基于FPGA和卷积神经网络量化的语音分类加速器

温 冬，姜晶菲，窦 勇，许金伟，肖 滔，韩 哲

国防科技大学计算机学院 长沙 410073

摘要：传统的卷积神经网络在执行语音分类任务时存在模型存储规模大、浮点数值格式算力需求高的问题，限制了算法在低功耗和高吞吐率场景下的应用。通过对网络参数进行二值化处理、网络激活值进行定点数量化，降低了神经网络的存储体积并将浮点数运算转化为较快的定点数运算。基于该量化优化，进行软硬件协同设计，并利用FPGA平台开发出全流水、低功耗和高吞吐率的语音分类加速器，与先进多核CPU平台相比，单PE加速器在吞吐率上取得了18-300倍的加速比。

关键词：可重构计算; 神经网络; 量化压缩; 语音分类

中图分类号：TP391.9 文献标志码：A

An Speech Classification Accelerator Based on FPGA and Convolution Neural Network Quantization

WEN Dong, JIANG Jing-fei, DOU Yong, XU Jin-wei, XIAO Tao, HAN Zhe

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

**Abstract**: Tradition convolution neural networks have defect of huge parameter storage space and demand for extensive computing power due to floating data type when handling speech classification tasks, thus limiting it’s use under low-power and high-throughput circumstance. Turning parameter into binary format and turning feature data into fix-point format can reduce neural network’s storage space and replace floating computing with faster fix-point computing. Based on this optimization and software-hardware-cooperating work, we design a full-pipeline, low-power and high-throughput speech classification accelerator on FPGA platform, which has 18-300x throughput accelerating ratio with single PE compared to state-of-art multi-core CPU platform.

**Keywords**: programmable computing, neural network, quantization, speech classification

1. 绪论

自从Deng、Yu等人[1]将前馈神经网络和LSTM（长短期记忆）声学模型引入语音问题领域以来，LSTM在语音识别和分类领域取得了一系列性能上的突破。然而，基于LSTM的深度神经网络结构复杂，其大量循环时序计算内容难以训练和并行化，限制了LSTM模型在实时场景下的应用。此外，CNN（卷积神经网络）在语音对象上具有同样优异的性能[2]。许多模型利用MFCC（梅尔频率倒谱系数）或其它滤波算法从音频中提取音频特征，进而把音频帧和音频文件转化为特征图[3]，然后像计算机视觉任务中的卷积神经网络一样，设置卷积层、池化层、批量归一化层和全连接层，让卷积神经网络模型运行在音频特征图上。通过利用3x3这样的小尺寸卷积核，语音卷积神经网络相比长短期记忆深度神经网络可以更快地训练和推理，并且拥有更多的并行加速设计空间。考虑到卷积神经网络在各种硬件平台上都有较多的设计方案与探索工作，因此我们在特定的硬件上设计一个声学卷积神经网络加速器是可行的。

然而，在嵌入式和边缘计算场景中部署卷积神经网络算法模型时存在两个重要的障碍: 功耗和速度。传统硬件平台在卷积神经网络的计算任务下表现不佳: 虽然嵌入式MCU（微控制单元）可以在低功耗模式下运行，但由于传统标量处理器的低计算并行度和嵌入式MCU较低的工作频率，传统嵌入式设备无法高效地处理卷积运算。另一方面，GPU和先进的CPU可以更快地运行卷积神经网络，但它们高达数百瓦的功耗对于嵌入式设备部署环境来说是不可接受的。为了保证计算精度，深度神经网络中通常在运算和存储时通常采用浮点数据格式，但浮点格式在存储和计算这两个方面上都对MCU、CPU和GPU平台很不友好。浮点数据需要更大的存储字长和更多的计算功能部件，这就导致了更大的存储空间开销和更大的电路设计面积，进而增加了硬件功耗。同时，浮点数据格式的计算复杂度决定了设计者很难通过硬件设计或算法改进减少计算周期数。

目前，一些工作[4]已经证明卷积神经网络在进行推理时并不一定需要浮点数据格式，低精度计算也可以让算法获得与全精度浮点数据相似的性能。这些工作为硬件设计人员提供了一些新的思路: 通过量化，研究人员在精度损失很小的情况下可以将权重和激活值数据转化为半精度浮点格式、8位浮点格式、定点格式甚至二值化格式[5]。基于各种卷积神经网络的量化方法，学术界和工业界出现了BNN（二值化神经网络）加速器和支持8bit浮点数据的GPU等。这些设计与传统的处理器平台相比，在大大降低了功耗的同时，也具有高达数百倍的加速比。基于卷积神经网络的声学算法和对硬件友好的卷积神经网络模型优化方法使我们能够设计一个实时语音分类应用加速器。

与GPU和CPU平台相比，ASIC（专用集成电路）和FPGA（现场可编程逻辑门阵列）高度定制化的设计特性使得开发者可以更灵活和更针对性地利用量化的卷积神经网络模型。ASIC和FPGA通过设置与量化模型相同的数据位宽与数据格式、设计不同层次的流水线、扩展并行度，进而实现更好的计算性能和更低的运行功耗。在运行卷积神经网络模型时，ASIC可以在计算性能和功耗上都取得很好的效果，但昂贵的设计和制造成本限制了它的广泛应用。而FPGA由于其可配置的特点和成熟的工业设计，可以在性能、功耗和成本之间保持较好的平衡，因此很适合作为卷积神经网络的加速平台。

由于FPGA灵活度高、易部署和功耗低的特点，各云主机厂商和数据中心都积极上线FPGA云或利用FPGA加速特定的云服务：微软在利用FPGA服务器加速必应搜索引擎时取得了优异的加速效果，亚马逊也在加速布局FPGA与人工智能服务的组合。在前人工智能时代，FPGA被广泛用于数据中心的网络通信加速、云服务器操作系统虚拟化和加速生物科学计算等，而在人工智能时代，功耗低面积小的FPGA不仅可以为边缘端带来更多算力，其配置灵活的特性可使用户在云端实现人工智能算法进行定制化加速、高吞吐率负载和低延迟优化，为传统高性能计算平台带来新的设计思路。

本文对一种基于卷积神经网络的语音分类算法进行了加速，网络权值为+1或-1，原始的激活数据为全精度浮点数据格式[6]。我们的加速器设计在Xilinx XCKU115上实现，与目前先进的CPU平台相比，不仅实现了18x-300x的加速比，并且实现了较低的运行功耗。 本文的主要工作如下:

我们利用Matlab内置的量化函数和设计空间探索方法将一个语音分类卷积神经网络的浮点型激活值数据逐层地转化为定点数据，FPGA平台上的定点计算性能弥补了定点数据的精度损失。

我们设计了一个基于FPGA的多PE BWN（权值二值化神经网络）加速器，该加速器具有共享权值存储设计、均衡流水线结构和卷积神经网络层间低延迟流水线设计。   
 我们在CPU平台上进行了单线程、多线程和多节点环境下的目标语音分类模型的性能测试，得到了多组高性能CPU平台上的基准性能数据。与上述测试结果相比，我们的设计在性能功耗比和计算加速比上具有绝对优势。

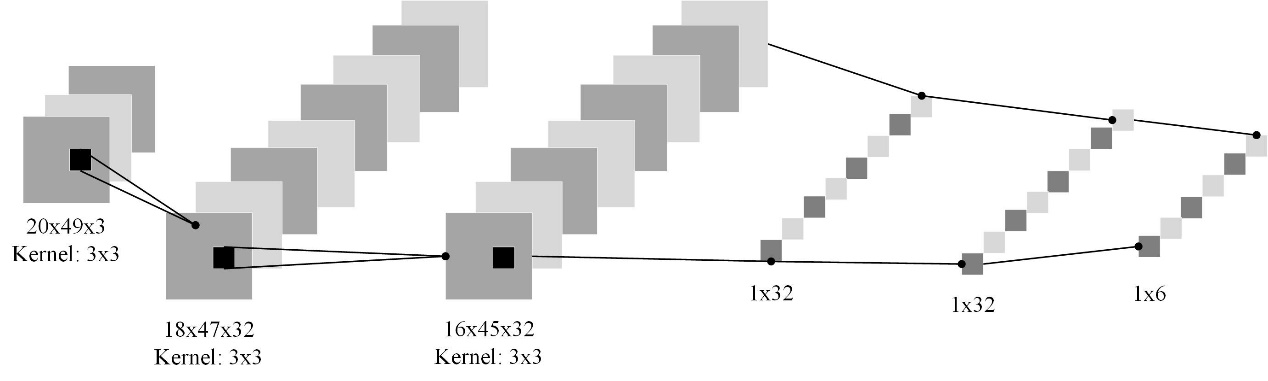
1. 神经网络推理量化

现在的深度神经网络模型通常包含大量的参数，为了更好地训练神经网络模型，研究人员通常选择全精度浮点数据格式作为参数数据格式。然而，在推理任务中，所有这些参数一旦训练完成在数值上就不会被改变，因此就不需要继续保留全精度数据格式或保留所有的参数。通过离线的对神经网络参数进行剪枝、量化压缩精度，我们可以显著的降低网络的存储体积、总计算量和计算延迟。[7] 提出了一种将浮点深度神经网络参数压缩为包含+1和-1值的二进制数据的算法。与原有的64位浮点数据格式模型相比，这种压缩方法不仅大大减少了参数存储空间需求，而且用加减运算替代卷积中的乘法运算，显著降低了单个卷积运算的计算延迟。该方法使设计人员可以将所有参数都放在芯片内，降低了内存数据交换开销；另外对于FPGA硬件平台而言，加减电路也比乘法器更易设计和实现。

另外，[7]中提出了将神经网络运行中的激活值数据也全部转化为二值格式的方法。与训练完成后不发生改变的神经网络参数不同的是，网络激活值数据会随着输入对象（如图片或音频特征图）的不同而产生数值上的波动，对激活值数据做二值化处理就会产生较大的精度损失。此时对网络激活值进行定点化处理就可以在精度和计算性能上得到较好的平衡：一方面定点数计算的计算周期数较浮点数大幅减少；另一方面定点格式可以根据激活值数据实际的数值分布范围来调整整数和小数位宽，小数位宽越大，对原数据的精度保持就更好，整数部分的位宽则决定了数据格式的数值表征范围。相比二值化数据格式，定点数据可以更好的在计算精度和计算速度上达到平衡。

1. 语音分类模型
   1. 模型结构和权值二值化

该卷积神经网络语音分类模型是在Tensorflow语音指令数据集上进行训练得到，能够分辨出“up”，“down”，“yes”，“right”，“left” 共5种单词语音段和未知单词语音(unknown)。最初得到的原始网络模型中的权值和中间结果（激活值）都由浮点格式表达。首先，该模型利用MFCC算法将一个音频文件变换成维度为20\*49\*1的浮点格式的张量，该张量即为音频的特征图矩阵。此张量将被送入包含两个卷积层、三个全连接层的卷积神经网络中，所有的卷积核大小均为3，卷积步长为1。该卷积神经网络没有边缘扩展，便于我们加速，该网络的结构如图1所示。最后，该模型通过softmax函数输出六种类型标签的预测概率。



**Figure 1 Convolution Neural Network’s Architecture**

**图1 卷积神经网络结构**

当模型训练完成、参数数值都确定之后，我们利用tanh函数将数值分布没有约束范围的浮点权值约束到（-1，1）的范围，然后使用一系列放缩与映射的方法将浮点权值离散为0或1，最后离散为-1或+1。 图2详细展示了我们处理权值数据的流程。经过测试，此时的模型（权值为二进制-1、+1，激活值为浮点数）的整体精度不低于85%。



**Figure 2 The Process of Parameter’s Binarization**

**图2 权值二值化流程**

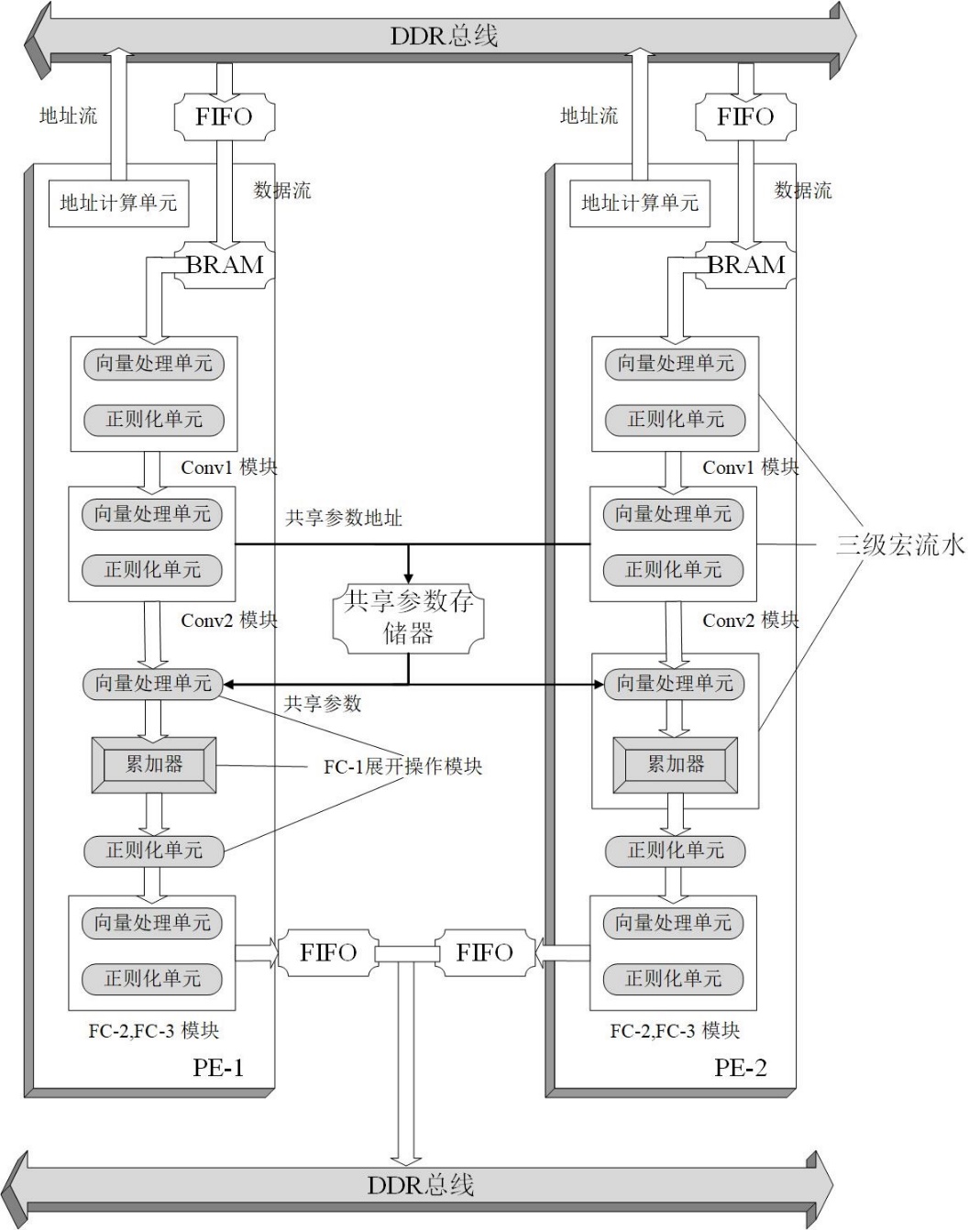
* 1. 激活值数据量化

神经网络中每一层计算的数据可分为两部分：乘累加运算后的中间结果、中间结果进行批量归一化后的最终输出数据。通常来说，中间结果的绝对值和方差可能很大，但经过批量归一化处理后，方差和数据分布范围都将缩小，最终输出数据的分布情况会得到改善。[8]论证了批量归一化对于模型总体精度的重要影响，而批归一化操作中的乘除法不仅依赖先验的平均值和方差参数，而且对于数值精度较为敏感，因此为了兼顾硬件性能与整体精度，我们需要对同一层内的中间结果和输出结果采取不同的量化数据位宽以尽可能保证批归一化的计算精度。

为了将激活数据转化为定点格式并使精度损失尽可能小，我们首先确定每个层内中间结果和输出结果的数值分布范围（即数据整数部分位宽），然后利用Matlab量化函数库和设计空间搜索法确定每个层中两种数据的最佳小数位宽量化格式组合。所以，我们提出了一种基于权值二值化、激活数据定点化的卷积神经网络语音分类模型。在本文的实验部分将讨论具体的数据量化结果。

1. 加速器体系结构

由于目标加速网络体积较小、层数较浅且权值都是二值化的，所以我们主要设计重点是在片上设置共享的二值化参数存储器、逐层加速神经网络以及在层间设计均衡流水线提高加速器的整体性能。



**Figure 3 Hardware Architecture**

**图3 硬件结构图**

一个加速器可以包含一个或多个PE、一个DDR总线和一个DRAM存储器。 每一个PE都有自己的功能模块，包括五层神经网络处理单元、地址计算模块、输入输出与数据存储。其中卷积层1模块、卷积层2模块和全连接层1模块按照层间流水线的方式工作。

* 1. 参数存储

与一些加速较大规模神经网络的加速器需要用DRAM辅助片上存储器来存储参数不同，我们的模型参数规模适中，在采用了二值化处理后参数存储空间进一步减少。 表1显示了该模型参数的详细信息。

**Table 1 Parameter Information**

**表1 参数信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 层数 | 滤波器数量 | 卷积核大小 | 参数量 | 参数存储大小 |
| conv1 | 32 | 3\*3\*1 | 288 | 36B |
| conv2 | 32 | 3\*3\*32 | 9216 | 1152B |
| fc1 | 32 | 16\*45\*32 | 737280 | 92160B |
| fc2 | — | — | 1024 | 128B |
| fc3 | — | — | 192 | 24 |

全连接层1占据了绝大部分的网络参数，其他层的参数数据规模相当小，可以直接存储在片上。考虑到全连接层1参数的巨大规模，一个很自然的想法是在多个PE之间共享它们。我们设置所有PE为同步工作，并在全连接层1的计算时提取完全相同的预训练参数。该共享存储模块由32个BRAM块组成，每个BRAM块用于存储全连接层1中的一个滤波器参数。

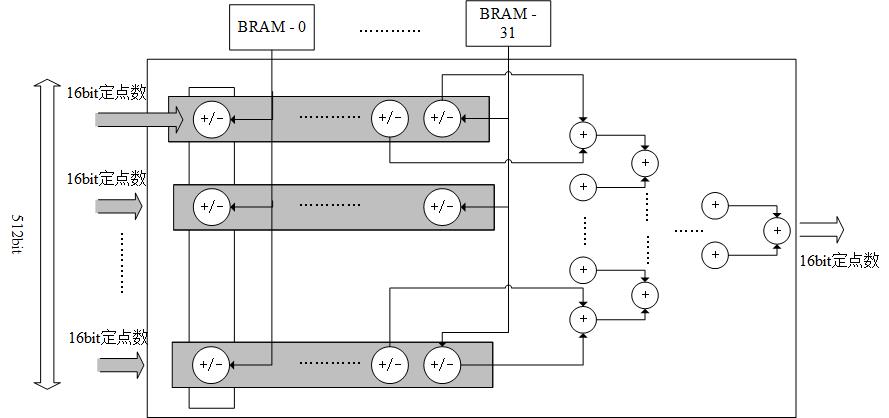


**Figure 4 FC-1’s Sharing-Parameter Design**

**图4 全连接层1的参数共享结构**

PE向参数共享模块发送目标参数地址，该地址是BRAM的行地址。BRAM中的参数沿第三维存储，其存储方式与全连接层1的输入数据组织方式一致。 通过广播，共享参数数据被发送到所有PE。

对于除了全连接层-1之外的二值化网络参数，我们将其直接存储于片内，每个BRAM块对应存储一个滤波器中的所有参数。每次运算时，BRAM块都将对应像素位置的参数直接交付向量处理单元并与输入激活值进行向量运算。向量运算器首先根据传入的1bit参数（代表+1，-1）对激活值进行原样输出或是取反操作得到32个参数运算结果，再利用加法树对32个激活值对应的参数运算结果进行逐层累加，最终得到该次运算的结果。



**Figure 5 Vector Processing Unit**

**图5 向量处理单元**

* 1. 位宽扩展

在一个卷积层或全连通层内，输入的定点激活数据往往是经过了批归一化的，因此通常服从正态分布、数值分布范围较小，而经过乘累加计算后，数据的方差可能会很大且不规则。批量归一化依赖于这些中间结果的均值和方差以改善乘累加后的数据分布，这对最终的精度至关重要。实验结果证明，如果在归一化步骤中方差和均值数据使用与输入激活值相同的量化数据格式，将会造成一定的数值精度损失并给模型的分类准确率带来不必要的损失。

为了解决这一问题，我们引入位宽扩展以改善计算精度：对于批归一化操作中的平均值、方差超参数，我们增加额外的小数位宽以增加此类参数在硬件上的表示精度；对于精度需求相对不高、小数位宽相对较少的计算中间结果，我们在批归一化操作时对其做简单的小数位宽扩充，使其能够和平均值、方差参数对齐。当DSP（数字信号处理器）输出归一化运算结果后，加速器再对多出的小数位宽进行截断，使数据恢复到原始的输入激活值量化格式。

同样的，乘累加运算的中间结果也有类似的位宽扩展需求。累加器对多组乘法运算的乘积进行累加的过程中可能会出现数值溢出的情况，因此乘累加的结果需要更多的整数位宽而不是小数位宽。我们的加速器对归一化步骤和乘累加结果累加器进行了扩展，实验结果表明该方法较好的保证了计算精度。

* 1. 层间流水线

实现层间流水线的关键是要平衡流水线每一层间的运行周期数。在VGG-16和AlexNet等深度卷积神经网络中通常很难保持这种平衡，因为随着网络的深入，更深的层需要前面的层以更快的速度生成激活值结果，这一设计要求大大超出了当前硬件平台的计算能力。

本文的目标加速网络层数较少、深度较浅，所以相比VGG和Alex网络更容易在两个卷积层之间保持平衡。我们改变了卷积层1的取数据地址生成方法，使之与卷积层2的计算模式相匹配。同时，为了能在运算周期数上实现两个卷积层的均衡，我们扩展了卷积层1功能模块的并行度，使加速器可以在一个流水线宏节拍内生成一组可被卷积层2完整使用的输入数据。



**Figure 6 Balancing Pipeline Between Conv-1 and Conv-2**

**图6 平衡卷积层1和卷积层2之间的层流水线**

为了在卷积层2中输出一组数据，网络模型需要卷积层1在输入音频特征图一个的5\*5区域中计算9个3\*3滑动窗口。为了平衡计算周期数，我们扩展了卷积层1的计算阵列规模，使其在一个宏流水线周期内产生9个32\*1向量，在下一个周期内这些向量将被发送到卷积层2，并为全连接层1产生一个32\*1向量，同时卷积层2的计算功能也将在一个宏周期内运行完毕。全连接层1是该加速器的瓶颈，由于计算量巨大，要保持该层与两个卷积层的平衡所需要的计算资源是当前平台难以承受的，但图4中的设计可以保证全连接层1在每个宏周期内保持一个向量进行累加。

通过对层流水线的调整，目标神经网络可以在不需要将层间结果写回DRAM中的情况下进行加速，从而降低存储开销，即从输入音频特征到最终预测结果的数据流始终保持在层次流水线中，加速器只在取数和最后回写时与DRAM通信。在流水线的各个层中，我们将所有的计算分解为向量计算单元、归一化单元等功能部件，提高了硬件的运行频率。

1. 实验结果及分析
   1. 量化模型性能

我们利用Matlab-2018a的量化函数将激活值数据和批量归一化参数以饱和溢出方式转换为定点数据。为了找到最佳的位宽设置方法，我们进行了多组量化实验，并比较了它们的精度性能。

**Figure 7 Accuracy Experiment Result**

**图7 精度实验结果**

考虑到数据格式设计要与硬件平台相契合，我们限定所有的数据格式位宽在32bit~16bit之间，并以此为限定条件对中间结果与归一化参数的位宽进行设计空间探索。在神经网络的量化算法中，激活值（中间结果）往往对数值精度有较高的鲁棒性，因此在寻找最佳方案时不需对中间结果的小数位宽做过多搜索。此外，考虑到归一化参数的精度需求比中间结果更高，因此在探索位宽组合时，一般设置归一化小数位宽大于等于中间结果小数位宽，但我们仍然在探索中做了与此设想相反的实验，实验结果验证了设想的正确性。

图7中的实验结果表明，当激活值数据采用8位小数位宽，归一化参数采用9位小数位宽时，该模型的性能最佳。需要注意的是硬件不能像Matlab代码那样简单地处理除法运算，因此我们将归一化函数中方差的除法转化为倒数乘法，并扩展小数位宽以保证精度。

我们在8700K平台上对量化版本的神经网络程序进行分段计时。我们使用matlab单线程和多线程并行库运行程序，在不考虑MFCC预处理段的情况下，卷积层2是CPU平台上的主要性能瓶颈，对程序性能有着决定性的影响，而参数的量化操作则几乎对性能没有影响。

**Table 2. Running Performance on CPU Platform**

**表2. CPU平台上的运行计时**

|  |  |  |
| --- | --- | --- |
| 8700K单线程 | | |
| 功能段 | 时间(Seconds) | 时间占比 |
| 数据加载与预处理 | 9.413601 | 7.40% |
| 卷积层1量化 | 0.000508 | <1% |
| 卷积层1 | 3.084229 | 2.43% |
| 卷积层2量化 | 0.000694 | <1% |
| 卷积层2 | 114.582 | 90.14% |
| 全连接层1量化 | 0.006863 | <1% |
| 全连接层1 | 0.023377 | <1% |
| 全连接层2量化 | 0.000834 | <1% |
| 全连接层2 | 0.001659 | <1% |
| 全连接层3量化 | 0.000945 | <1% |
| 全连接层3 | 0.001589 | <1% |
| 总计 | 127.1163 | 100% |
| 8700K 多线程并行 | | |
| 功能段 | 时间(Seconds) | 时间占比 |
| 数据加载与预处理 | 9.804942 | 19.74% |
| 卷积层1量化 | 0.000195 | <1% |
| 卷积层1 | 1.803836 | 3.63% |
| 卷积层2量化 | 0.000307 | <1% |
| 卷积层2 | 38.02353 | 76.56% |
| 全连接层1量化 | 0.006184 | <1% |
| 全连接层1 | 0.025257 | <1% |
| 全连接层2量化 | 0.000588 | <1% |
| 全连接层2 | 0.001316 | <1% |
| 全连接层3量化 | 0 | 0 |
| 全连接层3 | 0.000981 | <1% |
| 总计 | 49.66721 | 100% |

* 1. 硬件加速性能

我们在8700K和Matlab-2018a单线程平台、8700K和Matlab-2018a并行库平台和Intel Xeon 5220(2.2GHz,18核)多节点MPI环境下（使用Matlab-2018a分布式并行库）运行整个数据集共1512个音频段。考虑到我们的加速器只负责加速卷积神经网络部分的运算，不涉及程序预处理以及音频帧转化为音频特征图的部分，因此在实验中我们只测试神经网络前向传播所消耗的时间，并不计入程序预处理与MFCC算法的部分。我们以上述实验环境的运行时间为基准来计算我们设计的加速比。表4展示了我们的测试结果，测试结果表明与当前先进的CPU平台相比，我们的加速器可以取得18-300倍的加速比和更低的功耗。

**Figure 8 Performance Compared with Multiple Platforms**

**图8 多平台性能对比**

我们在Xilinx KU-115 FPGA和Xilinx Vivado 2018.3平台上实现了我们的单PE版本加速器设计，并测试了运行时间、运行精度和运行功耗。为了更好的测试上述数据，我们设置了加速器循环执行次数，进行了最长达2小时的压力测试。

**Table 3 Accelerator’s Performance**

**表3 加速器性能参数**

|  |  |
| --- | --- |
| 运行数据全集时间(*S*) | 0.4116 |
| 静态功耗 (*W*) | 9.63 |
| 动态功耗 (*W*) | 10.06 |
| 运行准确率 | 84.96% |
| FPGA平台资源利用率 | 38% |
| 吞吐量(语音段每秒) | 3675.9 |
| 运行频率 (MHz) | 150 |
| 峰值定点运算次数(GOP) | 23.85 |

表3的数据表明卷积层2是最耗费时间的功能段，然而通过使用平衡的逐级流水线，我们的加速器可以消除原有神经网络中的瓶颈，获得优异的加速性能。 同时，流水线内的数据流大大减少了DDR总线的通信量，消除了其他计算平台上内存带宽限制。

1. 结论

我们的加速器是一款针对特定语音分类模型的低功耗、高加速比、高吞吐量的专用硬件平台。我们的加速器在设计中突出了参数共享存储、精度灵活扩展和均衡的神经网络层间流水线的特点，通过软件算法与硬件平台的协同改进和设计使本加速器平台与现有的CPU平台相比具有优异的性能。我们在Xilinx FPGA上实现了设计方案并进行了压力测试，结果表明该加速器是一种可靠且高效的智能计算器件。

参考文献

[1] Geoffrey H, Li D, Dong Y, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition [J]. IEEE Signal Processing Magazine, 2012, 12:82-98.

[2] Tom S, Vaibhava G. Advances in Very Deep Convolutional Neural Networks for LVCSR[C/OL]. arXiv:1604.01792v2[cs.CL]

[3] Pakyurek M, Atmis M, Kulac S, et al. Extraction of Novel Features Based on Histograms of MFCCs Used in Emotion Classification from Generated Original Speech Dataset[J]. Electronics & Electrical Engineering, 2020, 26:46-51.

[4] Tung F, Mori G. Deep Neural Network Compression by In-Parallel Pruning-Quantization[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(3):568-579.

[5] Lu C, Xu W, Jin S, et al. Bit-Level Optimized Neural Network for Multi-Antenna Channel Quantization[J]. IEEE Wireless Communications Letters, 2020, 9(1):87-90.

[6] Bo L, Hai Q, Yu G, et al. EERA-ASR: An Energy-Efficient Reconfigurable Architecture for Automatic Speech Recognition with Hybrid DNN and Approximate Computing[J]. IEEE ACCESS, 2018, 6:52227-52237.

[7] Matthieu C, Itay H, Daniel S, et al. Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1[C/OL]. arXiv:1602.02830v3[cs.LG].

[8] Shibani S, Dimitris T, Andrew I, et al. How Does Batch Normalization Help Optimization[C/OL]. arXiv: 1805.11604v5[stat.ML].