光电技术及应用

DOI：10.16818/j.issn1001－5868.2023111201

一种基于 CNN 的混叠光谱解调算法及 FPGA 实现

任嘉楠[1,2,3]，焦　点[1,2,3]，杨　铎[1,2]，徐春锋[1,2]，辛璟焘[1,2,3]*

（北京信息科技大学 1. 光电测试技术及仪器教育部重点实验室，北京 100192；
2. 光纤传感与系统北京实验室，北京 100016；3. 广州南沙光子感知技术研究院，广州 511462）

摘　要： 为了解决光纤布拉格光栅(FBG)传感网络的光谱信号混叠问题，基于现场可编程门阵列(FPGA)提出了一种利用卷积神经网络(CNN)模型的混叠光谱信号解调算法，并对其进行硬件实现与加速。通过对模型参数进行定点数量化，压缩网络模型的存储空间，提高 FPGA 中 DSP 资源的利用率；利用循环展开和数组重排等硬件优化方法，提高了系统实时性，确定了算法的并行计算方案。研究结果表明，在 100 MHz 的时钟下，测试集解调精度为 1.19 pm，推理速度为每帧 14.96 μs，光谱解调速率为 60 kHz，对于 FBG 混叠光谱信号解调具有较高的精度和速率。

关键词： 光纤光栅；混叠光谱；FPGA；卷积神经网络；硬件加速

中图分类号： TN791　**文章编号：** 1001－5868(2024)02－0295－08

CNN-based Overlapping Spectral Demodulation Algorithm and Its FPGA Implementation

REN Jianan[1,2,3], JIAO Dian[1,2,3], YANG Duo[1,2], XU Chunfeng[1,2], XIN Jingtao[1,2,3]

(1. Key Laboratory of the Ministry of Education for Optoelectronic Measurement Technology and Instrument, Beijing 100192, CHN; 2. Beijing Laboratory of Optical Fiber Sensing and System, Beijing Information Science & Technology University, Beijing 100016, CHN; 3. Guangzhou Nansha Intelligent Photonic Sensing Research Institute, Guangzhou 511462, CHN)

Abstract: To solve the spectral signal overlapping problem in fiber Bragg grating (FBG) sensing networks, this study proposes a spectral signal demodulation algorithm using a convolutional neural network model based on a field-programmable gate array (FPGA) and implements it in hardware for acceleration. The models parameters are quantized to a fixed-point representation, reducing the storage space of the model and enhancing the utilization of DSP resources in the FPGA. Hardware optimization techniques such as loop unrolling and array rearrangement are employed to improve real-time system performance, establishing a parallel computing scheme for the algorithm. The results indicate that under a clock frequency of 100 MHz, the demodulation accuracy of the test set is 1.19 pm at an inference speed of 14.96 μs per frame and a spectral demodulation rate of 60 kHz. The proposed algorithm exhibits high precision and speed in the demodulation of overlapped FBG spectral signals.

Key words: fiber Bragg grating; overlapping spectral; FPGA; convolutional neural network; hardware acceleration

0　引言

光纤传感器技术因其结构紧凑、抗电磁干扰等优点，在结构健康监测、土木工程、生物化学和医疗设备等领域都得到了广泛应用[1]。其中，光纤布拉格光栅(FBG)传感器尤其具有反射信号可控、复用能力强、线性度好等优点[3]。在大规模机载 FBG 传感器网络中[6]，为了同时监测多个位置的实时情况，

收稿日期：2023－11－12.

* **通信作者：**辛璟焘

通常会将多个 FBG 分配到相近波长范围，以最大限度地利用宽带光源的有限带宽。但这会导致同一波长范围的 FBG 发生光谱混叠现象，大幅度降低其解调精度[7]。

近年来，将深度学习模型部署在光纤光栅混叠光谱解调算法中成为研究热点。2021 年，Li 等[8]提出了一种基于扩张卷积神经网络（DCNN）的 FBG 传感器网络信号解调方法，能够实现均方根（RMS）误差小于 0.05 pm 的解调精度，两个 FBG 传感器通道的重叠信号解调时间为 15 ms。尽管卷积神经网络（CNN）在 FBG 信号解调方面表现良好[9]，但目前，过长的信号解调时间仍会导致系统实时性不足。此外，许多研究在中央处理器（CPU）或图形处理器（GPU）这种成本和功耗较高的计算平台上实现，限制了其在算力有限的工程场景，如机载或空载平台中的应用。现场可编程门阵列（FPGA）具有开发周期短、能效高、可重构等优点，同时由于 CNN 的并行化计算特性和相对固定的模型结构，FPGA 成为 CNN的理想实现平台。但其仍然存在一些问题[10]，首先是如何在保证解调精度的同时降低 CNN 模型的复杂度，其次是如何在 FPGA 上合理设计 CNN 的并行计算方案。

本文结合 CNN 压缩方法和 FPGA 硬件加速设计，提出了一种用于混叠光谱信号解调的有效硬件实现方法，该方法使得基于一维 CNN 的信号解调模型能够以较低的硬件资源成本部署在 FPGA 中。在保证解调精度的前提下，发挥 FPGA 的优势，提高混叠光谱信号的解调速度。

1 算法设计与优化

为了在资源有限的边缘设备上实现 CNN 算法并实时解调混叠光谱信号，模型应具有较高的解调精度和较低的复杂度，最终可以通过低成本 FPGA 来满足算法所需要的计算和存储资源。

1.1 算法结构设计

基于上述准则，本文设计了一个轻量级一维 CNN 模型（如图 1 所示），其结构共有 5 层，包含 2 个卷积层、2 个池化层和 1 个全连接层。

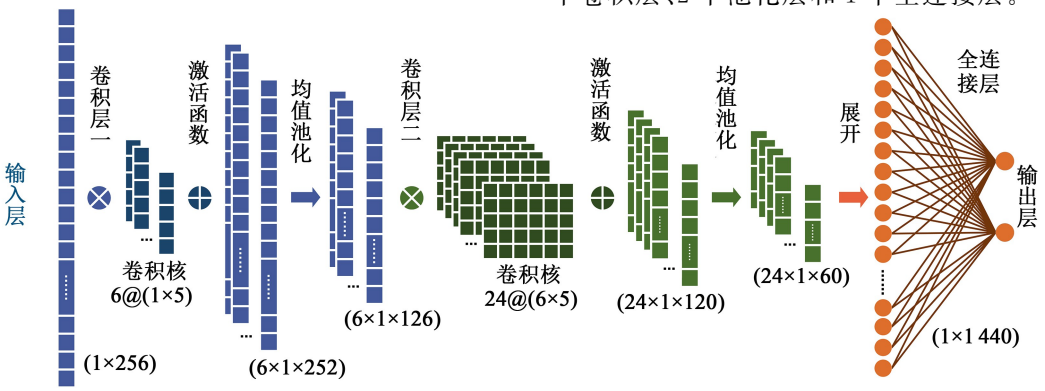


图 1 一维卷积神经网络结构图

一维卷积计算可表示为：

$$Conv_n = B_n + \sum_{1}^{R}\sum_{1}^{C}\sum_{1}^{F} X_n \otimes F \tag{1}$$

式中，B 是偏置参数，X 是每个卷积层的输入数据，K 是卷积核数据，$Conv$ 是卷积计算的结果，$\otimes$代表卷积计算，n 是层数，R 是输入数据尺寸，C 是卷积核数量，F 是卷积核尺寸。

先进行一维卷积计算，激活函数选择线性整流函数（ReLU），经过一维均值池化，最终输出层是一个全连接层。通过卷积计算进行信号处理，能够有效提取输入光谱的信号特征，同时有助于抑制符号间干扰等导致的信号损伤和失真。

1.2 参数量化

进一步降低存储和计算资源最直接的方法就是减少模型参数的位宽。CNN 在 GPU 上通常使用 32 位浮点数据进行训练，比定点数据格式复杂许多，所以量化操作可能会降低计算精度。

参数量化完成后，网络中的所有数据都被设置为相同位宽的定点格式，为了在 FPGA 上高效部署网络，本文采用对称量化方案，将卷积层的特征图和权重数据量化为 16 位定点数，包括 1 位符号位、2 位整数位、13 位小数位。考虑到 N 位对称量化的一般情况，量化函数定义如下：

$$q = \mathrm{clamp}\left(\mathrm{Int}\left(\frac{r}{S}\right),(-2^I+1),(2^I-1)\right) \tag{2}$$

$$N = B + I + F \tag{3}$$

式中，r 表示输入特征矩阵或权重矩阵中的浮点元素，q 表示对应的量化值，S 表示矩阵的缩放因子，

N 表示量化位宽，B 表示定点数中的符号位宽，I 表示整数位宽，F 表示小数位宽，所以量化后数字的最小单位是 $1/2^F$。其中 clamp(·)函数用于将量化值限制在$[(-2^I+1),(2^I-1)]$范围内，Int(·)函数用于将数据四舍五入为整数特征图和权重数据经过量化之后，卷积计算公式量化版如公式(4)所示：

$$Conv_{nq}=B_{nq}+\sum_{1}^{R}\sum_{1}^{C}\sum_{1}^{F}X_{nq}\otimes K_{nq} \tag{4}$$

式中，B_{nq} 和 X_{nq} 分别代表量化后的偏置参数和输入数据，K_{nq} 和 $Conv_{nq}$ 分别代表量化后的权重参数和卷积层计算结果。使用对称量化硬件友好优化技术，可以减少网络中的运算量以及降低网络模型的资源需求。

2　FPGA 设计与实现

2.1　解调系统架构

高速光纤光栅混叠光谱信号解调系统结构示意图如图 2 所示。系统主要包括光源、环形器、FBG 传感器、CCD 光纤传感解调模块、FPGA 计算平台以及数据接收与显示模块。本文主要介绍 FPGA 计算平台，它不仅可以实现与加速 CNN 解调算法，也可实现输入光谱数据和特征图数据的缓存。

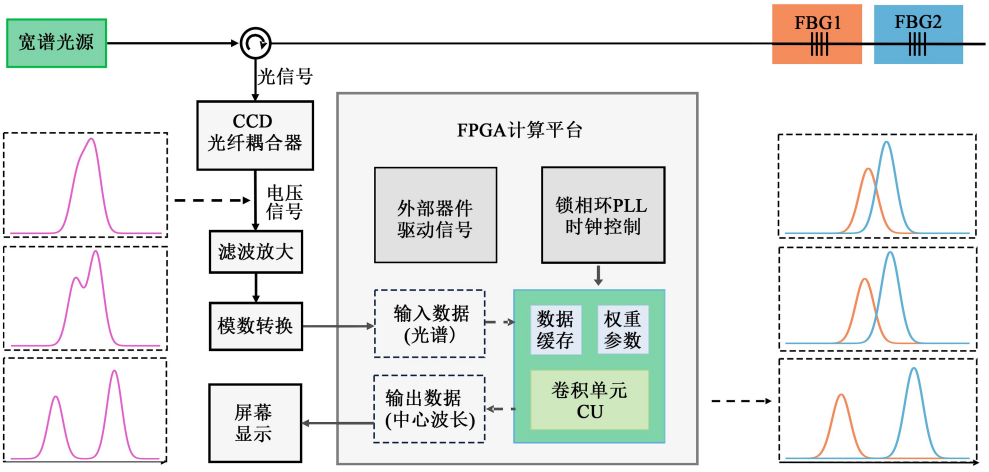


图 2　混叠光谱信号解调系统整体结构图

图 3 展示了混叠光谱信号解调系统的工作流程，首先宽谱光源发出中心波长为 1 550 nm 的光，经过环形器进入一组串联FBG传感器，FBG反射回的光经过 CCD 光纤传感解调模块输出模拟电压信号，再进行模数转换(AD 转换)输出数字信号，使用 FPGA 对其进行混叠光谱信号解调计算和数据传输，将计算出的中心波长数据在屏幕上显示。

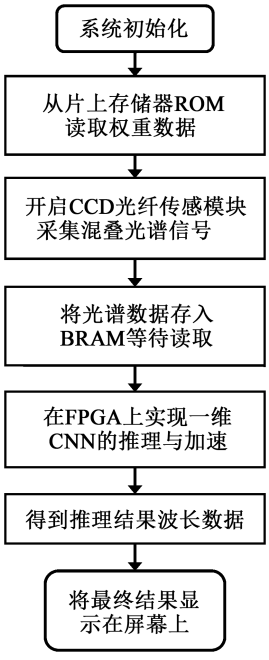


图 3　混叠光谱信号解调计算流程

2.2　硬件电路结构

在目前的研究中，深度学习算法在光谱信号解调方面取得了很好的效果，为了解决边缘设备算力有限的问题，需要设计基于 FPGA 的 CNN 硬件加速电路。在现代数字电路设计中，一般采用自顶而下(Top-down)的设计方法，对 CNN 进行模块化设计的具体过程为：从算法功能和原理入手，参考一维卷积神经网络结构示意图，将整个系统划分为卷积、池化和全连接子系统，其中卷积是 CNN 的重要部分，它包括输入特征图数据和卷积核权重的乘法累加(MAC)运算。每个子系统完成相应的功能，然后进行仿真设计，最终完成整体系统设计。

本文提出的轻量化一维 CNN 算法经过压缩之后大大减少了特征图数据和权重的存储资源需求，可以使用片上存储器 BRAM 来存储全部数据。因

此，本文提出的轻量化 CNN 加速器由两个层次组成：1）片上缓冲区；2）寄存器文件和卷积计算单元 Convolution Unit（CU）。其中的基本流程是从 BRAM 获取光谱和权重数据，然后将它们送入寄存器和 CU。计算完成后结果被传回数据缓冲区，作为下一层输入。

2.3 硬件电路优化

为了有效实现并优化卷积循环，同时考虑硬件资源和推理时间的权衡（Trade-off），需要优化各级结构的并行计算方案[11]。该模型各层之间采用全并行方案，即上一层的计算结果在写入缓存后，下一层立刻读取该结果进行计算。对于其中单个卷积层的循环优化，如图 4 所示，使用多个参数来描述每个卷积层的特征图数据和卷积核维度，循环展开操作中的硬件设计变量将决定计算并行度和硬件占用空间。

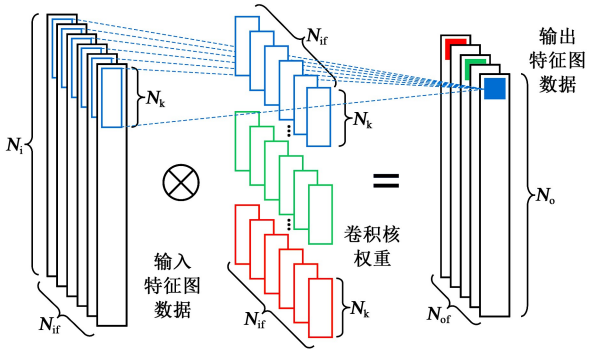


图 4 卷积运算及其参数

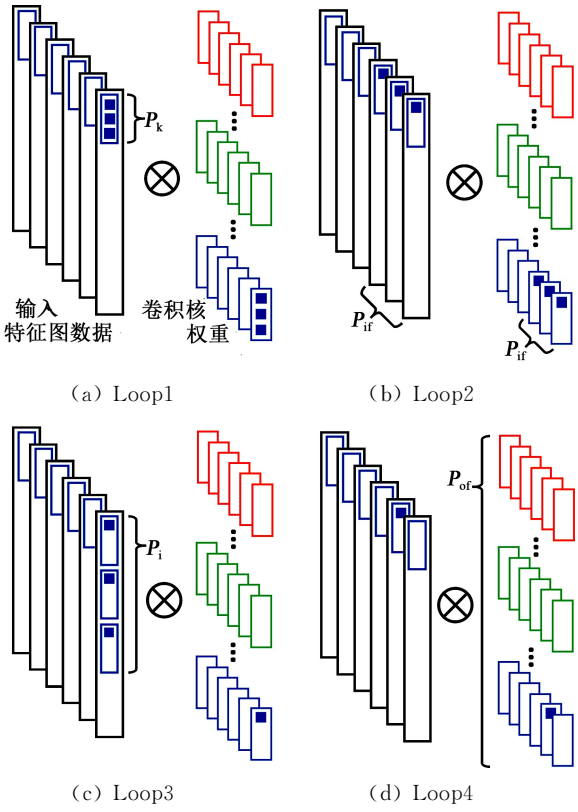
卷积内核窗口的尺寸为 N_k；输入和输出特征图的尺寸分别是 N_i 和 N_o；输入和输出特征图的数量分别是 N_{if} 和 N_{of}（即输入和输出通道数）；循环展开设计中的变量是 P_k，P_{if}，P_i 和 P_{of}，它们表示在不同特征图或卷积核中的并行计算数量。本文中使用的所有参数如表 1 所示。

表 1 卷积循环优化参数和变量设计

参数	卷积窗口	输入特征图	输出特征图	输入通道数	输出通道数
卷积循环	Loop1	Loop3	Loop3	Loop2	Loop4
卷积维度	N_k	N_i	N_o	N_{if}	N_{of}
循环展开	P_k	P_i	P_i	P_{if}	P_{of}

2.3.1 循环展开

如图 5 所示，循环展开决定了网络模型中各级结构的计算并行度，展开不同的卷积循环会影响 CU 架构设计、数据重用方案和片上存储器的读写控制。



（a）Loop1 （b）Loop2

（c）Loop3 （d）Loop4

图 5 循环展开

Loop1（图 5（a））：计算个数为 P_k 的像素和权重的内积，它们来自同一特征图的同一卷积窗口中，但是处于窗口的不同位置。使用输入尺寸为 P_k 的加法器树，对 P_k 个并行乘法结果求和，并将加法器树输出与之前的结果累加。

Loop2（图 5（b））：计算个数为 P_{if} 的像素和权重的内积，它们来自不同输入通道、卷积核，但是处于卷积窗口同一位置。需要将乘积输入加法器树，其中加法器树扇入尺寸与 Loop1 不同。

Loop3（图 5（c））：计算个数为 P_i 的像素和权重的内积，它们来自同一特征图中的不同卷积窗口。由于卷积窗口的输出互相独立，因此不需要加法器树，仅使用累加器来串行累加不同计算周期的乘法器输出。

Loop4（图 5（d））：计算个数为 P_{of} 的像素和权重的内积，权重数据来自不同卷积核的同一像素位置。计算结构与使用 P_{of} 乘法器和累加器，与没有加法器树的 Loop3 相同。

四个卷积循环展开的变量值共同决定了并行 MAC 运算的总数和所需乘法器的数量 P_m：

$$P_m = P_k \times P_{if} \times P_i \times P_{of} \tag{5}$$

2.3.2　数组重排

根据 Loop3 循环展开，在一个卷积窗口中同时进行多次卷积运算，需要并行输入（P_i+N_k-1）个特征图数据，如图 6 所示。

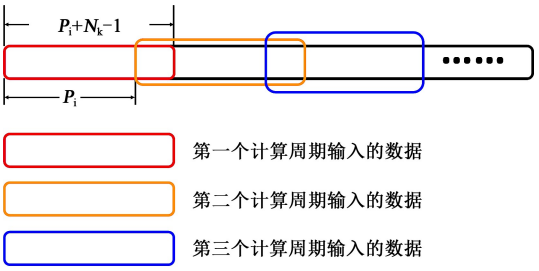


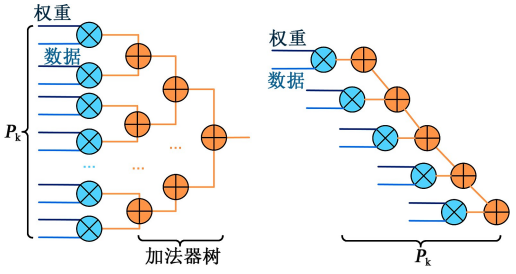
图 6　一维卷积神经网路卷积窗口滑动步骤

图像中红色框的部分是第一个计算周期输入的特征图数据，可以看出，相邻两次的输入特征图数据中会有重叠。但是在解调系统中，光谱数据经过 AD 采集后，将会按顺序串行输出到片上存储器，如果直接将（P_i+N_k-1）个数据输入 CU 进行计算，如图所示，会影响下一周期的数据完整性。为了解决数据重叠问题，需要开辟一块新的内存，将输入数据延迟一个计算周期之后写入内存中，对这两部分数据进行数组分割和重排，再读取处理后的数据传入 CU 进行计算。

2.4　硬件实现

2.4.1　卷积单元架构

卷积运算中的 MAC 单元主要使用 FPGA 底层资源 DSP48E1 实现。经过 2.2 节所述的硬件电路模块化设计和 Loop1 循环展开优化，同时考虑硬件资源消耗，提出了一个卷积计算单元架构，如图 7 所示。



（a）常见卷积单元架构　（b）本文提出的卷积单元架构

图 7　卷积计算单元架构

在目前研究的常见架构中通常采用全并行方案，如图 5(a)所示，卷积窗口尺寸为 P_k，需要同时输入卷积核的 P_k 个数据和特征图数据卷积窗口中的 P_k 个数据，使用组合逻辑，一个卷积单元需要使用（P_k+1）个 DSP 单元；使用时序逻辑，并行乘法器仍需要 P_k 个 DSP 单元。但是 FPGA 中的 DSP 资源数量有限，同时 DSP 单元通常在芯片中占据较大面积，使用过量 DSP 会增加芯片成本。

在本文提出的 CU 架构中，如图 5(b)所示，每个时钟周期的计算量近似相等，并且每个卷积单元只需要 1 个 DSP 单元。这种设计策略可以避免逻辑电路产生时序问题，还能最大化利用 DSP 资源，简化设计和调试过程，能够更好地满足性能、成本和资源利用的要求。

2.4.2　数据的缓存与传输

数据访问带宽是并行计算的瓶颈，特征图中间数据和权重参数都存储在 FPGA 片上存储资源中，通常使用块 RAM(Block RAM)来实现。如果只调用一个 RAM IP 核来存储所有的权重参数，FPGA 中 BRAM 的最小单位是 18 Kb，这种存储方案能够最大可能避免 BRAM 资源的浪费。但是 RAM 读写端口数量有限，不同卷积层无法从单个 RAM 同时读取权重数据，会发生数据访问冲突，无法进行并行计算。数组分割优化将每层的权重数据分别存储在多个 RAM 中，这种优化提高了数据读写效率和系统吞吐量，也更有利于硬件代码的编写和后期人员的维护。

由于卷积层 $Conv_2$ 输入输出通道数较多，卷积计算流程复杂，所以在 $Conv_1$ 中不使用循环展开，串行输出不同通道中的特征图数据。因此在每次的运算中，只调用 $Conv_2$ 卷积核中与该层输入通道对应维度的数据，所以本文重新设计了 $Conv_2$ 的数据存储和传输方案。如图 8 所示。

3　实验结果与分析

3.1　实验设置

在软件方面，该实验将一维 CNN 模型压缩简化，在高性能计算机上完成光谱信号解调算法的训练和验证过程，训练平台选择 NVIDIA GeForce RTX 3060 Laptop GPU，深度学习框架选择 PyTorch，完成训练后采集并记录验证集的预测结果。然后将该模型在测试集数据上进行测试，验证模型的泛化能力。之后对权重文件中的模型参数进行 16 bit 定点数量化，同时在量化后的模型中对测试集数据再次进行测试并记录实验结果。

在硬件方面，用于光谱信号解调系统的基于 FPGA 的 CNN 硬件加速器使用 Xilinx 公司的

XC7Z020CLG400-2 实验平台实现，即 Xilinx Zynq-7000 SoC FPGA 系列芯片，上位机所用的集成设计环境是 Xilinx 推出的 Vivado 2018.3 版本，使用 Verilog 硬件描述语言将模型部署在实验平台。

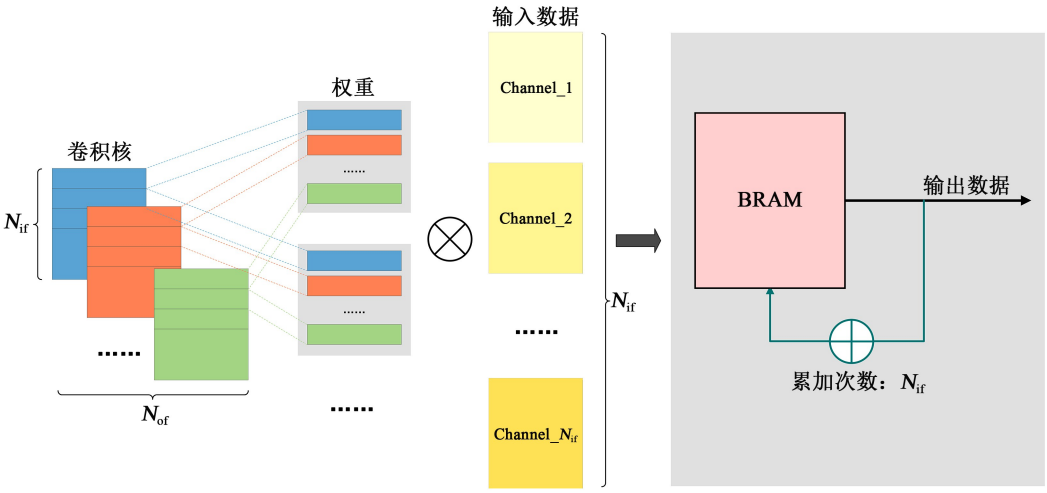


图 8 特征图数据和权重参数的缓存与传输

3.2 模型训练结果

根据上一小节所述的训练步骤，为评估光纤光栅传感器混叠光谱测量信号的误差指标，使用均方根误差(RMS)作为损失函数(Loss Function)，均方根误差损失函数的值越小，表示预测模型描述的样本数据的精确度越好。RMS 表示为：

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

式中，n 为样本数量，y_i 表示当前解调结果，$\bar{y}$ 表示标签数据真实值。为了减少深度学习中的随机性误差，需要进行多次实验并记录实验结果，如图 9 所示。

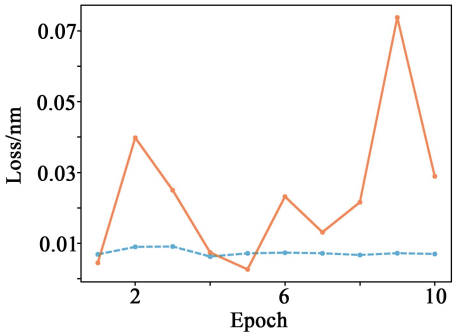


图 9 多次实验结果记录

可以看出，在第 5 次训练中得到了最优结果，绘制该次训练过程中训练误差和验证误差历史数据，并记录验证误差的最小值，图中横坐标为训练次数(Epoch)，纵坐标为误差(Loss)，如图 10 所示。

固定该次训练的随机种子数 42，进行多次实验验证，最小验证误差保持在0.00263nm，所以选用

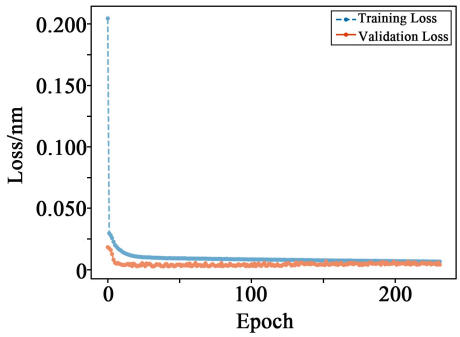


图 10 训练结果

该次训练得到的权重数据进行量化操作。由于 PyTorch 深度学习框架只能识别浮点数数据，所以需要对量化后的参数进行解量化(将定点数转回浮点数)，经过解量化的浮点数数据精度与定点数相同，所以它仍能表现模型的量化后性能。并且量化后的参数与 FPGA 中存储的参数能够一一对应，因此在解调精度方面，上位机中量化模型的推理精度与 FPGA 完全相等。为了消除不同混叠程度的光谱数据对精度的影响，在整体实验数据中均匀抽取 5 000 个数据放入测试集，量化前后的模型推理结果分别如图 11 和图 12 所示。

可以看出，量化前后的模型在测试集数据中都表现良好，模型误差均在 5 pm 以下，主要集中在 1～2 pm 附近。并且量化前后的平均误差分别是 1.189 5 和 1.193 2 pm，两者差距较小，原因是训练集和测试集的数据分布比较均匀，经过压缩后的模型复杂度较低，并且 16 bit 的量化范围较大，有足够的精度来表示模型中的权重参数，所以低精度量化

对性能的影响较小。根据以上测试结果，说明本文提出的轻量化一维 CNN 光谱信号解调算法在经过量化处理之后仍具有较高的解调精度。

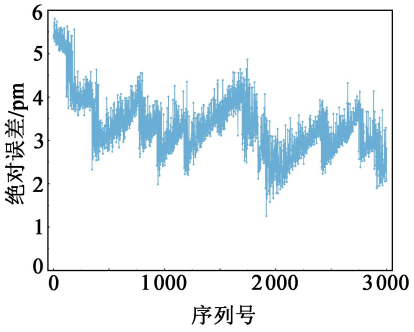


图 11　模型量化前的测试误差

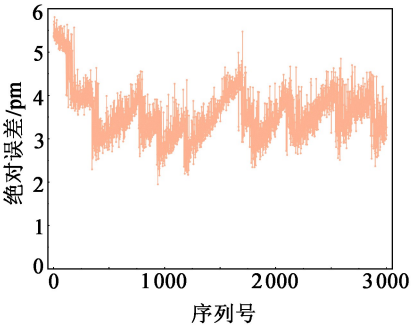


图 12　模型量化后的测试误差

3.3　资源消耗

为了评估轻量化 CNN 模型在 FPGA 中的推理时间和硬件资源消耗，需要在 Vivado 2018.3 中对硬件代码进行仿真、综合、验证和调试，生成比特流文件后便可得到该模型所消耗的资源数量和占比，如表 2 所示，其中包括 Block RAM(BRAM)、DSP 模块、触发器(FF)和查找表(LUT)，它们都属于 FPGA 中的底层硬件资源。

表 2　FPGA 硬件资源使用情况

资源	使用数量	拥有数量	占比/%
LUT	36 087	53 200	67.83
FF	9 830	106 400	9.24
BRAM	62	140	44.29
DSP	220	220	100

为了进一步减少资源消耗，除了需要压缩模型以外，还需要优化神经网络中的激活函数和池化操作。常见的激活函数有双曲正切函数(Tanh)、Sigmoid 函数和 ReLU 函数等，其中 ReLU 函数不仅可以缓解梯度消失问题，而且计算效率高、结构简单，适合硬件实现。常见的池化操作有最大值池化和均值池化，其中均值池化的求平均值操作可以使用两种硬件实现方法，一是使用乘法器，即求和之后乘以 0.5；二是求和之后进行右移一位操作。在 FPGA 中使用时序逻辑实现池化操作时，这三种池化方法所用的硬件资源如表 3 所示。

表 3　时序逻辑：池化层硬件资源使用情况

资源	种类		
	最大值池化	均值池化/乘法器	均值池化/移位
LUT	31	17	17
FF	16	4	15

使用组合逻辑实现时，这三种方法的硬件资源占用如表 4 所示。

表 4　组合逻辑：池化层硬件资源使用情况

资源	种类		
	最大值池化	均值池化/乘法器	均值池化/移位
LUT	31	17	17

可以看出，使用组合逻辑的均值池化所用的资源最少，但是组合逻辑可能会导致延迟冲突问题，因为过多的组合逻辑会增加信号传播延迟，导致时序不满足，从而限制 FPGA 系统的最大工作频率。

3.4　推理时间

首先在 GPU 上实现模型推理过程，量化前进行推理每帧数据需要 0.30 ms，量化后进行推理每帧数据需要 0.28 ms，可以看出，量化后的模型略微提高了推理速度。

对于在 FPGA 上实现并加速的轻量化一维 CNN 模型，如果不使用任何优化技巧，根据 2.3 节中的计算引擎设计和该实验的具体参数，每一次卷积计算需要 P_k 个时钟周期，第一层卷积需要进行 $252\times6\times P_k$ 次卷积计算，在第二层卷积中要进行 $120\times24\times P_k$ 次卷积计算，即使不统计其他计算操作所占用的时间，仅仅卷积计算一个步骤就会使用 $18\,792\times P_k$ 个时钟周期的时间，是优化后时间的 62 倍，所以要使用各种硬件加速技巧。

正如上一节所说，使用组合逻辑可能会导致时序冲突问腿，但可以节省大量资源，更适合于计算密集型的 CNN 模型，并且适当降低时钟频率就可以解决时序紧张问题，因为即使在 10 MHz 的时钟频率下，FPGA 上每帧数据推理的时间也只需要 0.15 ms，相较于 GPU 降低了 46.43%。

4　结论

本文提出了一种轻量级一维 CNN 模型来实现

混叠光谱信号解调。该模型仅包含 34.22 Kb 参数，可以在资源有限的边缘终端设备上实现。同时设计了基于 FPGA 的一维 CNN 加速器来加速推理任务，在 ZYNQ7020 SOC FPGA 平台上设计并实现了一维 CNN 电路，并采用循环展开、数组分割和重排方法进行优化。优化后的电路解调精度为 1.19 pm，速度为每帧 14.96 μs，功耗为 0.301 W。与 CPU 和 GPU 相比，FPGA 在功耗和便携性方面具有较大优势。综上所述，本文提出的基于 FPGA 的解调系统在混叠光谱中心波长解调方面实现了精度和速度之间的平衡，在光纤传感信号解调等领域具有广阔应用前景。

参考文献：

[1] Zhang Y, Zhao Y, Zhou T, et al. Applications and developments of on-chip biochemical sensors based on optofluidic photonic crystal cavities[J]. Lab Chip, 2018, 18(1): 57-74.

[2] Zheng Y, Wu Z, Shum P P, et al. Sensing and lasing applications of whispering gallery mode micro-resonators[J]. Opto-Electron Adv., 2018, 1(9): 180015.

[3] Yu X, Dong X, Chen X, et al. Large-scale multilongitudinal mode fiber laser sensor array with wavelength/frequency division multi-plexing[J]. J. Lightwave Technol., 2017, 35(11): 2299-2305.

[4] Keiser G. Optical Fiber Communications[M]. New York: McGraw-Hill, 2000.

[5] Joe H E, Yun H, Jo S H, et al. A review on optical fiber sensors for environmental monitoring[J]. Int. J. Precis. Eng. Manuf. Green Technol., 2018, 5: 173-191.

[6] 王文娟，薛景锋，张梦杰. 光纤传感在飞机结构健康监测中的应用进展和展望[J]. 航空科学技术，2020，31(7)：95101.
Wang Wenjuan, Xue Jingfeng, Zhang Mengjie. Application progress and prospect of optical fiber sensor in aircraft structural health monitoring[J]. Aeronautical Science & Technol., 2020, 31(7):95-101.

[7] Jiang H, Zeng Q, Chen J, et al. Wavelength detection of model-sharing fiber Bragg grating sensor networks using long short-term memory neural network[J]. Opt. Express, 2019, 27(15): 20583-20596.

[8] Li B, Tan Z W, Shum P P, et al. Dilated convolutional neural networks for fiber Bragg grating signal demodulation[J]. Opt. Express, 2021, 29(5): 7110-7123.

[9] Jiang H, Wang C, Zhao Y, et al. High-precision demodulation method for the cobweb FBG sensor network[J]. Appl. Opt., 2023, 62(2): 419-428.

[10] Luo Y, Cai X, Qi J, et al. FPGA-accelerated CNN for real-time plant disease identification[J]. Comput. and Electron. Agric., 2023, 207: 107715.

[11] Ma Y, Cao Y, Vrudhula S, et al. Optimizing loop operation and dataflow in FPGA acceleration of deep convolutional neural networks[C]// Proc. of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2017: 45-54.

作者简介：

任嘉楠(1999—)，男，安徽省亳州市人，硕士研究生，主要研究方向为 FPGA 硬件设计和光电数字信号处理；

辛璟焘(1980—)，男，博士，副教授，硕士生导师，主要从事光纤传感、光纤激光器、新型激光器件等方面的研究。