

GGCN: 基于 GPU 的高光谱图像分类算法

张明华¹, 邹亚晴¹, 宋巍¹, 黄冬梅^{1,2*}, 刘智翔¹

¹上海海洋大学信息学院, 201306;

²上海电力大学电子与信息工程学院, 上海 200090

摘要 高光谱图像分类是遥感领域的研究热点之一, 是对地观测的重要手段, 在地物的精细识别等领域具有重要的应用。使用卷积神经网络(CNN)可以有效地从原始图像中提取高级特征, 具有较高的分类精度。但 CNN 计算量巨大, 对硬件要求较高。为了提高模型计算效率, 可以在图形处理器(GPU)上进行 CNN 模型的训练。现有的并行算法, 比如 GCN(GPU based Cube-CNN), 无法充分利用 GPU 的并行能力, 算法加速效果并不理想。为了进一步提升算法效率, 提出基于通用矩阵乘法(GEMM)算法的 GGCN(GPU based Cube-CNN improved by GEMM)并行加速算法, 通过 G-PNPE(GEMM based Parallel Neighbor Pixels Extraction)对输入数据和卷积核进行重新组织排列, 实现卷积的并行计算, 有效地提高了 GPU 的利用率并进一步提升了算法的训练效率。通过分析在三个数据集上的实验结果发现, 改进算法的分类精度与原算法保持一致, 而且模型的训练时间缩短了 30% 左右, 表明算法的有效性和优越性。

关键词 成像系统; 高光谱图像; 图形处理器; 通用矩阵乘法; 并行计算

中图分类号 TP751

文献标志码 A

doi: 10.3788/LOP57.201101

GGCN: GPU-Based Hyperspectral Image Classification Algorithm

Zhang Minghua¹, Zou Yaqing¹, Song Wei¹, Huang Dongmei^{1,2*}, Liu Zhixiang¹

¹College of Information Science, Shanghai Ocean University, Shanghai 201306, China;

²College of Electronics and Information Engineering, Shanghai University of Electric Power, Shanghai 200090, China

Abstract Hyperspectral image classification is one of the research hotspots in the field of remote sensing. It is an important means of earth observation and has important applications in areas such as fine identification of ground objects. The use of convolutional neural networks (CNN) can effectively extract advanced features from the original image with high classification accuracy. However, CNN has a huge amount of calculations and requires high-performance hardware. In order to improve the computational efficiency of the model, the CNN model can be trained on the GPU. Existing parallel algorithms such as GCN (GPU based Cube-CNN) cannot make full use of the parallel capabilities of the GPU, and the algorithm acceleration effect is not ideal. In order to further improve the efficiency of the algorithm, the GGCN (GPU based Cube-CNN improved by GEMM) parallel acceleration algorithm based on the general matrix multiply (GEMM) algorithm is proposed. G-PNPE (GEMM based Parallel Neighbor Pixels Extraction) reorganizes and arranges the input data and convolution kernel to achieve parallel calculation of convolution, which effectively improves the utilization of GPU and increases the training efficiency of the algorithm. By analyzing the experimental results on the three datasets, the classification accuracy of the improved algorithm is consistent with the original algorithm, and the training time of the CNN network is shortened by about 30%, which proves the effectiveness and superiority of the algorithm.

Key words imaging systems; hyperspectral image; graphics processing unit; general matrix multiply; parallel computing

OCIS codes 110.4234; 200.4260; 200.4960

收稿日期: 2019-12-16; 修回日期: 2020-01-16; 录用日期: 2020-02-25

基金项目: 国家自然科学基金(41671431)、上海市科委部分地方院校能力建设项目(17050501900)、上海市青年科技英才扬帆计划(18YF1410100)

* E-mail: dmhuang@shou.edu.cn

1 引言

高光谱遥感是对地观测的重要手段,在获取地表图像信息的同时得到地物的光谱信息^[1]。通过对图像进行精细化分类,可以在农业、环境管理和城市规划等众多关键领域提供重要的决策参考^[2]。

传统的机器学习方法如基于支持向量机(SVM)的分类方法^[3]、随机森林^[4]等,缺乏遥感专业背景知识,对这类算法分类精度的提升已经遇到瓶颈。与传统机器学习方法相比,深度学习方法能自动从图像中提取特征,并且在处理高维度的数据时能避免过拟合现象^[5],因此成为研究热点^[6-7]。Leng等^[8]提出了Cube-卷积神经网络(CNN)-SVM模型用于高光谱图像分类,具有很好的分类效果。由于CNN的特性,该模型在传统处理器CPU上进行训练时耗时严重,无法满足实时性的需求。

近些年,随着GPU通用计算技术的迅速发展,人们开始在GPU上进行CNN网络的训练以寻求更快的训练速度^[9-10]。沈恬等^[11]在使用GPU实现CNN的加速优化时,使用更多的寄存器取代共享内存实现数据共享。由于寄存器的读写响应时间远小于共享内存,因此相较于CPU的实现,取得了不错的加速效果。但是,这种方式对输入数据的形状极为敏感,灵活性较低。Dong等^[12]提出基于GPU的GCN(GPU based Cube-CNN)算法实现Cube-CNN-SVM模型的并行。使用PNPE(Parallel Neighbor Pixels Extraction)并行近邻像素提取算法进行数据的预处理,以并行的方式从原始图像中提取Cube数据,然后载入到GPU中进行CNN的训练,有效地提高了模型训练效率。

卷积层是CNN模型中比较重要的一部分。通过对GCN模型的进一步分析发现,卷积过程在整个模型耗时中占比接近40%。在GCN模型中使用滑窗的方式实现卷积计算,这种实现方式最简单直接,但是对内存的访问比较混乱,难以实现并行,而且对GPU的资源利用率较低。因此,如果能够减少卷积的执行时间就能进一步提升模型效率。Nakasato^[13]提出CNN卷积实现的另一个方式:使用通用矩阵乘法(GEMM)算法,将卷积计算过程转化为矩阵相乘的方式。矩阵的乘法运算不仅方便数据在内存中的读写,同时还适合利用GPU实现并行计算。但是在卷积运算中使用GEMM算法需要对高光谱数据进行重新组织。由于高光谱图像具有维度高、波段多、数据量大等特点,在对高光谱数据

进行低维重新排列的时候会导致两个问题:1)增加额外操作;2)算法效率较低。因此,本文提出基于GEMM的通用并行近邻像素提取算法(G-PNPE),以并行的方式将Cube数据的提取和数据的重新组织结合在一起,处理后的数据输入到GGCN(GPU based Cube-CNN improved by GEMM)模型进行训练。最终整个模型能在GPU中实现Cube数据的并行卷积,模型训练过程中卷积消耗的时间大大缩短,进而整个算法的效率得到明显地提升。

2 Cube-CNN-SVM 分类模型

CNN是人工神经网络中的一种,其特点是通过图像卷积进行特征提取,可以很好地应用于图像分类,在图形处理器芯片上拥有很大的应用场景和加速空间。

在Cube-CNN-SVM分类模型中,为了提高分类精度,将高光谱图像的光谱信息和空间信息进行融合,输入到分类器进行训练。为了达到空-谱融合的目的,采用PNPE算法进行图像预处理。该预处理算法在提取带标签的像素的同时会提取该像素相邻位置的像素。将标签像素与其相邻像素一起作为训练数据输入到模型进行训练。这种提取像素的近邻像素提取策略主要有三种方式:1)单像素策略(1P),只提取带标签的单一像素作为样本;2)4近邻策略(4N),提取带标签的像素及其上下左右四个相邻位置的像素作为样本;3)8近邻像素策略(8N),提取以带标签像素为中心的 3×3 个像素作为样本。通过这些方式提取到的数据称为Cube样本。

首先对高光谱图像进行预处理,产生Cube样本,然后将样本输入到模型进行训练。训练后的数据作为高级特征使用SVM作为分类器进行分类。图1所示为Cube-CNN-SVM整体架构图。

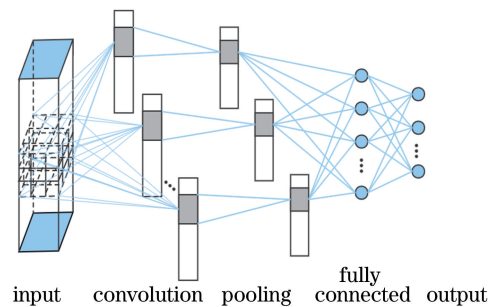


图1 Cube-CNN-SVM模型框架

Fig. 1 Cube-CNN-SVM model framework

Cube-CNN-SVM模型使用Cube立方体作为输入进行训练,每个立方体在训练过程中相互独立,因此很适合并行加速。

3 GGCN 并行计算模型

3.1 GEMM 算法

按照定义实现多个嵌套的卷积运算通常包括多层循环,其中,外部循环遍历输入的图像,内部循环遍历卷积核。图 2 所示为卷积层中的传统卷积操作。为了简单起见,已经省略卷积涉及的其他操作。输入数据采用三个 3×3 输入要素,并输出两个 2×2 输出要素。每个输入与它们相应的卷积核进行卷积,然后加在一起获得输出。

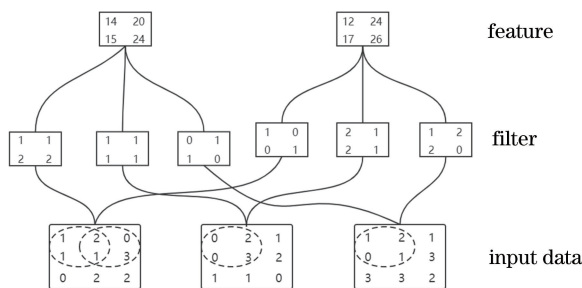


图 2 卷积运算的定义形式

Fig. 2 Definition form of convolution operation

按照卷积的定义进行卷积的计算存在执行效率较低、不适合使用并行计算等问题。可以通过对输入数据以及卷积核的重组和复制,将卷积的运算转换为矩阵相乘的形式来进行优化。

GEMM 的核心思想就是对输入数据进行展开、复制,对卷积核进行重新排列。使用这种方法在正向传播过程中,每个卷积运算都将转换为矩阵乘积的形式,相应地,可以将反向传播简单地看作另一个矩阵乘积的运算。这种方式不仅使得算法在训练过程中对内存的访问更加规则,而且方便使用并行的方式进行加速。

图 3 所示为卷积层的矩阵乘积形式。重新组织每个卷积的输入,使每一行包含计算输出特征的一个元素所需的所有输入值。卷积核同样被展开然后连接形成卷积核矩阵。当新的输入矩阵乘以卷积核矩阵时,将自动计算输出特征。

卷积层是 CNN 模型中最重要的一部分,同时也是整个模型耗时最严重的部分。因此,如果能有效提高卷积层的计算效率就能提高算法整体的运行效率。使用 GEMM 算法的思想将卷积运算变为矩阵乘积的形式,为并行化做好准备。

3.2 G-PNPE 预处理算法

引言中提到,要将 GEMM 应用在模型中需要重新组织高光谱数据,而这会增加额外的工作导致

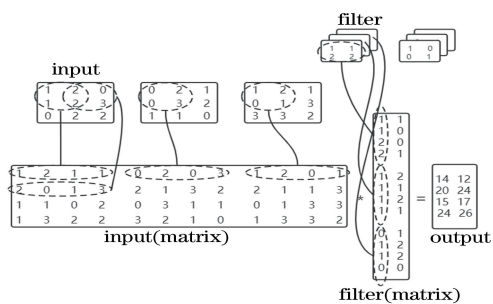


图 3 卷积运算的矩阵乘积形式

Fig. 3 Matrix multiplication form of convolution operation
算法复杂度增加。因此,结合 GCN 模型的特点,本文提出 G-PNPE 并行数据预处理算法。该预处理算法有两个特点:1)结合 Cube 数据提取的过程将数据转换为 GEMM 算法适用的结构,避免了额外的操作;2)该算法在 GPU 中运行,使用并行的方式将高光谱图像提取为 Cube 形式并存储在全局内存中,用于后续模型的训练,对算法效率的影响较小。

Cube 数据提取的过程如下。

1)将高光谱图像看作一个维度为 $H \times W \times C$ 的三维矩阵,在 CPU 中扫描每个有效元素(像素值非空的像素)的索引位置为 (x, y, c) ,通过该索引可以得知索引近邻像素的位置信息。对每个 Cube 中 $3 \times 3 \times C$ 个元素进行索引可以得到第 i 个元素在数据矩阵中的位置。

2)按照上述方式对卷积核矩阵进行扫描,计算得到卷积核中第 j 个元素在转变之后的矩阵中的位置

$$x \times H + y \times W + c \times 9 + i \times C \times H \times W, \quad (1)$$

$$x \times H + y \times W + c \times 9 + j \times H \times W, \quad (2)$$

式中: H 为高光谱图像的高; W 为高光谱图像的宽; C 为高光谱图像的波段数。

综上所述,首先使用 G-PNPE 算法从高光谱数据中提取 Cube 数据,提取到的 Cube 数据被组织成数据矩阵,然后输入模型中进行卷积运算。图 4 给出了高光谱图像数据预处理以及卷积运算的过程。

3.3 模型训练

在 GGCCN 框架中,训练样本在经过预处理之后被存放在全局内存,在模型训练中可以被 GPU 直接访问。在使用 CNN 训练模型时,可以选择不同的激活函数和损失函数。

以本文采用的输出层 softmax 激活函数与交叉熵损失函数为例,损失值的计算式表示为

$$J = -\frac{1}{n} \sum_{i=1}^n [y_i \ln a_i + (1 - i) \ln(1 - a_i)], \quad (3)$$

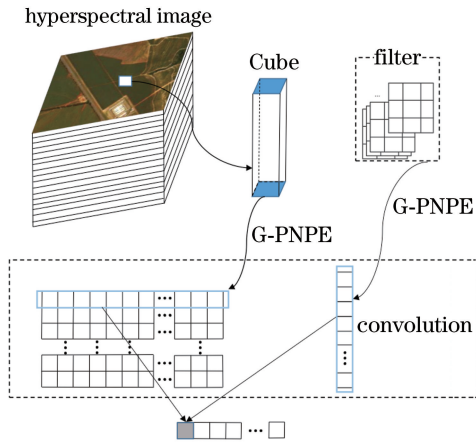


图4 图像预处理以及卷积运算

Fig. 4 Image preprocessing and convolution operation

式中: n 为样本类别; y_i 为模型期待的输出值; a_i 为模型实际输出。

为了提高模型精准度,使用 MBGD(Mini-batch Gradient Descent)算法调整算法的参数。以第 i 个

神经元为例,通过(3)式定义的损失函数可以计算出对输出值的偏导为

$$\frac{\partial J}{\partial a_i} = \frac{a_i - y_i}{a_i(1 - a_i)} \quad (4)$$

输出层的函数被定义为 $a_i = f(z_i)$, 其中 z_i 是权重与输入数据之间的点积。根据链式法则,可以得到:

$$\frac{\partial J}{\partial z_i} = \frac{\partial J}{\partial a_i} \times \frac{\partial z_i}{\partial w_{ij}} = x_{ij}(a_i - y_i) \quad (5)$$

通过计算损失函数 a_j^{L-1} , 对通过反向传播到前一层的参数进行更新。

$$a_j^{L-1} = \sum_i w_{i,j} \times \frac{\partial J}{\partial z_i} \quad (6)$$

模型的训练阶段被分成两个部分:前向传播和反向传播。在前向传播阶段计算每个 Cube 的分类结果,后向传播阶段更新网络的权重以便于获得更好的分类结果。具体实现的伪代码如表 1 所示。

表1 算法伪代码

Table 1 Algorithm pseudocode

Algorithm: GGCN

Input: Hyperspectral image

1, Data preprocessing: processing <<< gridsize, blocksize >>>

 i -th iteration;

Forward propagation

2, Convolutional: convol <<< gridsize, blocksize, 0, stream >>>

3, Pooling: maxpooling <<< gridsize, blocksize, 0, stream >>>

4, Fully connected: fullyconnected <<< gridsize, blocksize, 0, stream >>>

5, Output: output <<< gridsize, blocksize, 0, stream >>>

6, Copy classification results to CPU to calculate the loss:

7, Copy data: cudaMemcpy()

8, Calculate the loss: lossfunction()

Backward propagation

9, Output: bp_output <<< gridsize, blocksize, 0, stream >>>

10, Fully Connected: bp_fullyconnected <<< gridsize, blocksize, 0, stream >>>

11, Pooling: bp_maxpooling <<< gridsize, blocksize, 0, stream >>>

12, Convolutional: bp_update_kernel <<< gridsize, blocksize, 0, stream >>>

Output

End

4 实验与结果分析

为了验证改进模型的精度和效率,本文在非并行、原模型、改进模型上进行了对比实验。高光光谱图像在获取的过程中由于拍摄区域、分辨率不同而导致的两个主要差异:数据量大小差异和地

物信息差异。本文使用 KSC (Kennedy Space Center)、PU (Pavia University Scene) 和 Indian Pines 三个高光光谱图像作为实验数据,表 2 展示了数据集的具体信息。实验平台使用的 CPU 是 Intel Core i7-6700, 主频 3.4 GHz, 内存 16 GB, 搭载的显卡型号是 NVIDIA GeForce GTX 1050。实验数

据采用双精度浮点类型, CUDA 版本为 8.0。

将数据集按照 4:1 的比例被分为训练集和测试集, 表 3 展示了使用 CPU(非并行近邻像素提取方法)、PNPE、G-PNPE 三个不同实现方式, 对三个数据集分别使用 1P、4N、8N 三种近邻像素提取策略

进行数据预处理的实验结果。

从表 3 可以看出, 使用 G-PNPE 算法与 PNPE 算法进行预处理的时间消耗相差无几, 说明本文算法与原模型的并行预处理算法相比没有增加时间消耗, 比非并行的预处理算法效率提高 5~8 倍。

表 2 遥感数据集信息

Table 2 Information of the remote sensing datasets

Dataset	Sensor	Class number	Dimension	Top 5 classes	Size /MB
KSC	AVIRIS	13	512 × 614 × 176	Water, scrub, spartna-marsh, mud-flats, salt-marsh	56.8
PU	POSIS	9	610 × 340 × 103	Meadows, asphalt, bare-soil, self-blocking bricks, trees	33.2
Indian Pines	AVIRIS	16	145 × 145 × 224	Soybean-mintill, corn-notill, woods, soybean-notill, corn-mintill	5.7

表 3 不同数据预处理方式时间消耗对比

Table 3 Comparison of time consumption of different data preprocessing methods

Dataset	Neighbor pixel extract strategy	Time /s		
		CPU	PNPE	G-PNPE
KSC	1P	2.65	0.45	0.51
	4N	4.89	0.88	0.92
	8N	6.21	1.12	1.22
PU	1P	2.21	0.31	0.33
	4N	3.30	0.52	0.49
	8N	3.87	0.66	0.71
Indian Pines	1P	1.05	0.17	0.18
	4N	1.65	0.21	0.20
	8N	2.17	0.28	0.26

然后在 Cube-CNN-SVM(非并行)、GCN、GGCN 三种模型下, 使用 MBGD 算法(Batchsize 为 10 和 Batchsize 为 100 两种情况)分别对 3 组数据进行了 5 次训练, 统计模型的平均训练时间以及训练结果的平均精度。每次实验中均使用 20 个 3 × 3 × 19 的卷积核, 学习率设置为 0.01, 迭代次数为 100, 像素提取的策略均为 8N。

表 4 所示为不同模型的平均训练时间和加速比的结果。从表 4 可以看出, 两种并行模型在三个数据集的表现相较于非并行的模型, 平均训练时间都有大幅减少。改进的模型 GGCN 相较于 GCN 模型在时间效率上提升了 20%~30%左右, 速度提升明显。前面提到三个数据集大小差异明显, 从表中也可以看出, 改进模型在数据量

表 4 不同分类模型平均训练时间和加速比

Table 4 Comparison of running time and speedup of different classification models

Dataset	Method	Time /s		Speedup ratio
		MBGD(batchsize is 10)	MBGD(batchsize is 100)	
KSC	Cube-CNN-SVM	23123.62	23012.49	1.0
	GCN	3487.34	3322.34	6.6
	GGCN	2834.01	2598.78	8.2
PU	Cube-CNN-SVM	2231.23	2187.75	1.0
	GCN	351.46	338.11	6.3
	GGCN	286.22	230.06	7.8
Indian Pines	Cube-CNN-SVM	453.62	422.49	1.0
	GCN	107.34	102.34	4.2
	GGCN	84.01	80.78	5.1

较大时加速效果更好,加速比达到 6.6。在小型数据集上加速比仍达到 5.1,这说明改进模型对不同数据量的数据集有很好的适应能力。

表 5 所示为不同模型的分​​类精度。可以看出,本文模型与非并行模型相比,分类精度几乎一致,都维持在很高的水平,表明对算法的并行不会影

响算法的精度。从数据集角度分析,三个数据集中相同的地物较少,地物信息差异较大。但从结果来看,对不同数据集不同地物的分类模型都取得了很好的分类效果。这说明本文的并行计算模型不仅分类精度高,而且对不同地物的适应能力较强。

表 5 不同分类模型准确率

Table 5 Accuracy of different classification models

Dataset	Method	Accuracy / %	
		MBGD(batchsize is 10)	MBGD(batchsize is 100)
KSC	Cube-CNN-SVM	93.78	93.47
	GCN	93.33	93.12
	GGCN	93.67	93.92
PU	Cube-CNN-VM	96.67	95.21
	GCN	96.23	95.61
	GGCN	96.34	95.69
Indian Pines	Cube-CNN-SVM	94.78	94.67
	GCN	94.73	94.52
	GGCN	94.87	94.42

使用 nvprof 工具对 GGCN 模型训练过程进行监督和统计分析,可以得到模型的每一层的时间占比,表 6 所示为模型各部分在训练过程中消耗时间占比。可以看出,在改进后的模型中,卷积层在整个模型运行过程的时间占比降低了 10%,整个模型优化效果明显。

表 6 改进模型与原模型各层级时间占比
Table 6 Ratio of time between the improved model and the original model

Layer	Percentage / %	
	GCN	GGCN
Preprocessing	1.0	2.4
Convolution	38.2	28.0
Pooling	2.4	5.6
Fully connection	27.1	24.0
Output	19.0	22.0
Others	13.3	18.0

图 5 所示为训练过程中的分类精度和损失值的变化曲线,可以看出,模型收敛速度较快。

为了验证本文模型对卷积实现并行化的通用性、稳定性,本文逐层增加卷积层的次数,并计算在不同层数下的模型的加速比。

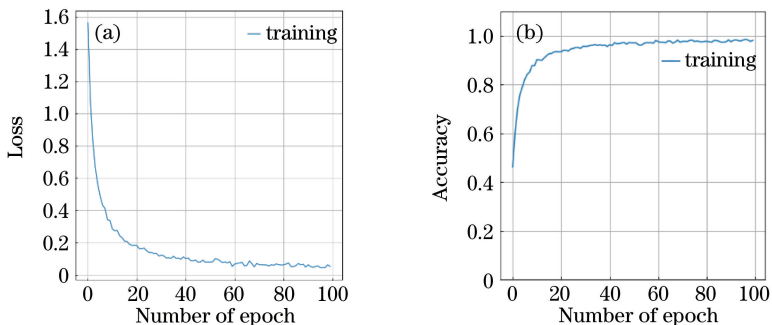


图 5 模型训练损失和精度变化曲线。(a)损失;(b)精度

Fig. 5 Model training loss and accuracy variation. (a) Loss; (b) accuracy

图 6 所示为 1~9 层的 GCN 模型以及本文改进的 GGCN 模型的加速比。可以看出,在卷积层数为 1 时本文改进的 GGCN 网络模型相比于 GCN 网络有更大的加速比,这说明了本文模型在效率方面的

优越性。还可以看出,随着卷积层数的增加,GCN 模型和 GGCN 模型的加速比呈现上升趋势并在卷积层为 4 时达到最大值,然后加速比开始下降。同时在下降过程中,本文提出的 GGCN 模型下降过程

更加缓慢,这说明本文模型相比于原模型具有更好的泛化能力,在应对多层网络时具有更好的表现。

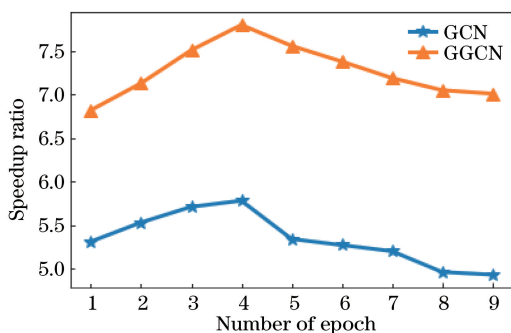


图6 不同数量卷积层加速比的变化

Fig. 6 Changes in the speedup ratio of different numbers of convolution layers

5 结 论

本文在高光谱图像并行分类模型 GCN 的基础上进行改进,利用 GEMM 的思想实现卷积运算并行化,针对 GEMM 在模型中应用的相关问题提出了 G-PNPE 数据预处理算法,通过该算法对数据进行处理并保存到全局内存,由 GPU 读取直接进行模型训练。实验结果表明,改进后的模型在保持精度不变的情况下比原模型速度提升 25%~30%,并且在多个数据集进行的实验表明,模型对高光谱数据集具有鲁棒性。同时验证了在多个卷积层的情况下,本文模型具有更好的加速表现和泛化能力。因此,所提 GGCCN 模型不仅在效率上有进一步的提升,而且具有较好的扩展能力。

参 考 文 献

[1] Zhang B. Advancement of hyperspectral image processing and information extraction[J]. Journal of Remote Sensing, 2016, 20(5):1062-1090.
张兵. 高光谱图像处理与信息提取前沿[J]. 遥感学报, 2016, 20(5):1062-1090.

[2] Bioucas-Dias J M, Plaza A, Camps-Valls G, et al. Hyperspectral remote sensing data analysis and future challenges[J]. IEEE Geoscience and Remote Sensing Magazine, 2013, 1(2):6-36.

[3] Camps-Valls G, Bruzzone L. Kernel-based methods for hyperspectral image classification [J]. IEEE Transactions on Geoscience and Remote Sensing, 2005, 43(6):1351-1362.

[4] Belgiu M, Drăguț L. Random forest in remote sensing: a review of applications and future directions [J]. ISPRS Journal of Photogrammetry and Remote

Sensing, 2016, 114:24-31.

- [5] LeCun Y, Bengio Y, Hinton G. Deep learning [J]. Nature, 2015, 521(7553):436-444.
- [6] Yan M, Zhao H D, Li Y H, et al. Multi-classification and recognition of hyperspectral remote sensing objects based on convolutional neural network [J]. Laser & Optoelectronics Progress, 2019, 56(2): 021702.
闫苗, 赵红东, 李宇海, 等. 基于卷积神经网络的高光谱遥感地物多分类识别 [J]. 激光与光电子学进展, 2019, 56(2): 021702.
- [7] Liu Y Z, Jiang Z Q, Ma F, et al. Hyperspectral image classification based on hypergraph and convolutional neural network [J]. Laser & Optoelectronics Progress, 2019, 56(11):111007.
刘玉珍, 蒋政权, 马飞, 等. 基于超图和卷积神经网络的高光谱图像分类 [J]. 激光与光电子学进展, 2019, 56(11):111007.
- [8] Leng J B, Li T, Bai G, et al. Cube-CNN-SVM: a novel hyperspectral image classification method [C]// 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI). 6-8 Nov. 2016, San Jose, CA, USA. New York: IEEE Press, 2016: 1027-1034.
- [9] Li P, Guan G X, Wu T X, et al. GPU optimization of hyperspectral real-time anomaly detection based on Cholesky decomposition [J]. Transducer and Microsystem Technology, 2019, 38(3):7-10.
李萍, 关桂霞, 吴太夏, 等. 基于 Cholesky 分解的高光谱实时异常探测的 GPU 优化 [J]. 传感器与微系统, 2019, 38(3):7-10.
- [10] Jiang X R, Wei W. Application of CUDA technology in digital image matching [J]. Modern Information Technology, 2019(18):61-63.
姜雪茸, 魏威. CUDA 技术在数字图像匹配中的应用 [J]. 现代信息科技, 2019(18):61-63.
- [11] Shen T, Hu F. Acceleration of CNN on GPU [J]. Applications of IC, 2017(6):18-22.
沈恬, 胡飞. 卷积神经网络在图形处理 GPU 芯片上的优化 [J]. 集成电路应用, 2017(6):18-22.
- [12] Dong H, Li T, Leng J B, et al. GCN: GPU-based cube CNN framework for hyperspectral image classification [C]//2017 46th International Conference on Parallel Processing (ICPP). 14-17 Aug. 2017, Bristol, UK. New York: IEEE Press, 2017:41-49.
- [13] Nakasato N. A fast GEMM implementation on the cypress GPU [J]. ACM SIGMETRICS Performance Evaluation Review, 2011, 38(4):50-55.