速比
50×50	0.009534	0.002172	4.389
512×512	0.822	0.005497	149.536

根据 B/S 结构遥感影像并行计算系统,得到的图像处理结果如图 7—图 9 所示。

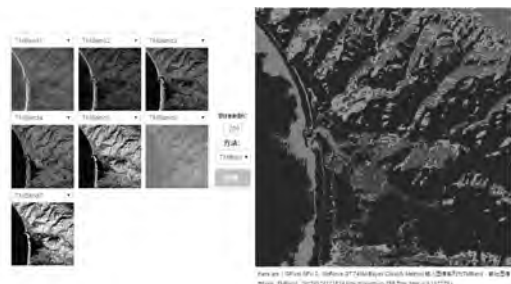


图 7 简单贝叶斯分类的图像处理结果

86%,2015年的观测效率为89%,增加了3个百分点。用云量仪引导望远镜观测后,观测设备得到有效利用,观测到的目标更多,观测效率显著提高。因此,用云量仪引导望远镜观测将更具实用性。

结束语 光学望远镜的弱点之一就是光路不能透过云层观测空间运动目标,导致在局部天气不佳的状况下光学望远镜的使用效率大大降低。针对这一问题,本文将云量图分为24个区块,再对每一个区块进行云量计算,根据云量计算的结果,把望远镜优先引导到无云区域或云量相对较少的区域,使望远镜在布局天气不佳的情况下进行有效观测。

通过云量仪引导望远镜定向,实现简单、方法可行、资金投入低,具有较大的研究空间和较高的使用价值。针对未来规模化的空间运动目标观测网具有实际的应用价值,特别是对自动跟踪和自动搜索的光电筒望远镜和光电望远镜阵能够发挥很好的效应,下一步的研究计划就是利用云量仪上的云层分布特性,进行周期性的自动预报更新,以保证预报能够引导望远镜在无云和避开月亮的空间区域进行观测,当云量分布图发生变化时再次进行预报更新和引导望远镜定向,以保证在局部多云天气不佳的情况下有效提高望远镜的观测效率。

参考文献

[1] 孙航,张海波,曹立华,等.大口径光电探测设备主镜晃动的误差补偿[J].光学精密工程,2014,22(1):85-91.

[2] YANG J,ZUO Y X,LOU Z, et al. Conceptual design studies of the 5 m terahertz antenna for Dome A, Antarctica[J]. Research in Astronomy and Astrophysics, 2013, 13(12): 1493-1508.

[3] 王玥,周城,熊承义,等.基于纹理自适应全变分滤波的图像分块压缩感知优化算法[J].计算机科学,2016,43(2):307-310,315.

[4] 王荣芳,焦李成,刘芳,等.利用纹理信息的图像分块自适应压缩感知[J].电子学报,2013,41(38):1506-1514.

[5] 邓莉,鲁瑞华.一种基于三角形网格的图像分块盲复原算法[J].计算机科学,2008,35(9):237-238.

[6] 王恩旺,王恩达.改进的帧差法在空间运动目标检测中的应用[J].天文研究与技术,2016(9):29.

[7] 杨芳,滕桂法,田学东.视觉乐谱图像动态多阈值二值化方法[J].计算机科学,2016,43(1):310-314.

[8] YIN H J. Blind Source Separation and Genetic Algorithm for Image Restoration[C]//Proceedings of IEEE Conference on Advances in Space Technologies, 2006:167-172.

[9] WONG J, SHENG Y, LING Z. Study on the application of ROI, [J]. Computer Block Matching Panoramic Image Mosaic Method Based on Robust, 2007, 24(1): 229-231.

[10] OHM J R, GRUNEBERG K, HENDRIKS E, et al. A Realtime Hardware System for Stereoscopic Video Conferencing with Viewpoint Adaptation[J]. Signal Processing Image Communication, 1998, 14: 147-171.

[11] 王恩旺,王恩达.运动目标检测中光流扰动效应应用研究[J].计算机系统应用,2017,26(3):1-7.

(上接第145页)

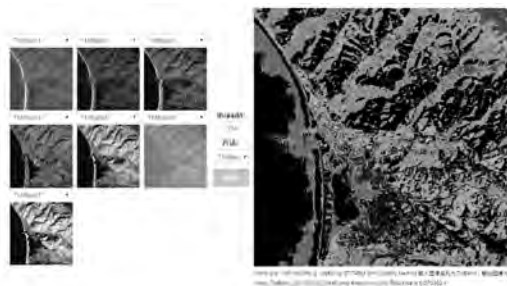


图8 SNN分类的图像处理结果

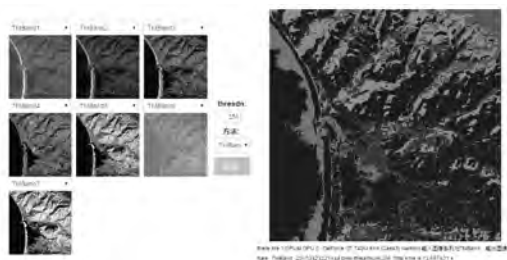


图9 KNN图像的分类处理结果

结束语 通过基于GPU的遥感图像简单贝叶斯分类、KNN和SNN分类并行处理,得到图像大小为 50×50 的3个分类算法的并行加速比,分别为5.467,1.7525和4.389,图像大小为 512×512 的3个分类算法的并行加速比分别为125.39,72.472和149.536。最大加速比149.536为图像大小 512×512 的SNN算法结果,最小加速比1.7525是图像大小 50×50 的KNN算法结果,实验结果证明了3个并行算法的合理性,对遥感图像分类处理的加速起到了一定的作用。

最后利用自己编写的B/S结构系统进行测试,并得到了所需要的图像分类结果。同时,本文利用并行得到的图像效果和串行计算得到的图像效果大致相同。由此可见,提出的3个基于GPU的遥感图像分类算法符合精度要求,本文实现的并行算法是正确的,计算得出的加速比也较为理想。在这3个算法中,KNN算法最合适进行基于GPU的图像处理并行化,这是因为KNN算法有大量的矩阵计算,适合密集计算的GPU。

参考文献

[1] 沈欣.光学遥感卫星轨道设计若干关键技术研究[D].北京:清华大学,2012.

[2] 林前.图像分割质量的协同评价框架设计与实现[D].成都:西南交通大学,2011.

[3] 严蔚敏,吴伟民.数据结构(C语言版)[M].北京:清华大学出版社,2011.

[4] 史春.基于遥感处理的交通信息提取研究和实现[D].北京:清华大学,2008.

[5] 陈锦伟.遥感图像配准和拼接技术研究[D].杭州:浙江大学,2004.

[6] 陈忠.高分辨率遥感图像分类技术研究[D].北京:中国科学院遥感应用研究所,2006.

[7] 赵春晖,刘春红.超谱遥感图像降维方法研究现状与分析[J].中国空间科学技术,2004(5):28-36.

[8] 周海芳,赵进.基于GPU的遥感图像配准并行程序设计及存储优化[J].计算机研究和发展,2012,49(2):281-286.

[9] 陈志远,黄少滨,韩丽丽.现代模式逻辑在计算机科学中的应用研究[J].计算机科学,2013,40(S1):70-76.