DOI:10.19651/j.cnki.emt.2416535

交通速度预测时空图卷积网络及其FPGA实现研究*

谭会生 杨 威 严舒琪
(湖南工业大学轨道交通学院 株洲 412000)

摘 要:时空图卷积网络(STGCN)通过图卷积和时间卷积捕获交通数据的空间依赖性和时间依赖性,可有效提升交通速度预测的精度。但是硬件实现交通速度预测STGCN具有计算量大难以满足实际应用的实时性要求、资源消耗大导致成本增高等问题,在优化交通速度预测STGCN模型基础上,提出了一种交通速度预测STGCN的FPGA实现结构组合优化的方法。首先,通过轻量化裁剪和预测数据位宽的精确选择,对交通速度预测STGCN进行了模型优化,以降低计算复杂度和资源消耗,并经过Python仿真验证其可行性。其次,通过采用流水线、并行计算和数据交替流水存取等组合优化策略,提出了一种交通速度预测STGCN的FPGA实现结构组合优化的方法,以提升系统计算速度。最后,使用Verilog编程对交通速度预测STGCN进行了FPGA的实现仿真和硬件测试。利用PeMSD7(M)数据集进行实验,结果显示FPGA实现单数据交通速度预测的时间为355.5 μs,相比CPU、GPU平台及FPGA设计方案1对比,其处理速度最大分别提高了25.9倍、6.7倍和3.5倍,证明了交通速度预测STGCN的FPGA实现结构组合优化方法,在保持预测准确性的前提下可较大幅度的提升系统处理速度。

关键词:交通速度预测;时空图卷积网络;FPGA;硬件实现结构;流水线;并行结构
中图分类号:TN791 **文献标识码**:A **国家标准学科分类代码**:510.4030

Research on spatio-temporal graph convolutional network for traffic speed prediction and their FPGA implementation

Tan Huisheng Yang Wei Yan Shuqi
(College of Railway Transportation, Hunan University of Technology, Zhuzhou 412000, China)

Abstract: Spatio-temporal graph convolutional network (STGCN) enhances the accuracy of traffic speed prediction by capturing the spatial dependencies and temporal dependencies in traffic data through graph convolution and time convolution. However, the hardware implementation of traffic speed prediction using STGCN faces challenges such as high computational demands that do not meet the real-time requirements of practical applications and high resource consumption leading to increased costs. To optimize the traffic speed prediction STGCN model, a method for optimizing the FPGA implementation structure combination of traffic speed prediction STGCN is proposed. Initially, the model is optimized through lightweight pruning and precise selection of prediction data bit-width to reduce computational complexity and resource consumption, verified by Python simulation for feasibility. Subsequently, an optimization strategy using pipeline, parallel computing, and alternating data stream storage is introduced to enhance system computational speed. Finally, the traffic speed prediction STGCN is implemented and tested on FPGA using Verilog programming. Experiments with the PeMSD7(M) dataset show that the FPGA implementation reduces the time for single data traffic speed prediction to 355.5 μs, maximum processing speed increases of 25.9×, 6.7× and 3.5× compared to CPU, GPU platform and FPGA design option 1 comparisons, respectively, proving that the proposed method significantly improves processing speed while maintaining prediction accuracy.

Keywords: traffic speed prediction; spatio-temporal graph convolutional network; FPGA; hardware implementation structure; pipeline; parallel structure

0 引 言

近年来,深度学习与神经网络的研究取得了令人瞩目的进展,而随着大数据时代的到来,各种复杂的数据结构和模式不断涌现。其中,图结构因其在表现对象间复杂联系方面的高效性,已成为许多研究领域的关注重点。图卷积

收稿日期:2024-07-26
*基金项目:湖南省教育厅科学研究重点项目(20A163)、湖南省学位与研究生教学改革研究项目(2022JGYB183)资助

网络[1](graph convolutional network,GCN)作为一种专门处理图结构数据的深度学习模型,具有巨大的潜力和广泛的应用前景,已经被应用于交通预测[2-4]、多标签图像识别[5-6]、骨架动作识别[7-8]等领域。但由于图结构数据的复杂性和大规模性,传统的计算框架在处理大规模图数据时面临计算效率低和资源消耗大的问题,尤其是在交通预测和计算机视觉等实时性要求较高的应用场景中。现有的计算架构难以满足这些需求,因此,对神经网络进行硬件加速的研究具有重要意义。

交通预测是智能交通系统(intelligent transportation systems,ITS)领域的关键任务,对城市交通控制和引导至关重要[9-10]。在交通预测方面,Yu 等[11]提出的时空图卷积网络(spatio-temporal graph convolutional network,STGCN)模型首次将 GCN 应用于交通领域中的时间序列预测问题。Guo 等[12]针对现有的大多数交通流预测方法在动态时空相关性建模方面的不足,提出了基于注意力的时空图卷积网络(attention spatio-temporal graph convolutional network,ASTGCN)模型来解决交通流预测问题。Li 等[13]中提出的时空融合图神经网络(spatio-temporal fusion graph neural network,STFGNN)通过生成"时间图"和融合操作,有效学习了隐藏的时空依赖性,同时通过集成融合图模块和门控卷积模块来处理长序列数据,实现了更加准确的交通流预测。尽管这些研究在提高交通预测准确性方面取得了显著进展,但主要集中于网络结构的优化,而忽略了计算效率的提升。

现场可编程门阵列(field programmable gate array,FPGA)作为一种灵活且高效的硬件计算平台,在多个应用领域中扮演着重要角色[14-15],具有并行处理和定制化设计的优势。GCN 作为近年来兴起的神经网络,是一种典型的计算密集型任务,FPGA 为其提供了部署方案,受到国内外学者的广泛关注。Zhang 等[16]通过开发了消减冗余的边连接和节点重新排序技术来增加数据局部性并减少特征聚合中的冗余计算,提出了一种通用的 FPGA 架构来流水线化 GCN 中的两个主要计算内核,但其并未考虑沿特征维度的图分区,因此会导致较大的内存流量问题。Zhang 等[17]提出以分区为中心的特征聚合方案以及优化的任务调度策略来加速 GCN 模型的前向推理,减少了内存流量并提高了执行效率,但其发生流水线停顿时,采取了将计算单元的输出写回到外部存储器的方式,这增加了外部存储器的访问频率,从而影响了系统实时性。Nair 等[18]通过自定义数据流、统一计算核心和软件预处理来优化 GCN 在 FPGA 平台的前向推理,提高了数据重用和内存访问效率,但由于设计中依赖于特定的对称性假设和定制的数据流,不适用于特定的图结构。在现有研究中,尽管 FPGA 对 GCN 的加速效果已有显著提升,但优化方案主要集中在 GCN 的部署上,并应用于社交媒体分析和学术文献分析等领域,尚未涉及到 STGCN 的 FPGA 加速实现。交通速度预测作为智能交通领域中的关键任务,涉及大量的时序和空间数据,其数据特点与 GCN 处理的数据存在显著差异。STGCN 作为 GCN 的扩展网络,专门用于处理包含时序信息的图数据,其计算复杂度和内存需求显著增加。其研究难点在于如何有效结合时序信息进行图卷积操作,以及如何在 FPGA 上处理高复杂度的时序信息。因此,目前对于 STGCN 在交通速度预测的 FPGA 硬件加速方案还尚未有研究。将 STGCN 部署在 FPGA 上并应用于交通速度预测,可以提高实时处理能力和分析交通数据,从而提升交通管理效率,减少交通拥堵和交通事故,并推动智慧城市系统的发展,具有重要的实际应用价值。

本文针对交通速度预测 STGCN 模型优化及其 FPGA 实现进行了研究,通过对模型结构进行调整,实现了模型的轻量化和预测数据位宽的精确选择,从而降低计算复杂度和资源消耗。采用流水线结构、并行计算技术和数据交替流水存取设计等优化策略,有效提升了模型计算速度。使用 Verilog 硬件描述语言(hardware description language,HDL)在 FPGA 上部署了交通速度预测 STGCN 模型并进行了仿真和验证。采用 PeMSD7(M)数据集对交通速度预测 STGCN 的 FPGA 优化结构进行处理速度的验证。

1 交通速度预测时空图卷积网络模型与优化

1.1 图卷积网络原理

GCN 是当前图神经网络(graph neural network,GNN)中最常见的模型之一,主要通过卷积操作来实现对邻居节点信息的聚合与节点特征的更新。在 GCN 中,信息聚合:通过网络捕捉和利用图中节点之间的局部连接模式,将每个节点的邻居信息通过加权平均的方式进行综合。特征更新:融合邻居信息和节点特征,通过变换更新特征,使其包含自身和邻居节点信息。两层 GCN 模型表示为式(1):

$$\boldsymbol{f}(\boldsymbol{X},\boldsymbol{A}) = \sigma(\hat{\boldsymbol{A}}Relu(\hat{\boldsymbol{A}}\boldsymbol{X}\boldsymbol{W}_0)\boldsymbol{W}_1) \tag{1}$$

其中,$\boldsymbol{X}$ 为特征矩阵,$\boldsymbol{A}$ 为邻接矩阵,$\hat{\boldsymbol{A}} = \tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}} - \frac{1}{2}$ 为预处理步骤,$\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}_N$ 是具有附加自连接无向图的邻接矩阵。$\tilde{\boldsymbol{D}} = \sum_j \tilde{\boldsymbol{A}}_{ij}$ 为对角矩阵,其元素值表示为各个节点的度数。预处理步骤的目的是对邻接矩阵进行归一化,使得拉普拉斯矩阵的特征值在合理的范围内。$\boldsymbol{W}_0$ 和 $\boldsymbol{W}_1$ 是特定层的可训练权重矩阵,$\sigma(\cdot)$ 和 $Relu(\cdot)$ 为激活函数。

1.2 时空图卷积网络结构

一种通用的时空图神经网络(spatio-temporal graph neural network,STGNN)结构如图 1 所示,GCN 作用于输入和邻接矩阵以捕获空间相关性,一维卷积沿时间轴滑动以捕获时间相关性。STGNN 核心在于同时处理空间和时间的依赖关系。许多研究通过 GCN 结合卷积神经网络

(convolutional neural network，CNN)或循环神经网络(recurrent neural network，RNN)的方法来捕获空间依赖性，进而对时间依赖性进行建模。

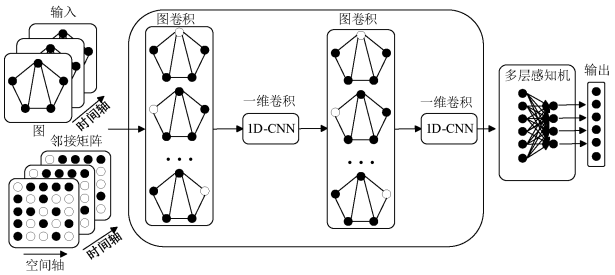


图 1　STGNN 结构

Fig. 1　STGNN structure

本文在分析现有网络结构的基础上，选用 STGCN 结构[11]进行交通速度预测，如图 2 所示。该结构由两个时空卷积块及一个输出层组成。其中时空卷积块含有两层时间卷积和一层空间图卷积，而空间图卷积层作为桥梁连接两层时间卷积，实现空间图卷积到时间卷积的高效空间状态转移。在每个块的内部，采用了残差连接和瓶颈设计策略。为了防止过拟合，每个时空卷积模块都引入了层归一化技术。输入(V_{t-m}，…V_t)由时空卷积块统一处理，以一致地探索空间和时间相关性。通过输出层集成综合特征以生成最终预测 V_{t+T}。

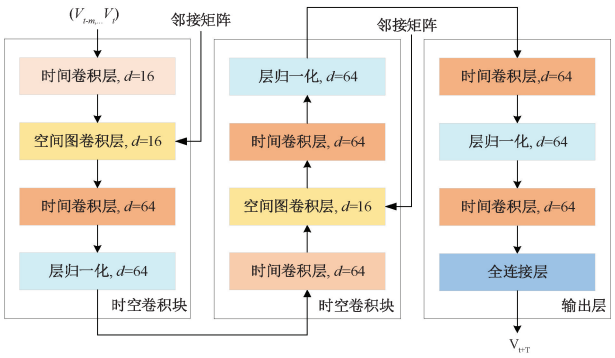


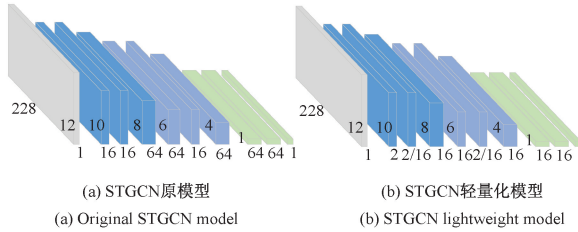
图 2　STGCN 结构

Fig. 2　STGCN structure

1.3　交通速度预测时空图卷积网络模型优化改进

为了在 FPGA 上更有效地实现网络加速，采用了轻量化裁剪、更换传播模型和优化算法及超参数调优等策略。先通过轻量化裁剪减少特征维度来实现网络规模的缩小，以降低计算复杂度和资源消耗；然后通过更换传播模型和优化算法及超参数调优等策略，保证了预测的准确性。

1)轻量化裁剪：初始输入有 228 个节点，12 个观测数据点，1 个特征维度。STGCN 原模型如图 3(a)所示，第一层时间卷积的特征维度为 16 个，其余时间卷积层均为 64 个特征维度，空间图卷积层的聚合和更新为 16 个特征维度。STGCN 轻量化模型如图 3(b)所示，第一层时间卷积的特征维度为 2 个，其余时间卷积层均为 16 个特征维度，空间图卷积层的聚合为 2 个特征维度，更新为 16 个特征维度。



(a) STGCN原模型　(b) STGCN轻量化模型

(a) Original STGCN model　(b) STGCN lightweight model

图 3　STGCN 原模型及轻量化模型

Fig. 3　STGCN original model and lightweight model

2)更换传播模型：STGCN 原模型采用切比雪夫滤波器，传播公式如式(2)所示。STGCN 轻量化模型采用重归一化技巧，传播公式如式(3)所示。

$$\boldsymbol{X}' = \sum_{k=0}^{K} T_k(\tilde{\boldsymbol{L}})\boldsymbol{X}\boldsymbol{\Theta}_k \tag{2}$$

$$\boldsymbol{X}' = \tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}} - \frac{1}{2}\boldsymbol{X}\boldsymbol{\Theta} \tag{3}$$

其中，$\boldsymbol{X}$ 为输入节点特征矩阵，$\tilde{\boldsymbol{L}}$ 为归一化的图拉普拉斯矩阵，$T_k(\tilde{\boldsymbol{L}})$ 为第 k 阶切比雪夫多项式在归一化图拉普拉斯矩阵 $\tilde{\boldsymbol{L}}$ 上的应用，$\boldsymbol{\Theta}_k$ 为第 k 阶多项式对应的滤波器参数矩阵，$\tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}} - 1/2$ 归一化后的拉普拉斯矩阵，$\boldsymbol{X}'$为输出节点特征矩阵。

3)优化器配置和超参数调优：选择 Adam 优化算法作为优化器后，经过超参数调优，设置训练轮次为 50，批量大小为 50，初始学习率为 0.005，每 5 轮衰减率为 0.7。

为了测量 STGCN 轻量化模型的预测性能，使用 3 个常用评价指标来评估真实交通信息 y_i 与预测 $\hat{y}_i$ 之间的差异，具体计算公式如式(4)～(6)所示。

(1)平均绝对百分比误差(mean absolute percentage error，MAPE)：

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{4}$$

(2)平均绝对误差(mean absolute error，MAE)：

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5}$$

(3)均方根误差(root mean squared error，RMSE)：

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{6}$$

其中，MAPE 适用于评估不同尺度数据的误差。MAE 反映预测误差的平均水平。RMSE 适用于评估预测误差的整体波动性。具体来说，MAPE、MAE 和 RMSE 这 3 个评价指标值越小，预测效果越好。

经实验验证，采用 PeMSD7(M)数据集中 1 340 组数据进行测试，STGCN 轻量化模型测试结果如表 1 所示。表 1 中的中央处理器(central processing unit，CPU)和图形处理

器(graphics processing unit,GPU)分别为对应 Python 环境下进行交通速度预测的执行时间,其中:CPU:12th Gen Intel(R) Core(TM) i7-12700H,2.3 GHz。GPU:NVIDIA GeForce RTX 3060 Laptop。

表 1 STGCN 轻量化模型测试结果

Table 1 STGCN lightweight model test results

预测时间/min	MAPE/%	MAE	RMSE	CPU 执行时间/s	GPU 执行时间/s
5	2.929	1.314	2.190	1.129	0.295
10	4.320	1.883	3.281	2.209	0.482
15	5.355	2.293	4.094	3.164	0.621
20	6.190	2.619	4.738	4.111	0.779
25	6.883	2.889	5.261	5.332	0.991
30	7.484	3.124	5.713	6.417	1.184
35	8.021	3.338	6.119	7.651	1.351
40	8.506	3.536	6.488	8.639	1.543
45	8.957	3.721	6.830	9.857	1.739

为了评估 STGCN 轻量化模型的预测性能,选择与以下 3 种基线模型进行比较,如表 2 所示。(1)图卷积门控循环单元(graph convolutional gated recurrent unit,GCGRU)[19];(2)一阶近似模型时空图卷积网络 STGCN(1st)[11];(3)切比雪夫多项式近似模型时空图卷积网络 STGCN(Cheb)[11]。

表 2 不同方法在数据集 PeMSD7(M)上的性能比较

Table 2 Performance comparison of different methods on dataset PeMSD7(M)

时间/min	评价指标	GCGRU	STGCN(1st)	STGCN(Cheb)	STGCN 轻量化
15	MAPE/%	5.54	5.24	5.26	5.36
	MAE	2.37	2.26	2.25	2.29
	RMSE	4.21	4.07	4.04	4.09
30	MAPE/%	8.06	7.39	7.33	7.48
	MAE	3.31	3.09	3.03	3.12
	RMSE	5.96	5.77	5.70	5.71
45	MAPE/%	9.99	9.12	8.69	8.96
	MAE	4.01	3.79	3.57	3.72
	RMSE	7.13	7.03	6.77	6.83

表 2 记录了 STGCN 轻量化模型和其他 3 个基准模型在 PeMSD7(M)数据集上 15 min、30 min 和 45 min 交通速度预测的性能。从表中可以看出,STGCN 轻量化模型在 MAPE、MAE 和 RMSE 这 3 个评价指标上的表现与 STGCN(1st)和 STGCN(Cheb)模型相近。例如,在 15 min 的预测中,尽管 STGCN 轻量化模型的 MAPE 略高为 5.36%,相比 STGCN(1st)和 STGCN(Cheb)的 5.24%和 5.26%,差异仅为 0.1%左右,表明模型在精度上的轻微损失可以忽略不计。同样,在 30 min 和 45 min 的预测中,STGCN 轻量化模型的 MAE 和 RMSE 指标也仅比基线模型稍有增加,但整体仍保持了高水平的预测精度。

在网络规模缩小的同时,通过优化传播模型、更换激活函数以及超参数调优等策略,有效地平衡了模型的复杂性与预测准确性。经过轻量化设计使 FPGA 的计算量减少了约 70%,极大地提升了模型在资源受限环境中的适用性。

1.4 数据量化位数的优化

为了进一步在 FPGA 上提升网络模型的加速性能,对模型参数进行了量化,将原始模型中以 32 位浮点数表示的参数转换成定点数表示。量化过程涉及到两个层面:一是对数据的小数部分进行量化,二是确定整个数值的总比特数,包括符号位、整数部分及小数部分。

经实验验证,小数部分量化后的误差与量化前的误差相减结果如图 4 所示,将小数部分量化至 11 位时仅在 MAE 预测的第 15/30 min 比 12 位略高一点,其他时间点的误差值均低于 10 位和 12 位。因此综合考虑资源消耗和预测准确率,选择将整体数值量化为 16 位定点数表示,其中包括 1 位符号位、4 位整数位及 11 位小数位。

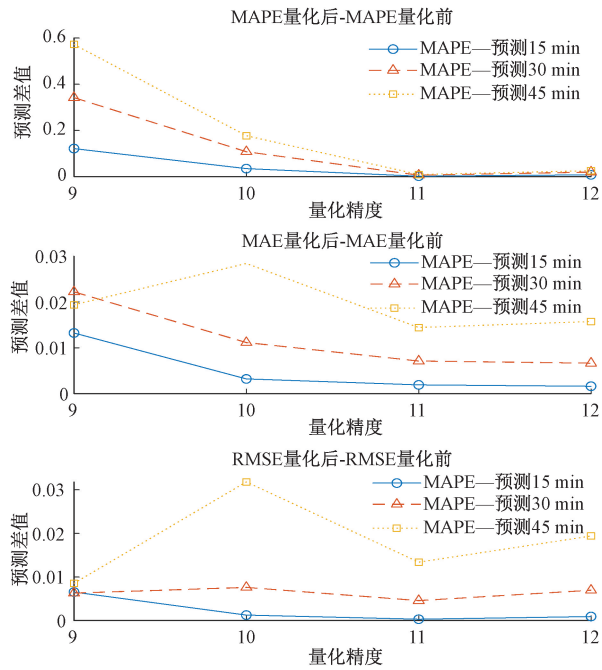


图 4 数据量化验证结果

Fig. 4 Data quantification validation results

2 交通速度预测时空图卷积网络的 FPGA 设计

本文使用 Vivado 2018.3 作为开发和设计的核心平台,并使用 Verilog HDL 作为主要的硬件设计描述语言。

经过量化和 z_score 标准化处理后交通速度数据以及训练后的权重和偏置，均用 16 位有符号定点数的格式表示，其中低 11 位为小数部分，而最高位作为符号位。

2.1　交通速度预测时空图卷积网络的 FPGA 实现结构设计

交通速度预测 STGCN 的 FPGA 实现结构如图 5 所示。主要包括时间卷积、聚合、更新、层归一化等模块。首先，在数据输入阶段，速度数据通过输入数据选择存入伪双端口随机储存器(random access memory，RAM)的 dina 端口中，随后从 dinb 端口读出，经过输出数据选择再进行时间卷积后进入聚合模块，同时将乘加运算后数据存储于 RAM 中。聚合模块输出结果后，进行更新模块运算，并同时引入聚合前的数据进行综合运算。接着，进行时间卷积和层归一化运算。时间卷积模块用于对多个时间点的特征数据进行累加，层归一化模块用于降低神经网络中内部协变量偏移。随后，再经过一个时空卷积块处理。最后，在输出层中时间卷积模块将模型输出转换为单步预测，实现对最终输出的提取，而后在通道上执行线性变换，从而能够有效地计算出节点的速度预测，再将其存储在伪双端口 RAM 中用于进行下一个时间点的速度预测。

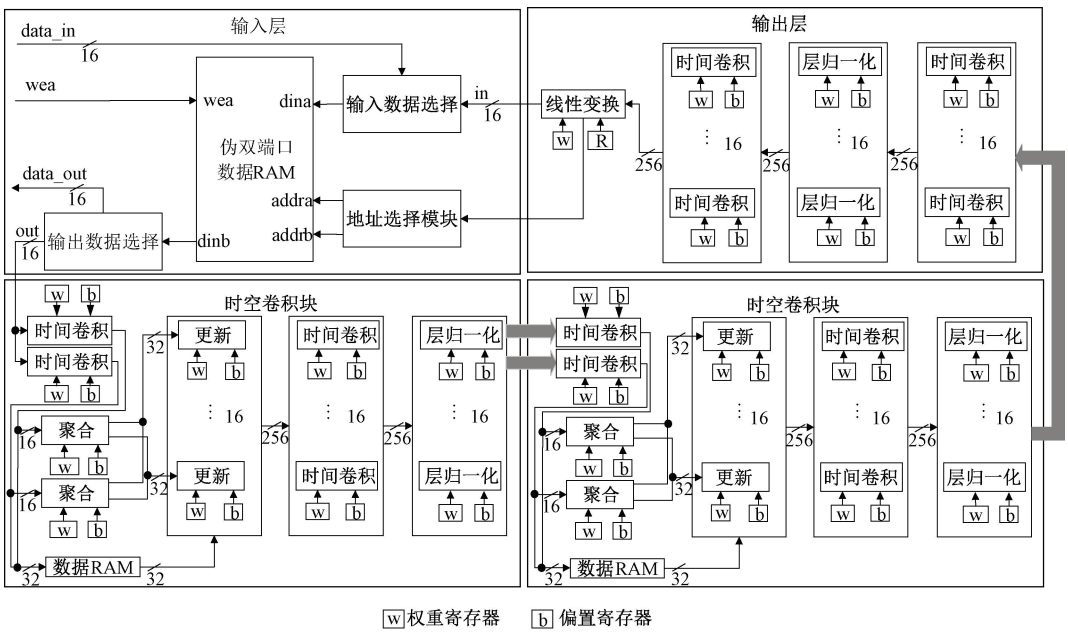


图 5　交通速度预测 STGCN 的 FPGA 实现结构

Fig. 5　Structure of FPGA implementation of STGCN for traffic speed prediction

2.2　卷积模块设计

卷积模块采用时序逻辑电路，并且使用符合特征维度及输入输出维度的高度并行化设计，单个周期可输出 256 个乘法结果。当开始信号 start 有效时，256 位输入数据 data_in 通过位拆分处理，得到 16 个 16 位数据 in[0]－in[15]分别同时进入到乘法器模块，4 096 位权重数据 w_in 通过位拆分处理，得到 256 个 16 位数据 w[0]－w[255]，w[0]－w[15]对应第一个输出维度，w[16]－w[31]第二个输出维度，以此类推。将相应的权重 w 输入到乘法器模块。下面只说明第一个输出维度的乘法结果，其他维度类似。当得到 a[0]－a[15]后，考虑到时序的设计采用加法树的形式进行累加，每进行 3 个加法器相加就进行一次缓存。经过验证并给予足够的整数位以保证数据进行累加不会溢出。得到最终的累加输出进行截位，由于使用了残差连接，在下一个周期加上相应的输入数据 in[0]和偏置参数 bias[0]，再输入 Rule 激活函数处理。当输入小于 0 时，输出为 0；当输入大于 0 时，输出等于输入。卷积模块设计如图 6 所示。

时间卷积模块设计如图 7 所示。当得到卷积的输出结果后，再进行时间轴上卷积，输入数据是沿着空间轴进行输入，第 1 个时间点卷积结果需要等待 228 个时钟周期才能与对应的第 2 个时间点卷积结果进行相加并输出，每个时间点的权重 w 不一样，也需要经过 228 个时钟周期进行输出新的权重 w。

2.3　图卷积模块设计

在 PeMSD7(M)数据集中，用交通网络中节点之间的实际距离来计算道路图的邻接矩阵。其加权邻接矩阵 $\boldsymbol{W}$ 的计算如式(7)所示。

$$\boldsymbol{w}_{ij}=\begin{cases}\exp\left(-\dfrac{d_{ij}^2}{\sigma^2}\right), & i\neq j \text{ 和 } \exp\left(-\dfrac{d_{ij}^2}{\sigma^2}\right)\geqslant\epsilon\\ 0, & \text{其他}\end{cases}\tag{7}$$

其中，权重 $\boldsymbol{w}_{ij}$ 是由节点 i 和 j 之间的实际距离 d_{ij} 决定的。σ^2 和 ϵ 分别作为控制加权邻接矩阵 $\boldsymbol{W}$ 的分布和稀疏性的参数，分别指定为 10 和 0.5。按照上述设计，通过计

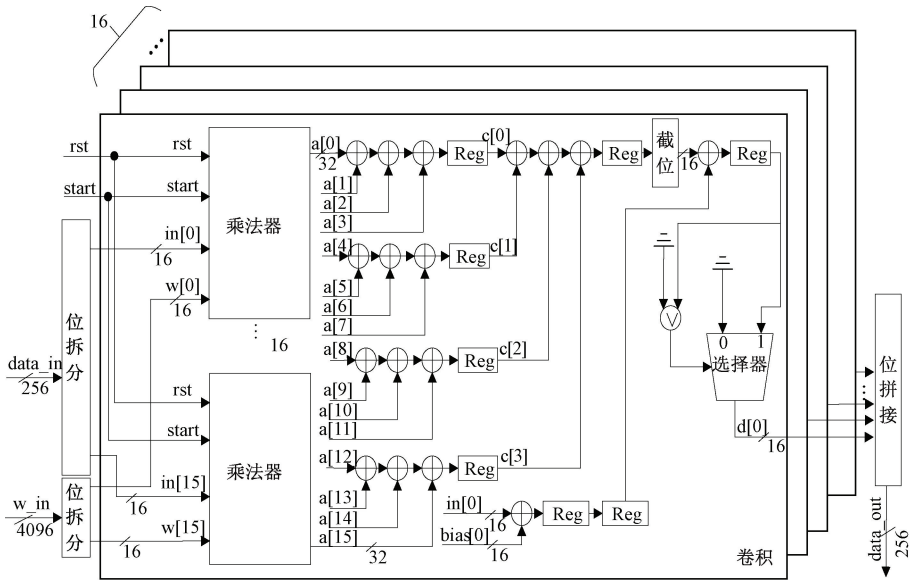


图 6 卷积模块的设计

Fig. 6 Design of convolution module

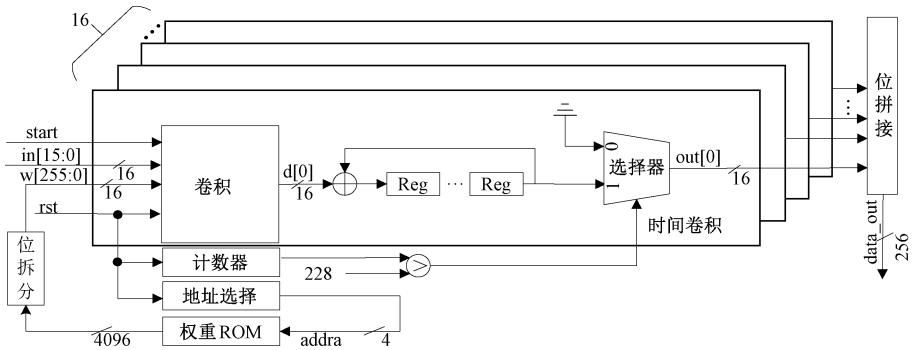


图 7 时间卷积模块的设计

Fig. 7 Design of time convolution module

算得出 $\boldsymbol{w}_{ij}$ 属于密集矩阵。

聚合模块使用高度并行化设计,当 start 信号为高时,通过时序设计输入数据和权重数据对应处理,乘法器输出结果。GCN 操作是矩阵相乘,其中最终结果的每个元素 $w[i,j]$ 都包含了输入特征矩阵的第 i 行与邻接矩阵矩阵的第 j 列的点积。因每个时钟周期只输入一个数据,需要 228 个周期后得到聚合结果。得到聚合结果后,按照时序设计输出。由于更新模块要用到聚合前的输入数据,因此将其存入 RAM 中。聚合模块设计如图 8 所示。

更新模块设计如图 9 所示。考虑到整体推理加速,更新模块也采用高度并行化设计,由于应用了瓶颈策略和剩余连接,需要将特征维度增加至 16,并且需要将聚合前的输入数据从 RAM 取出,由于输入和输出的特征维度不同,只需要将聚合前的输入数据加到第 1 个和第 2 个特征维度即可,得出结果后再进行 Rule 激活函数。

2.4 层归一化模块设计

层归一化(layer normalization,LN)是深度学习中用于规范神经网络每个层输出的一种技术。其主要目标是减小神经网络中内部协变量转移的影响,从而提高网络的稳定性和训练速度。层归一化的计算过程可用式(8)~(10)表示:

$$\mu = \frac{1}{M}\sum_{i=1}^{M} x_i \tag{8}$$

$$\sigma^2 = \frac{1}{M}\sum_{i=1}^{M} (x_i - \mu)^2 \tag{9}$$

$$LN(x) = \frac{x-\mu}{\sqrt{\sigma^2+\epsilon}} \times \gamma + \beta \tag{10}$$

其中,μ 是输入 x 在该层的均值,σ^2 是输入 x 在该层的方差,M 是节点个数,γ 和 β 是可学习的参数,分别用于对归一化后的数据进行缩放和平移。

在层归一化模块的设计中,由于均值和方差是基于当前输入层值计算而得,因此需要在计算完成前对输入数据进行存储至数据 RAM 中。此外,由于均值和方差在不同时间步计算时会有不同的取值,因此需要根据输入数据的

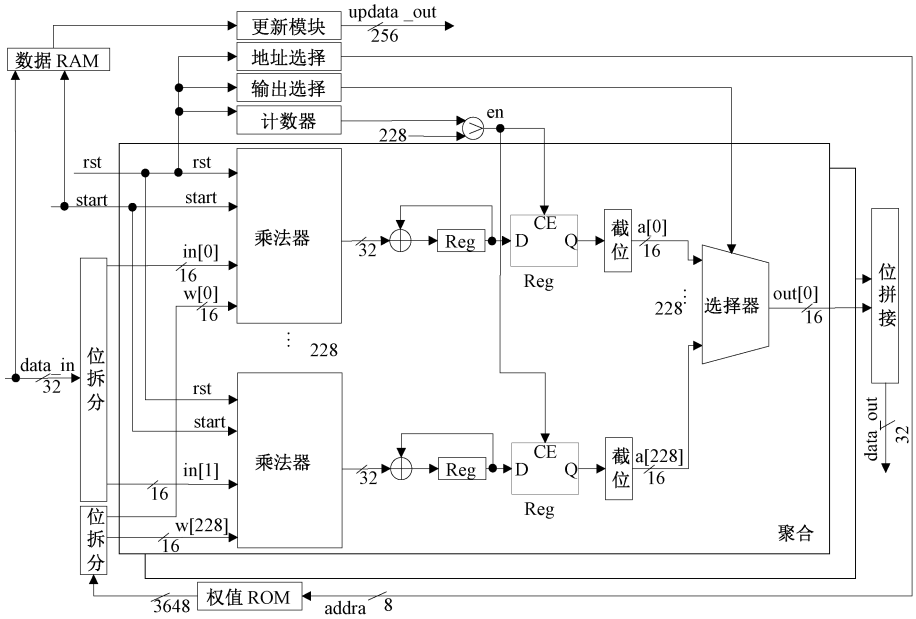


图 8　聚合模块的设计

Fig. 8　Design of the aggregation module

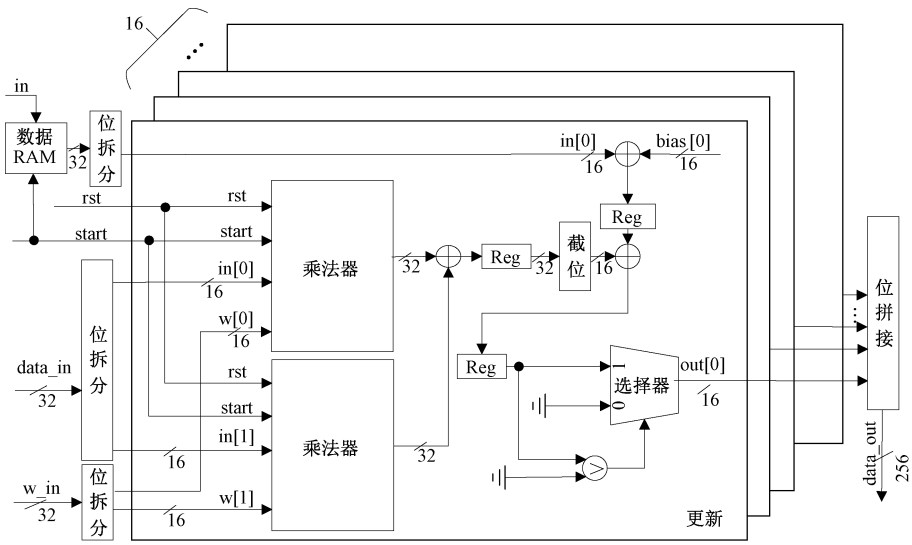


图 9　更新模块的设计

Fig. 9　Design of the update module

时序进行输出调整。考虑到时序设计，其中累加操作都采用加法树形式。首先进行特征维度的并行相加操作，其中特征维度为 16。累加得到结果后采用加法树形式，然后右移 4 位。在当前层所有节点累加完成后，进行除法操作以得出该层的均值。由于输入数据已通过寄存器缓存，因此可以同时对所有数据进行计算，以提高推理速度。接着，将输入数据减去均值后进行平方运算，之后进行截位操作。采用加法树形式进行累加，累加结果再右移 4 位。同样，在当前层所有节点累加完成后，进行除法操作，得出该层的方差。均值和方差归一化模块设计如图 10 所示。

通过此设计，可以有效实现均值和方差的计算，而参数 γ 和 β 是通过训练过程中学习直接得到的，然后通过简单的加减乘除运算即可得到层归一化的结果。

2.5　数据存取优化设计

为了实现输出结果的存储，并将其用于下一个时间段的预测，选择了分布式伪双端口 RAM 作为主要的存储器。将 228×12 的交通速度数据沿时间轴展开成 2 736×1 的向量，并存储于伪双端口 RAM 中。采用交替流水线的数据存取方法[20]，既提高了处理速度，又有效减少了资源消耗。

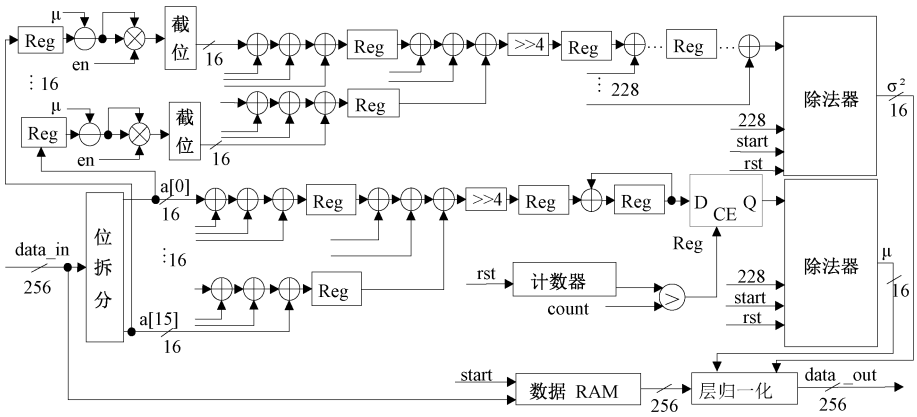
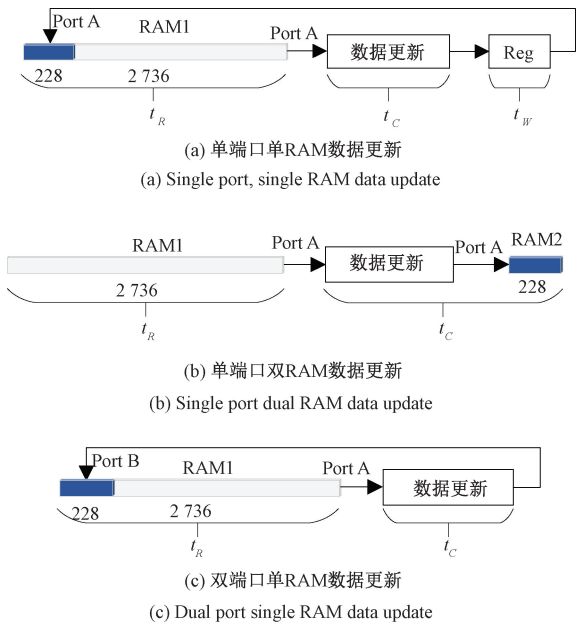


图 10　均值和方差归一化模块的设计

Fig. 10　Design of the mean and variance normalisation module

采用单端口 RAM 进行数据更新有两种主要方式。首先是单 RAM 方式图 11(a)所示,在数据更新过程中,旧数据首先从 RAM1 中读出,然后进行计算得到最终输出结果后临时存储在寄存器。由于最初使用的 228 个数据点在第一个时间段后将不再被使用,这些新计算出的数据便会被重新写回 RAM1 中。尽管这种方法仅使用单个 RAM,但其整体所需时间为 $t_R+t_C+t_W$。另一种方式是双 RAM 方式图 11(b)所示,首先从 RAM1 中读取旧数据,通过计算得出最终结果后,将其写入 RAM2 中。由于可以同时从 RAM1 读取数据和向 RAM2 写入数据,总耗时为 t_R+t_C。然而,这种方法虽然在时间上更高效,却需要占用两个 RAM。为了提高数据存取速度并优化资源使用,采用了图 11(c)所示的数据交替流水存取设计,并通过伪双端口 RAM 实现。在数据更新过程中,旧数据通过端口 A 从 RAM1 读出进行计算得到最终结果后,将其覆盖最初使用的 228 个数据点。总耗时为 t_R+t_C,且只需使用单个 RAM。



(a) 单端口单RAM数据更新

(a) Single port, single RAM data update

(b) 单端口双RAM数据更新

(b) Single port dual RAM data update

(c) 双端口单RAM数据更新

(c) Dual port single RAM data update

图 11　数据存储优化设计

Fig. 11　Data storage optimisation design

3　交通速度预测时空图卷积网络的 FPGA 系统仿真与测试

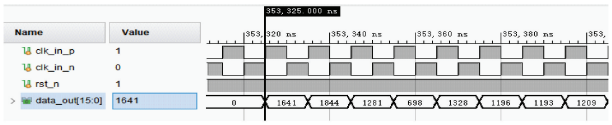
利用前述交通速度预测 STGCN 的 FPGA 实现结构,设计交通速度预测 FPGA 系统。该系统采用 Vivado 2018.3 进行综合,利用 Xilinx Vivado Simulantion 进行仿真。FPGA 芯片型号为 xc7z100ffg900-2。

3.1　交通速度预测时空图卷积网络 FPGA 系统仿真

在系统仿真过程中,将训练好的权重参数和交通速度数据保存至.coe 文件中,并将其分别导入到 ROM 和分布式伪双端口 RAM 中。随机选择 PeMSD7(M)测试集的一组数据进行输入,在 100 MHz 时钟条件下,Vivado 仿真结果如图 12 所示,预测未来第 45 min 交通速度最后一个速度数据在 355.6 μs 处完成。

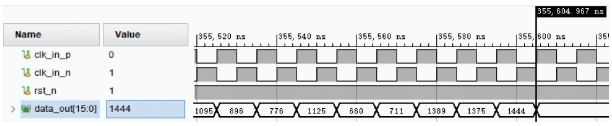
Vivado 仿真结果数据经过反量化和反标准化,可得出最终的速度预测结果如图 13 所示,其中红色为真实值,蓝色为 Vivado 仿真结果的预测值。该数据集的均值为 58.5,标准差为 13.73,分别验算预测第 45 min 交通速度第 1 个节点和第 228 个节点结果:$(1\ 641/(2^{11}))\times 13.73+58.5=69.5$ 和 $(1\ 444/(2^{11}))\times 13.73+58.5=68.18$,可以对应上实际交通结果,仿真结果正确。

选择 PeMSD7(M)数据集中的 268 组速度数据进行输入,其线性回归分析结果如图 14 所示,其中横坐标 T 是期望输出,纵坐标 Y 是实际输出。预测 15 min 时回归系数 R 为 0.949 4,预测 30 min 时回归系数 R 为 0.902 96,预测 45 min 时回归系数 R 为 0.863 84,使用的模型在短期内预测交通速度时表现出较高的准确性,但随着预测时间的增加,其准确性逐渐下降。因为长期预测涉及更多不确定因素,如交通规则变化、突发事件等,增加了预测难度。



(a) 第1个节点至第8个节点Vivado仿真结果

(a) Vivado simulation results from node 1 to node 8

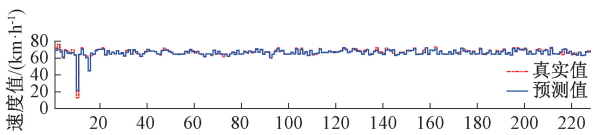


(b) 第221个节点至第228个节点Vivado仿真结果

(b) Vivado simulation results from node 221 to node 228

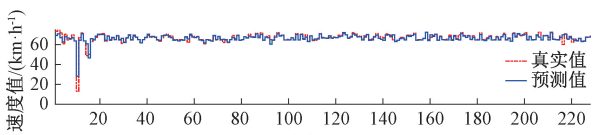
图 12　预测未来第 45 min 交通速度 Vivado 仿真结果

Fig. 12　Predicting traffic speeds at the 45th minute in the future Vivado simulation results



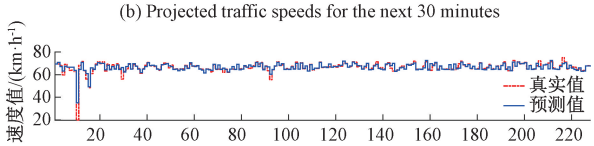
(a) 预测未来15 min交通速度

(a) Projected traffic speeds for the next 15 minutes



(b) 预测未来30 min交通速度

(b) Projected traffic speeds for the next 30 minutes



(c) 预测未来45 min交通速度

(c) Forecasting traffic speeds for the next 45 minutes

图 13　FPGA 系统速度预测仿真结果

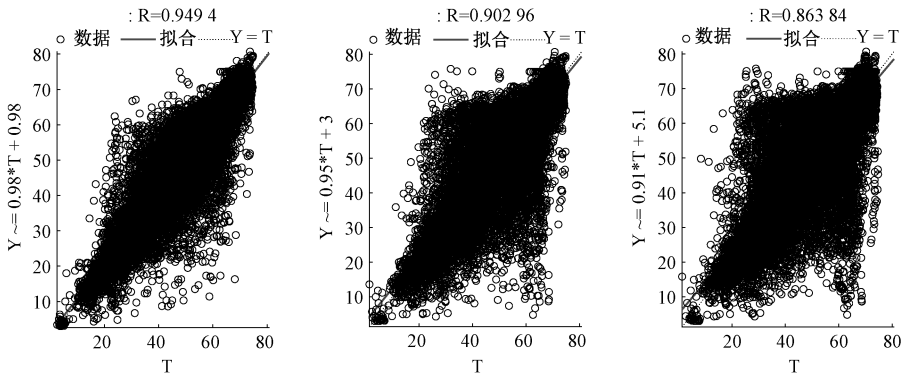
Fig. 13　FPGA system speed prediction simulation results

3.2　交通速度预测时空图卷积网络 FPGA 系统实现

交通速度预测 STGCN 的 FPGA 系统实现如图 15 所示，选取交通速度预测 STGCN 的 FPGA 系统仿真中使用的数据作为输入，液晶屏显示真实交通数据及预测交通数据。其中红色为交通速度数据真实值，蓝色为通过 FPGA 系统运行结果的交通速度数据预测值。从结果分析来看，预测 15/30/45 min 时各节点的交通速度预测值与真实值的差异较小，表明交通速度预测时空图卷积网络 FPGA 系统设计成功，并在不同时间尺度上均具有较高的预测精度。

使用在线逻辑分析仪(Integrated Logic Analyzer，ILA)对 FPGA 系统的运行耗时进行观测，FPGA 系统整体运行耗时如图 16 所示，在 100 MHz 时钟条件下，从第 13 时钟周期开始 FPGA 系统运行，到第 3 272 个时钟周期预测完第 5 min 交通速度，以此类推，到 35 560 个时钟周期预测完第 45 min 交通速度，总耗时为(35 560－13)×10 ns≈355.5 μs。

本文在优化时空图卷积网络模型后，首先实现了 FPGA 设计方案 1，并在此基础上进一步优化，提出并实现了 FPGA 设计方案 2。FPGA 设计方案 1，卷积模块内部 16 个数字信号处理器(digital signal processor，DSP)并行工作，输入 16 个 16 bit 数据并输出单个通道 16 bit 数据，卷积模块间串行工作，因需要长时间缓存中间结果数据，影响系统实时性。采用单端口双 RAM 数据更新策略虽然保证了时间上的高效，但会增加块随机存取存储器(block random-access memory，BRAM)资源消耗。FPGA 设计方案 2，卷积模块内部 16 个 DSP 并行工作，16 个卷积模块间并行工作，从而整体实现特征级并行化，同时得到 16 个通道的卷积计算结果。结构中还引入了深度流水线机制，使得每层的卷积计算结果可以直接传递给下一层，多个层可以同时进行计算，实现层间流水线并行化。为了优化卷积和层归一化过程中的加法操作，设计中采用了加法树结构。加法树能够并行处理多个卷积结果的累加操作，相较



(a) 预测15 min

(a) Predict 15 min

(b) 预测30 min

(b) Predict 30 min

(c) 预测45 min

(c) Predict 45 min

图 14　交通速度预测线性回归图

Fig. 14　Linear regression plot for traffic speed prediction

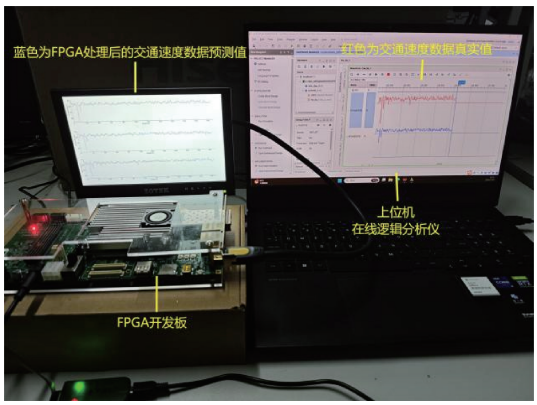


图 15　交通速度预测 STGCN 的 FPGA 系统实现

Fig. 15　FPGA system implementation of STGCN for traffic speed prediction

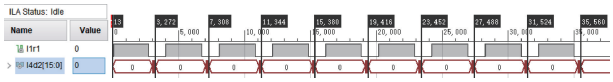


图 16　FPGA 系统整体运行耗时

Fig. 16　Overall running time of FPGA system

于一次性全部相加或串行加法操作,加法树通过分层逐步减少计算时延,提高了整体计算速度,减少了计算延迟。得益于并行流水线设计和加法树的结合,优化后的设计方案有效避免了大规模中间数据的缓存需求。同时,通过引入交替流水线的数据存取优化设计策略,不仅提高了处理速度,还降低了 BRAM 资源消耗。

综上,本文针对两种 FPGA 设计方案进行了验证及对比,FPGA 系统资源消耗如表 3 所示,处理速度如表 4 所示。本文对 FPGA 设计方案 1 进行优化,并经过设计验证实现了 FPGA 设计方案 2,其 DSP 资源消耗增加 960 个为代价,处理速度提升 3 倍,且查找表(lookup table,LUT)、触发器(flip-flop,FF)和 BRAM 资源消耗仅为 FPGA 设计方案 1 的 87.7%、63.9%和 82.9%。

交通速度预测 STGCN 的 FPGA 系统资源占用情况如表 3 所示,包括 LUT、FF、BRAM 和 DSP 等硬件资源。

表 3　FPGA 系统资源占用情况

Table 3　FPGA system resource usage

资源	占用量		可用	占用比/%	
	方案 1	方案 2	数量	方案 1	方案 2
LUT	154 113	135 231	277 400	55.56	48.75
FF	352 364	225 075	554 800	63.51	40.57
BRAM	386	320	755	51.12	42.38
DSP	633	1 593	2 020	31.33	78.86

表 4 给出了软硬件执行时间对比。其中软件执行时间为表 2 中的 CPU 和 GPU 对应 Python 环境下交通速度预测实现的时间再除以 1 340 组数据的执行时间,其中:CPU:12th Gen Intel(R) Core(TM) i7-12700H,2.3 GHz。GPU:NVIDIA GeForce RTX 3060 Laptop。硬件时间为 FPGA:xc7z100ffg900-2 且时钟频率为 100 MHz 环境下交通速度预测实现的执行时间,与现有的 CPU、GPU 及 FPGA 设计方案 1 对比,速度最大分别提高了 25.9 倍、6.7 倍和 3.5 倍。

表 4　软硬件执行时间对比

Table 4　Hardware and software execution time comparison

预测时间/min	T_{CPU}/ms	T_{GPU}/ms	T_{FPGA1}/ms	T_{FPGA2}/ms	加速比/%		
					T_{CPU}/T_{FPGA2}	T_{GPU}/T_{FPGA2}	T_{FPGA1}/T_{FPGA2}
5	0.843	0.220	0.114 8	0.032 6	25.9	6.7	3.5
10	1.649	0.360	0.228 1	0.073 0	22.6	4.9	3.1
15	2.361	0.464	0.341 4	0.113 3	20.8	4.1	3.0
20	3.068	0.581	0.454 7	0.153 7	20.0	3.8	3.0
25	3.979	0.739	0.568 0	0.194 0	20.5	3.8	2.9
30	4.789	0.884	0.681 3	0.234 4	20.4	3.8	2.9
35	5.710	1.008	0.794 6	0.274 8	20.8	3.7	2.9
40	6.447	1.151	0.907 9	0.315 1	20.5	3.7	2.9
45	7.356	1.298	1.021 2	0.355 5	20.7	3.7	2.9

4　结　　论

针对交通速度预测 STGCN 模型优化及其 FPGA 实现进行了深入研究。首先,通过轻量化裁剪和预测数据位宽的精确选择,提出了一种交通速度预测 STGCN 的优化模型,以降低计算复杂度和资源消耗,并经过 Python 仿真验证其可行性。其次,通过采用流水线、并行计算和数据交替流水存取等组合优化策略,提出了一种交通速度预测 STGCN 的 FPGA 实现结构组合优化方法,以提升交通速度预测 STGCN 的计算速度。最后,使用 Verilog 编程对交通速度预测 STGCN 进行了 FPGA 仿真和硬件测试。实验结果表明,使用 PeMSD7(M)数据集,在 FPGA 上实现

单数据交通速度预测的时间只需要 355.5 μs，相比 CPU、GPU 平台及 FPGA 设计方案 1 对比，其处理速度最大分别提高了 25.9 倍、6.7 倍和 3.5 倍，证明了本文提出的交通速度预测 STGCN 的 FPGA 实现结构组合优化方法，在保持预测准确性的前提下可较大幅度的提升处理速度。

参考文献

[1] 王勤凡，翟江涛，陈伟，等. 一种基于图卷积神经网络的加密流量分类方法[J]. 电子测量技术，2022，45(14)：109-115.
WANG Q F，ZHAI J T，CHEN W，et al. A method of encrypted traffic classification based on graph convolutional neural network [J]. Electronic Measurement Technology，2022，45(14)：109-115.

[2] ZHAO L，SONG Y J，ZHANG CH，et al. T-GCN：A temporal graph convolutional network for traffic prediction [J]. IEEE Transactions on Intelligent Transportation Systems，2019，21(9)：3848-3858.

[3] BAI J D，ZHU J W，SONG Y J，et al. A3T-GCN：Attention temporal graph convolutional network for traffic forecasting[J]. ISPRS International Journal of Geo-Information，2021，10(7)：485.

[4] ZHU J W，WANG Q J，TAO CH，et al. AST-GCN：Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting[J]. IEEE Access，2021，9：35973-35983.

[5] CHEN ZH M，WEI X SH，WANG P，et al. Multi-label image recognition with graph convolutional networks[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition，2019：5177-5186.

[6] SINGH I P，OYEDOTUN O，GHORBEL E，et al. IML-GCN：Improved multi-label graph convolutional network for efficient yet precise image classification[C]. AAAI-22 Workshop Program-Deep Learning on Graphs：Methods and Applications，2022.

[7] YAN S J，XIONG Y J，LIN D H. Spatial temporal graph convolutional networks for skeleton-based action recognition [C]. AAAI Conference on Artificial Intelligence，2018.

[8] CHEN Y X，ZHANG Z Q，YUAN CH F，et al. Channel-wise topology refinement graph convolution for skeleton-based action recognition[C]. IEEE/CVF International Conference on Computer Vision，2021：13359-13368.

[9] 殷礼胜，魏帅康，孙双晨，等. 基于 FEEMD-SAPSO-BiLSTM 组合模型的短时交通流预测[J]. 电子测量与仪器学报，2021，35(10)：72-81.
YIN L SH，WEI SH K，SUN SH CH，et al. Short-term traffic flow prediction based on the FEEMD-SAPSO-BiLSTM composite model [J]. Journal of Electronic Measurement and Instrumentation，2021，35(10)：72-81.

[10] 王逸文，王维莉，陈怡霏，等. 基于奇异谱分析的长时交通流混合预测模型[J]. 电子测量与仪器学报，2022，36(11)：98-106.
WANG Y W，WANG W L，CHEN Y F，et al. Hybrid long-term traffic flow prediction model based on singular spectrum analysis [J]. Journal of Electronic Measurement and Instrumentation，2022，36(11)：98-106.

[11] YU B，YIN H T，ZHU ZH X. Spatio-temporal graph convolutional networks：a deep learning framework for traffic forecasting [C]. 27th International Joint Conference on Artificial Intelligence，2018：3634-3640.

[12] GUO SH N，LIN Y F，FENG N，et al. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting[C]. AAAI Conference on Artificial Intelligence，2019，33(1)：922-929.

[13] LI M ZH，ZHU ZH X. Spatial-temporal fusion graph neural networks for traffic flow forecasting[C]. AAAI Conference on Artificial Intelligence，2021，35(5)：4189-4196.

[14] 于希明，彭宇，姚博文，等. 基于 FPGA 并行计算的多阈值分级海陆分割方法[J]. 仪器仪表学报，2022，43(9)：166-177.
YU X M，PENG Y，YAO B W，et al. Multi-threshold level sea-land segmentation method based on FPGA parallel computing [J]. Chinese Journal of Scientific Instrument，2022，43(9)：166-177.

[15] 胡汉昌，王茂森，戴劲松. DSP＋FPGA 的双核串行通信系统设计与实现[J]. 国外电子测量技术，2022，41(8)：125-132.
HU H CH，WANG M S，DAI J S. Design and implementation of a dual-core serial communication system with DSP + FPGA [J]. Foreign Electronic Measurement Technology，2022，41(8)：125-132.

[16] ZHANG B，ZENG H，PRASANNA V. Hardware acceleration of large scale GCN inference[C]. 2020 IEEE 31st International Conference on Application-specific Systems，Architectures and Processors (ASAP). IEEE，2020：61-68.

[17] ZHANG B，KANNAN R，PRASANNA V. Boostgcn：A framework for optimizing gcn inference on fpga[C]. 2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines(FCCM). IEEE，2021：29-39.

[18] NAIR G R, SUH H S, HALAPPANAVAR M, et al. FPGA acceleration of gcn in light of the symmetry of graph adjacency matrix [C]. 2023 Design, Automation & Test in Europe Conference & Exhibition(DATE). IEEE, 2023: 1-6.

[19] LI Y G, YU R, SHAHABI C, et al. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting [C]. International Conference on Learning Representations, 2018.

[20] 谭会生,徐界铭,张驾祥. BP 神经网络 FPGA 实现结构的优化设计[J]. 计算机工程与应用,2022,58(21):264-271.
TAN H SH, XU J M, ZHANG J X. Optimization design of BP neural network FPGA implementation structure[J]. Computer Engineering and Applications, 2022, 58(21): 264-271.

作者简介

谭会生(通信作者),硕士,教授,主要研究方向为 EDA/SOPC 技术、数字 VLSI 系统、嵌入式系统和功率半导体器件。
E-mail:huisheng21nd@163.com

杨威,硕士研究生,主要研究方向为 FPGA 应用、图神经网络应用。
E-mail:yw143hq@163.com

严舒琪,硕士研究生,主要研究方向为 FPGA 应用,神经网络硬件加速。
E-mail:1275091542qq.com