Speech Enhancement Accelerator Based on Recurrent Neural Network*

LANG Biao[1], CHEN Fei[2*], MA Cai[1], SHI Dongyu[1], JI Junyu[3]
(1.Tianjin Key Laboratory of Imaging and Sensing Microelectronic Technology, School of Microelectronics, Tianjin University, Tianjin 300072, China;
2.Research Institute of Tsinghua University in Shenzhen, Shenzhen 518057, China;
3.Shenzhen Eartech Company Limited, Shenzhen 518172, China)

Abstract: Deep learning speech enhancement technology is capable of helping listeners improve verbal recognition rate, but the large size of the neural network makes it difficult to implement in edge devices. Therefore, a recurrent neural network speech enhancement accelerator is put forward which can be utilized in edge devices such as hearing aids. The calculation of each layer of the neural network is realized through hardware level independent matrix multiplication and the hardware level pipelining operation between the layers of multi-layer neural network is also realized, thus reducing the calculation delay through parallelism and pipelining. Experiments indicate that compared with noisy speech, the signal-to-noise ratio of the proposed algorithm is improved by 17.302 dB, 8.412 dB and 4.732 dB respectively in the noise environment of volvo, factory2 and babble, the short-time objective intelligibility is increased by 1.4%, 0.8% and 0.4% respectively, the perceptual evaluation of speech quality is improved by 1.498, 0.504 and 0.234 on average, and these three indicators are higher than the traditional speech enhancement algorithm and neural network algorithm compared. When the clock frequency is at 10 MHz, the processing delay of the accelerator is 9.2 ms, which can meet the real-time requirements of edge-side applications.

Key words: speech enhancement; edge devices; recurrent neural network; accelerator; real-time
EEACC: 7230 **doi**: 10.3969/j.issn.1004-1699.2023.04.009

基于循环神经网络的语音增强加速器设计*

郎 标[1], 陈 霏[2*], 马 偲[1], 石东宇[1], 姬俊宇[3]
(1.天津大学微电子学院, 天津市成像与感知微电子技术重点实验室, 天津 300072;
2.深圳清华大学研究院, 广东 深圳 518057; 3.深圳市智听科技有限公司, 广东 深圳 518172)

摘 要: 使用深度学习的语音增强技术能够提升听者的言语识别率, 但因神经网络的规模较大难以应用于边缘设备中。因此, 提出了一种可用于助听器等边缘设备的循环神经网络语音增强加速器。该加速器将神经网络的计算用独立矩阵乘法硬件实现, 并在多层神经网络的层之间实现硬件级的流水操作, 通过并行和流水降低了计算延时。实验表明, 与带噪语音相比, 在 volvo、factory2、babble 噪声环境下, 所提算法的信噪比分别平均提升了 17.302 dB、8.412 dB 和 4.732 dB; 短时语音可懂度分别平均提高了 1.4%、0.8%和 0.4%; 语音质量感知评估平均提高了 1.498、0.504 和 0.234; 这三项指标均高于所对比的传统语音增强算法与神经网络算法。当时钟频率为 10 MHz 时, 加速器的处理延时为 9.2 ms, 可以满足边缘侧应用的实时性需求。

关键词: 语音增强; 边缘设备; 循环神经网络; 加速器; 实时性
中图分类号: TN492 **文献标识码**: A **文章编号**: 1004-1699(2023)04-0555-10

语音增强[1]是指当纯净语音被噪声污染后, 尽量从受污染的语音中恢复出原始的语音信号, 并使处理后的语音具有更高的可懂度与清晰度的一项技术。这项技术已经被用在可穿戴设备、移动设备等边缘设备中。基于神经网络的语音增强技术的推理运算实质上是矩阵的乘法和加法, 而大量的神经元节点势必会造成非常大的功耗和内存消耗, 因此, 如何在不同噪声环境、资源受限的情况下, 同时兼顾高精度和低延时来构建基于循环神经网络(Recurrent Neural Network, RNN)的语音增强加速器是非常有挑战性的。

传统的语音增强算法, 主要是利用无话段的语音活动检测(Voice Activity Detection, VAD)来估计噪声模型, 进而从带噪语音中去除掉噪声, 优点是不

项目来源: 国家重点研发计划项目(2016YFA0202201); 深圳市科技计划项目(JSGG20191129141019090, JCYJ20210324115610028, JSGG20210713091808027)
收稿日期: 2022-04-20 **修改日期**: 2022-06-09

依赖先验语音信息,所需计算量较少,对硬件实现比较友好,在实际应用中比较受欢迎。比如谱减法[2-3]、维纳滤波法[4]、子空间法[5-6]等算法。这些方法对平稳噪声的抑制效果比较显著,对于非平稳噪声,其往往不能得到很好的降噪效果[7]。因此,学者们提出了基于神经网络的语音增强算法,并取得了不错的效果,尤其是单通道语音增强算法[8-11]。

由于神经网络强大的非线性建模能力,深度学习被广泛地运用在了语音增强领域。事实上,对于边缘侧应用来说,除了高精度和高能效以外,低延时也是必需的要求之一。近年来,一些实时的语音增强算法被提出[12-14]。2018 年,Valin[12]提出了一种基于深度学习的实时全频带语音增强方法。其使用具有四个隐藏层的深层循环神经网络来估计理想临界频带增益,取得了不错的效果。同样在 2018 年,Parviainen 等人[13]提出一种基于低延迟神经网络的语音增强系统,并证明了其可行性。2019 年,Bhat 等人[14]提出了基于多目标学习卷积神经网络的语音增强技术,以提高助听器用户感知的整体语音质量,并在智能手机端实现了实时语音增强技术。然而,这些语音增强算法的规模与参数量较大,未在实际的硬件中实现,没有考虑到便携式边缘计算硬件设备的资源约束。

近年来,已训练好的 DNN 嵌入到智能的边缘计算设备越来越受到关注。边缘计算的一个优点是计算在最终用户设备上本地完成,以减少延迟并保护隐私[15]。近些年提出了许多关于 RNN 神经网络的加速器[16-22]。2016 年,Nurvitadhi 等人[16]对比了门控循环单元(Gated Recurrent Unit,GRU)分别在中央处理器(Central Processing Unit,CPU)、图形处理器(Graphics Processing Unit,GPU)、现场可编程门阵列(Field Programmable Gate Array,FPGA)以及专用集成电路(Application Specific Integrated Circuit,ASIC)中的加速实现,结果表明使用 FPGA 的性能功耗比要优于 CPU 与 GPU。2018 年,Wang 等人[17]提出使用基于 FPGA 的 RNN 加速器,使用结构压缩技术降低了一定的存储内存使用,虽然性能指标不错,但是其工作在 200 MHz 系统时钟下的功耗为 22 W,并不适合运用在低功耗的边缘设备中。2020 年,Zhu 等人[18]提出了基于网络压缩的高性能 RNN 加速器,通过网络修剪和量化的压缩方法,可以降低计算冗余、内存开销和能耗。然而,这些加速器研究的应用范围较宽,不能很好地适用某一种应用类型,没有一个专用于边缘设备的 RNN 语音增强加速器,尤其是在 FPGA 的嵌入式低延迟、高能效平台上。因此本文提出的创新性理论成果和关键技术,可以为边缘计算应用的语音增强加速器的研究与设计提供一定的研究基础,具体如下:

首先,利用梅尔滤波器组进行时域上的频段划分,避免了时频域变换的硬件实现消耗;

其次,使用各个频段的归一化对数能量谱作为输入特征,并利用跨频段信息与网络跳越连接的方式进行网络的搭建,取得了不错的效果;

最后,在硬件层面上使用时分复用与流水线技术来加速 RNN,并设计了一款小型的 FPGA 加速器。

本文的其余部分组织如下。第一节进行算法描述,并简述其实现流程。第二节介绍硬件的设计与实验结果分析。最后一节对全文进行了总结。

1 RNN 语音增强算法描述

本文提出了一种基于 RNN 的单通道语音增强加速器设计与实现,语音的采样频率为 16 kHz。本文所采用的加速器系统处理流程如图 1 所示。

接收到带噪语音后,首先经过分频带模块,在时域上进行 16 个频段的划分,并将相邻频段合并以得到 4 个跨通道信息,将结果存在随机存取存储器(Random Access Memory,RAM)中,等待语音增益的计算;其次,利用通道结果计算每个通道每帧的对数能量,并经过归一化后作为 RNN 的输入;经过 RNN 处理得到 16 个通道的语音增益,再与 RAM 中的数据进行相乘即可得到增强后的通道语音;最后通过语音合成得到增强语音。

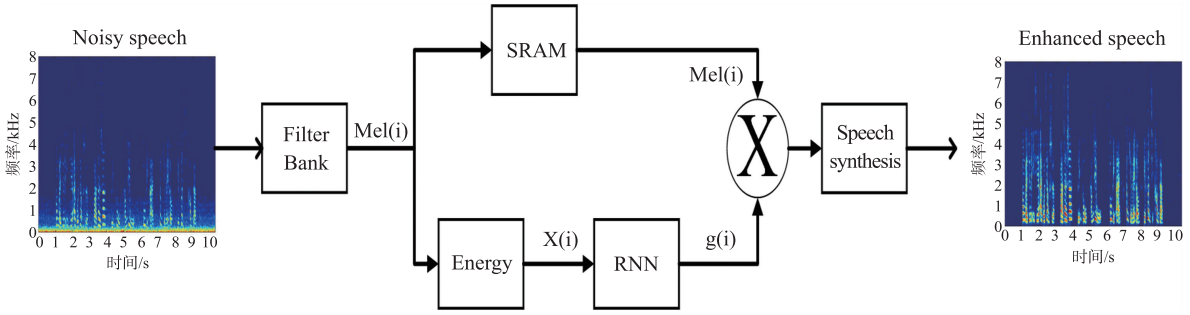


图 1 单通道语音增强算法的加速器系统结构图

1.1　通道划分与特征计算

语音增强性能的好坏与所使用的语音特征有密不可分的关系。如果采用语音的频域特征,需要计算快速傅立叶变换(Fast Fourier Transform,FFT),语音合成前还需做逆傅里叶变换 IFFT,FFT、IFFT 包括神经网络的计算都是复杂的向量计算,对于硬件实现来说是很大的消耗。因此,尝试用时域 FIR 分析滤波器组代替傅里叶时频域 FFT 转换以降低特征提取复杂度。由于部分边缘设备的应用场景需要单独对一些频段进行增强,如听损患者部分频段听不清需要单独对该频段给予一定的增益,使用文献[23]中的矩形滤波器组,虽然不同频带之间的耦合较少,但是很容易在对部分频段补偿增益后造成频带间语音信号的陡变,因此本文对文献[23]提出的矩形滤波器组进行修改,用梅尔滤波器组代替原先的矩形滤波器组。梅尔滤波器组有着频带间 50%的交叠,可以有效地防止增强语音频带间的陡变,具体步骤如下:

①首先按照梅尔尺度进行频段的划分。梅尔频率尺度更加符合人耳对音高的感知,定义为:

$$M=1\ 125\ln\left(1+\frac{f}{700}\right) \tag{1}$$

式中:M 表示以梅尔(Mel)为单位的感知频率,f 表示以 Hz 为单位的实际频率。音频采样率为 16 kHz,根据奈奎斯特采样定律,式中 f 的范围为 0~8 000 Hz,因此可以得出 M 的范围 0~2 835 mel。在梅尔域内对整个音高范围进行均分,并通过式(1)即可得到各个频段的中心频率。

首先计算频率向量:

$$\Delta f=\frac{fs}{N} \tag{2}$$

$$\boldsymbol{F}(k)=(k-1)\Delta f \quad 1\leqslant k\leqslant\frac{N}{2} \tag{3}$$

式中:f_s 为采样率,N 为滤波器阶数,Δf 为频率分辨率,$\boldsymbol{F}$ 为频率向量,即实际频率点。由于梅尔滤波器的图形是左右对称的三角形,因此式中的 k 只取到 $N/2$。

②计算每个频段的滤波器频率向量 $\boldsymbol{H}_i$,其中 $f(m)$ 为中心频率:

$$\boldsymbol{H}_i=[\alpha_{i,1},\alpha_{i,2},\cdots,\alpha_{i,k},\cdots,\alpha_{i,N/2}]$$

$$\alpha_{i,k}=\begin{cases}\dfrac{F(k)-f(m-1)}{f(m)-f(m-1)} & f(m-1)\leqslant F(k)\leqslant f(m)\\ \dfrac{f(m+1)-F(k)}{f(m+1)-f(m)} & f(m)<F(k)\leqslant f(m+1)\\ 0 & F(k)<f(m-1)\\ 0 & F(k)>f(m+1)\end{cases} \tag{4}$$

对响应向量 $\boldsymbol{H}_i$ 进行中心对称延拓,即:

$$\boldsymbol{H}_i=[\alpha_{i,1},\alpha_{i,2},\cdots,\alpha_{i,k},\cdots,\alpha_{i,N/2},\alpha_{i,N/2-1},\cdots,\alpha_{i,2}] \tag{5}$$

③对频率向量 $\boldsymbol{H}_i$ 进行 IFFT 后乘双窗(汉明窗和矩形窗卷积归一化),即可得到时域滤波器系数 h_i。

④得到滤波器系数后,将其与对应的输入序列点相乘求和即可得到 16 个频段的值,如下:

$$x_i(t)=\sum_{k=1}^{N}h_i(k)\times[x(t-N+k)+x(t+N-k)] \tag{6}$$

本文使用的滤波器采用时域对称的结构,因此计算量可以缩减一半,便于硬件实现。由上式可知,当计算当前时刻的滤波器输出时,会使用到该时刻之后的信息,因此滤波器会产生一个 N/f_s 的延时,称为群延时。考虑到边缘设备的实时性应用需求,因此滤波器的延时应尽量小以满足整体系统的实时性需求,将该滤波器的阶数 N 设置为 128,即群延时为 8 ms,其幅频响应如图 2 所示。

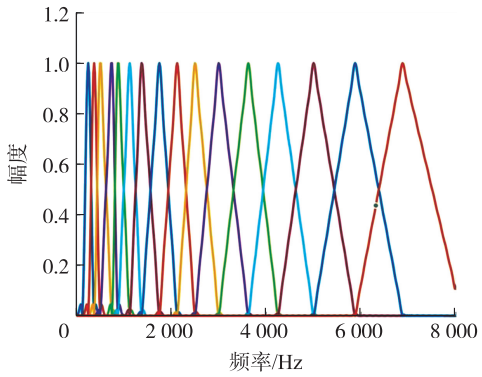


图 2　滤波器幅频响应

通道划分完成之后,按下式计算神经网络的输入特征:

$$E(i)=\sum_{k=1}^{F_r}x_i^2(k) \tag{7}$$

$$E(i)=\frac{\log_2E(i)+8}{16} \tag{8}$$

式中:F_r 表示一帧的长度,这里取 10 ms。$E(i)$ 表示第 i 个通道的一帧语音的能量,$x_i(k)$ 表示语音信号第 i 个通道的第 k 个频点。由于语音信号的采样位数为 16 bit,因此其能量的最小值为 2^{-24}~2^8,因此这里加 8 除以 16 可以把语音信号归一化到-1~1 之间。

将 16 个通道的相邻频段合并为 4 个通道,利用这 4 个跨通道信息和另外 16 个通道的对数能量作为神经网络的输入特征。此外,利用纯净语音与带噪语音的关系,我们将网络的输出特征定义为:

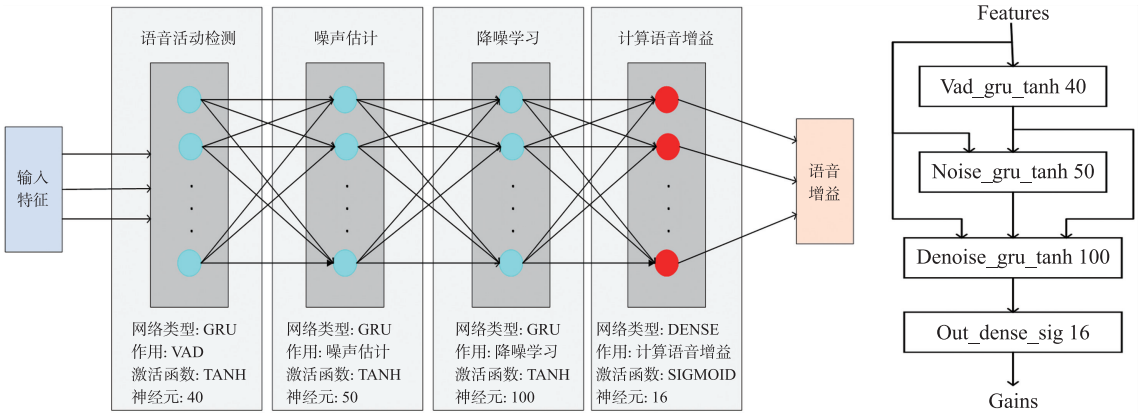
$$g(i)=\sqrt{\frac{E_s(i)}{E_x(i)}} \tag{9}$$

式中:$E_s(i)$表示频段 i 的纯净语音的能量,$E_x(i)$表示频段 i 的带噪语音的能量。

1.2 网络结构

在语音处理中,语音的信息是上下文相关的序列,其输出是受输入影响的。因此,使用带有记忆能力的 RNN 神经网络对语音的序列化信息进行建模。传统的 RNN[24]存在梯度爆炸和梯度消失问题,并不能真正地处理好长距离的依赖。长短时记忆网络(Long Short-Term Memory,LSTM)和门控循环单元(Gated Recurrent Unit,GRU)可以很好地处理长距离依赖问题。为了节省硬件资源消耗,本算法使用计算复杂度更低、计算效率更高的 GRU 来构建网络架构[25]。

图 3 是本算法使用的网络结构示意图,网络的输入特征是 20 个通道归一化后的对数能量,输出是 16 个通道的语音增益。图 3(a)代表 RNN 的网络模型结构,网络结构由 GRU 层和全连接层(Dense)共四层组成,它们的作用分别是语音活动检测(Voice Activity Detection,VAD)、噪声估计、降噪学习以及语音增益的计算。图中指示了所使用的激活函数类型、神经元节点以及其网络类型。图 3(b)显示了具体的网络连接细节,本算法使用了跳越连接(Skip Connection),跳越连接表明每层网络的输入可以来自不同层的输出,这确保了特征的可重用性,且收敛速度比传统网络更快[26]。



(a) RNN的网络模型结构 (b) RNN的网络连接

图 3 RNN 结构示意图

1.3 语音合成

最后,进行波形重建。将 RNN 获得的子带增益与通道噪声语音相乘,得到经过降噪调整后的每个通道的信号,将每个时域点的 16 个通道信息相加来重构波形,如下式:

$$s=\sum_{i=1}^{16} g(i)\times \mathrm{mel}(i,1:F_r) \tag{10}$$

式中:i 为频段的索引,F_r 为帧长,mel 为各个通道的值。

1.4 实验数据与评价指标

本算法使用的纯净语音数据选自清华大学中文语料库 THCHS-30[27],其中的文本选取自大容量的新闻,基本包含了所有的发音语句,语音采样率为 16 kHz,采样位数为 16 bit。使用的噪声来自 NOISEX92 库[28],该噪声库中包含 16 种不同环境的噪声,采样率为 19.98 kHz,因此在使用之前需将采样率重采样到 16 kHz,将纯净语音与噪声按照不同的信噪比进行混合,生成带噪语音。本算法使用生活中常见的带噪语音,如车内噪声 volvo、生产车间噪声 factory2、以及人群噪声 babble 来测试语音增强的效果。

本文使用的语音评价指标为信噪比(Signal-to-Noise Ratio,SNR)、短时客观可懂度[29](Short-Time Objective Intelligibility,STOI)和语音质量感知评估[30](Perceptual Evaluation of Speech Quality,PESQ)。其中,SNR 为信号与噪声的功率比,其值越大越好;STOI 的典型值在 0~1 之间,可以认为是听懂语音的百分比;PESQ 与人的主观听觉感受具有很强的一致性,其典型值在-0.5~4.5 之间,分值越大代表语音质量越好。

2 硬件设计与实验结果分析

2.1 电路设计与实现

本文提出的神经网络算法需要满足边缘侧应用小尺寸、实时性的需求。因此选择面积为 10 mm×10 mm 的 Xilinx Artix7 系列的 XC7A50T-2CPG236I 芯片,该芯片采用高性能、低功耗的 28 nm 工艺,与上一代产品相比降低了 50%的功耗。

根据所采用的 FPGA 芯片的数据手册进行 PCB(Printed Circuit Board)的原理图和封装设计,图 4 所示为加速器 PCB 测试电路图。该 PCB 采用了四层板的设计方式,主要模块为数字麦克风、有源晶振

以及 FPGA 芯片。首先根据需要编写各个模块的 VHDL 代码,仿真通过后通过 JTAG 编程口即可将二进制文件下载进加速器电路中,进而通过扬声器即可听到麦克风采集的声音。

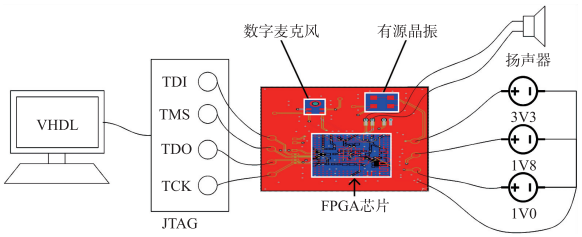


图 4　加速器 PCB 测试电路图

神经网络的运算本质上就是矩阵求和的过程,每一个输入与对应的权重相乘并求总和,再与偏置相加的结果经过激活函数输出。神经网络逻辑结构到电路结构的映射方式主要有两种:空间展开和时分复用。空间展开的方式将每一个神经元都映射为一个独立的包含乘法器和加法器的线性运算单元(Processing Element,PE)。优点在于其并行计算程度高,更适合用来加速某些对性能要求和吞吐率要求很高的特定应用。相比之下,时分复用的方式将网络中的多个神经元映射为同一个运算单元电路,该处理单元在时间轴上逐层串行计算出层内每个节点的输出,再进行下一层的节点计算,这种方式需要的运算资源和芯片面积少,更适合计算资源受限的边缘侧语音信号处理的应用。因此本算法采用时分复用的设计方式。PE 的结构如图 5 所示。

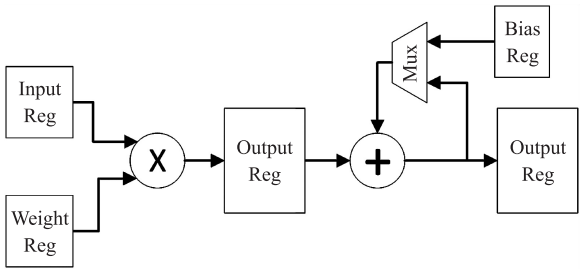


图 5　PE 的结构

GRU 层的计算占据非常大的资源,因此,进一步优化 GRU 的计算是非常有必要的。如图 6 所示,将 GRU 的运算分成了三步,使用步进控制单元即可控制网络计算。图中展示了计算单元流水、层间流水的特性,计算单个节点时,在乘法和加法后各加一级寄存器,经过若干周期后,计算加法同时也在计算乘法,形成对权重和输入的流水计算;当前层计算完毕后,将层输出存入层间寄存器中,然后继续从输入寄存器中获取输入计算层输出,经过若干个周期后就可以实现对输入连续不断的流水计算,以流水线的形式降低了多层、多步计算的延时。全连接层类似于 GRU 的单步计算,这里不再赘述。

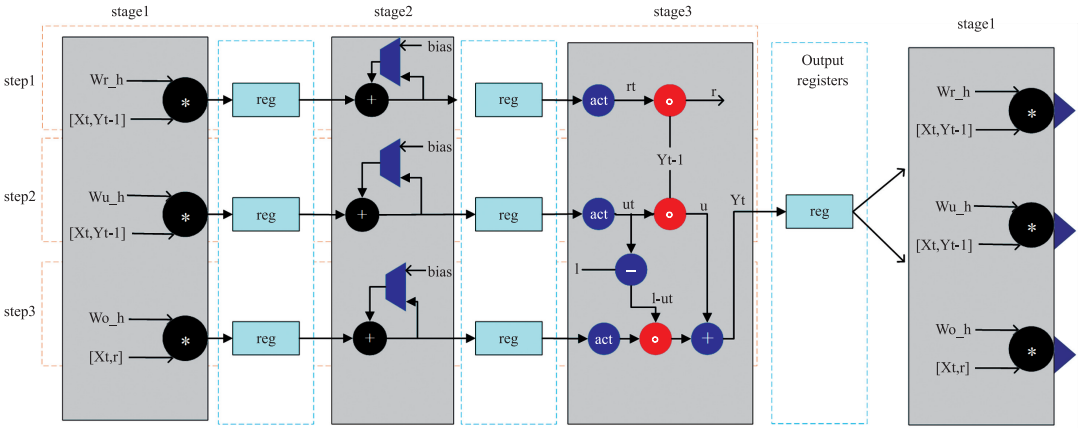


图 6　GRU 流水线

图 7 所示为本文设计的加速器放在助听器壳里的实例,该助听器壳虽然相对其他部分助听设备较大,但是本加速器对于音响等其他尺寸较大的边缘设备而言完全符合需求。此外,该加速器板仍可以通过重新布局设计或通过 ASIC 设计实现的方式进一步减小尺寸大小。

图 7　助听加速器实例图

2.2　软件仿真结果对比

在这部分中,为了验证本文提出算法的效果以及可行性,将其与深度学习算法 Rnnoise[13]、传统语音增强算法多带谱减法[3](Multi-Bands Spectral Subtraction,MBSS)、维纳滤波法[4](wiener_wt)、最优改进对数谱估计算法[31](Optimally Modified Log-Spectral Amplitude,OMLSA)进行对比。

利用 MATLAB 生成不同信噪比下的带噪语音，并利用语音增强算法进行语音降噪，最后通过语音评价指标进行对比。为了体现测试的充分性，从 THCHS-30 库中选取未参与训练的 100 段语音，分别与 NOISEX92 中的 volvo 噪声、factory2 噪声、babble 噪声按照-10、-5、0、5、10 的信噪比进行合成，得到 1 500(100×3×5)段带噪语音，利用这 1 500 段带噪语音进行测试。

首先采用信噪比进行评估。表 1 给出了不同算法语音增强后的 SNR。

表 1 不同算法处理后的平均 SNR 的比较

噪声	混合信噪比/dB	Noisy	Rnnoise	MBSS	wiener_wt	OMLSA	This Work
volvo	-10	-10	9.29	-3.35	-1.93	12.22	11.24
	-5	-5	12.80	1.42	1.83	15.66	14.49
	0	0	15.86	5.42	4.99	18.10	17.77
	5	5	18.10	8.11	8.40	19.34	20.59
	10	10	19.44	9.45	12.33	20.03	22.42
	平均	**0**	**15.098**	**4.21**	**5.124**	**17.070**	**17.302**
factory2	-10	-10	1.48	-0.63	1.46	2.61	2.76
	-5	-5	6.66	3.36	3.56	5.12	5.16
	0	0	10.50	6.80	6.0	8.35	8.00
	5	5	13.47	8.95	8.91	11.97	11.28
	10	10	16.32	9.80	12.27	15.82	14.86
	平均	**0**	**9.686**	**5.656**	**6.440**	**8.774**	**8.412**
babble	-10	-10	-0.49	-8.0	-1.17	-1.57	0.73
	-5	-5	1.76	-2.80	1.86	1.59	1.85
	0	0	5.33	1.96	4.93	4.97	3.70
	5	5	9.47	5.79	8.41	8.97	6.74
	10	10	13.29	8.17	12.24	12.87	10.64
	平均	**0**	**5.872**	**1.024**	**5.254**	**5.366**	**4.732**
整体平均		**0**	**10.219**	**3.63**	**5.606**	**10.403**	**10.149**

可以看到，相对于带噪语音，本算法的信噪比在三种噪声环境下分别提升了 17.302 dB、8.412 dB、4.732 dB。传统的语音增强算法中，OMLSA 取得效果最好，甚至在 factory2 和 babble 噪声环境下，提升幅度都略高于深度学习的算法；而深度学习的算法 Rnnoise，其整体提升与本文算法基本上是一致的。整体而言，本文在 volvo 噪声环境下可以取得最好的信噪比提升，虽然在另外两种噪声环境下的提升不是最优，但与最优的之间仅相差 0.284 dB，基本是可以接受的。

信噪比只能反映整体语音质量的提升，无法直接反映语音的失真度，因此本文对语音增强后语音的可懂度进行比较，采用的比较因子是 STOI，其值越大代表语音的可懂度越好。表 2 给出了不同算法语音增强后 STOI 的平均得分。其中，以带噪语音 Noisy 的 STOI 得分作为参考因子，本算法在 volvo 和 factory2 噪声条件下可以显著提高语音的可懂度，分别提高 1.4%和 0.8%。而在 babble 噪声下，仅提升了 0.4%，这是因为 babble 噪声中的部分人声会被认为是语音而被保留。整体而言，本算法相对于带噪语音可提升 0.9%，具有最高可懂度。

为了进一步比较语音的客观质量，选用与人的听觉感受有一致性的 PESQ 作为比较因子。表 3 是不同算法平均 PESQ 的比较结果。

可以看到，整体而言，本文提出的语音增强模型 PESQ 提升了 0.746，远高于其他算法模型。相比于带噪语音，本算法在三种噪声环境下分别可提高 1.498、0.504、0.234。

总之，传统的语音增强算法适合处理高信噪比下相对平稳的噪声，比如 MBSS 和 OMLSA。而对于低信噪比或者非平稳类型的噪声却不太理想。此外，可以看到，本算法 SNR 的提升虽然略低于 OMLSA 方法，但是 STOI 与 PESQ 的提升却远高于该方法，显然，在牺牲较小 SNR 的同时显著提高 STOI 与 PESQ 对于边缘设备的应用是很有意义的。结合三种评估结果综合来看，本文提出的算法能够很好地处理平稳和非平稳噪声，弥补传统语音增强算法的缺陷。

表 2　不同算法处理后平均 STOI 的比较

噪声	混合信噪比/dB	Noisy	Rnnoise	MBSS	wiener_wt	OMLSA	This Work
volvo	−10	0.87	0.87	0.85	0.74	0.83	0.88
	−5	0.90	0.91	0.89	0.79	0.90	0.91
	0	0.92	0.94	0.93	0.84	0.94	0.94
	5	0.94	0.95	0.95	0.90	0.95	0.96
	10	0.96	0.97	0.96	0.94	0.96	0.97
	平均	**0.918**	**0.928**	**0.916**	**0.842**	**0.916**	**0.932**
factory2	−10	0.60	0.53	0.60	0.52	0.53	0.61
	−5	0.71	0.72	0.71	0.64	0.61	0.72
	0	0.81	0.83	0.80	0.74	0.70	0.82
	5	0.87	0.90	0.87	0.83	0.81	0.88
	10	0.92	0.94	0.92	0.89	0.90	0.92
	平均	**0.782**	**0.784**	**0.78**	**0.724**	**0.71**	**0.79**
babble	−10	0.44	0.32	0.42	0.37	0.34	0.39
	−5	0.55	0.49	0.54	0.51	0.46	0.58
	0	0.66	0.67	0.67	0.66	0.60	0.66
	5	0.77	0.81	0.79	0.77	0.75	0.78
	10	0.84	0.89	0.87	0.86	0.87	0.87
	平均	**0.652**	**0.636**	**0.658**	**0.634**	**0.604**	**0.656**
整体平均		**0.784**	**0.783**	**0.785**	**0.733**	**0.743**	**0.793**

表 3　不同算法处理后平均 PESQ 的比较

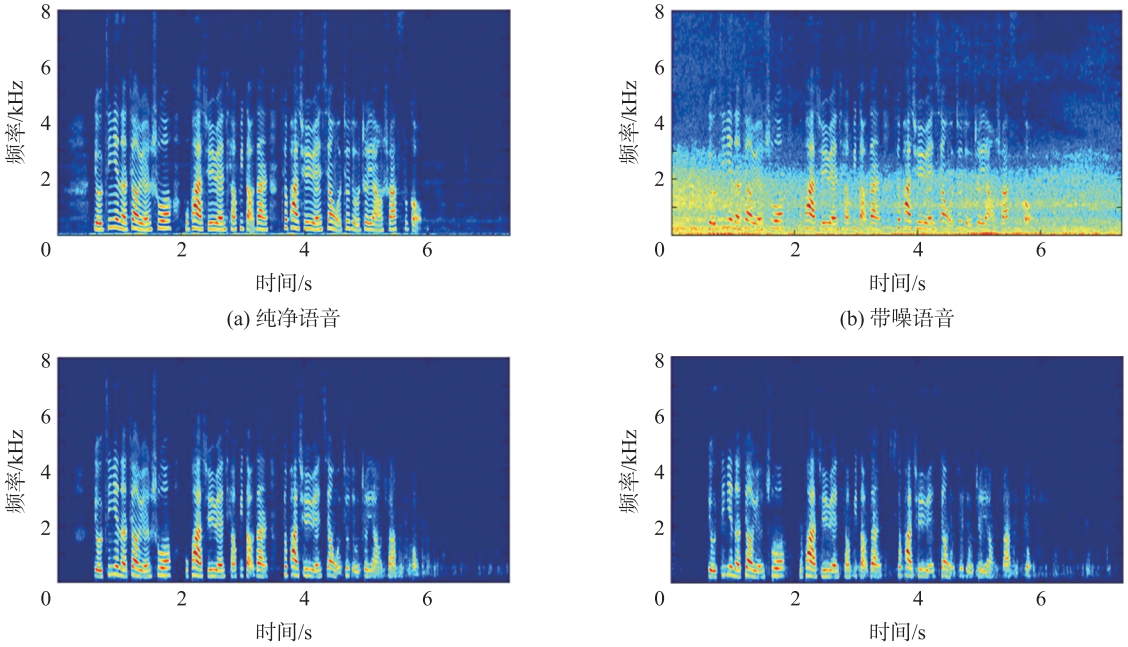
噪声	混合信噪比/dB	Noisy	Rnnoise	MBSS	wiener_wt	OMLSA	This Work
volvo	−10	1.09	1.34	1.41	1.13	1.79	2.28
	−5	1.21	1.61	1.73	1.20	2.20	2.67
	0	1.53	2.02	2.07	1.32	2.64	3.16
	5	2.00	2.57	2.34	1.58	2.97	3.67
	10	2.51	3.09	2.59	1.98	3.19	4.05
	平均	**1.668**	**2.126**	**2.028**	**1.442**	**2.558**	**3.166**
factory2	−10	1.04	1.13	1.05	1.04	1.12	1.21
	−5	1.05	1.18	1.15	1.07	1.21	1.39
	0	1.12	1.37	1.39	1.13	1.40	1.66
	5	1.32	1.69	1.77	1.25	2.68	2.01
	10	1.67	2.17	2.19	1.46	2.13	2.45
	平均	**1.240**	**1.508**	**1.510**	**1.190**	**1.708**	**1.744**
babble	−10	1.06	1.71	1.06	1.05	1.15	1.15
	−5	1.04	1.11	1.06	1.04	1.09	1.13
	0	1.07	1.13	1.13	1.06	1.16	1.26
	5	1.15	1.34	1.30	1.14	1.13	1.50
	10	1.38	1.73	1.61	1.32	1.66	1.83
	平均	**1.140**	**1.404**	**1.232**	**1.122**	**1.238**	**1.374**
整体平均		**1.349**	**1.679**	**1.590**	**1.251**	**1.835**	**2.095**

2.3　加速器的处理效果分析

本文所提出的增强算法通过 Verilog HDL 代码在硬件加速器系统中实现,并利用计算机仿真工具 VCS 对所设计的 RTL 代码进行仿真验证。硬件输出与利用 C 语言仿真输出的结果进行对比,输出相关性大于 0.97,符合设计要求。

本文实际录制了公路的噪声,并与纯净语音进行混合生成 0 dB 带噪语音。图 8 所示为软硬件增强语音的语谱图。其中,图 8(a)为纯净语音的语谱图,可用来对比;图 8(b)为带噪语音的语谱图;图 8(c)为

软件仿真生成的增强语音语谱图;图 8(d)为硬件运行生成的语谱图。通过对比,可以看出,硬件与软件生成的语谱图几乎完全一致,与纯净语音相比几乎消除了所有的噪声。



(a) 纯净语音 (b) 带噪语音 (c) 软件的增强语音 (d) 硬件的增强语音

图 8 软硬件增强语音的语谱图

此外,本文使用软硬件协同设计的方法进行硬件设计,即 PC 端和硬件端同步设计。在 PC 端进行网络训练的同时,进行硬件环境的搭建,最后将训练的权重参数传给硬件端即可完成硬件的设计。表 4 所示为软硬件处理带噪语音后的 PESQ 和互相关系数对比,其中语音 1、语音 2、语音 3 分别对应 volvo、factory2、babble 的噪声环境。可以看到,本文算法基于软件和硬件的 PESQ 得分基本一致,相关系数接近为 1,进一步证明了本文算法的实用性和可靠性。

表 4 软硬件处理带噪语音后的 PESQ 和互相关系数对比

测试样本	PESQ			互相关系数
	Noisy	软件	硬件	
语音 1	1.761 0	3.438	3.398 0	0.999 7
语音 2	1.235 0	1.851	1.792 0	0.999 0
语音 3	1.116 0	1.346	1.324 0	0.995 7
平均	1.370 7	2.2117	2.171 3	0.998 1

本文提出的加速器系统除了 RNN 网络模块外,还有分频带、特征提取以及语音合成模块,但由于这些模块只是简单的乘法和加法运算,因此没有详细叙述。经过对整体系统的测试后,FPGA 经过大约 91 548 个周期后输出结果,时钟频率为 10 MHz 时,总延时约为 9.2 ms,满足实时性的要求。另外本加速器在 FPGA 平台的功耗为 32.61 mW,对于助听器中使用的锌空电池(电压 1.45 V,容量为310 mAh),本加速器可以工作 13 h 左右;而对于音箱等具备较大电池的其他边缘设备,可以使用更长的时间。未来的研究会进一步将 FPGA 中的电路硬件设计用 ASIC 方式实现,可望将硬件加速器功耗再下降一个数量级。

表 5 显示了加速器整个系统在布局布线完成后的资源利用情况。其中查找表(LUT)利用率最高为 62.25%,但是 DSP 和 BRAM 占用率并不高,这意味着该加速器系统仍可以支持更大的网络结构,且可以在边缘侧应用中应用。

表 5 硬件设计消耗的 FPGA 资源

资源名称	使用资源	可用资源	使用率/%
LUT	20 292	32 600	62.25
FF	31 576	65 200	48.43
DSP	30	120	25
BRAM	31	75	41.33

3 总结

本文创新性地提出了一种基于循环神经网络的语音增强加速器,并通过软硬件协同设计的方法证明了该加速器可以应用在助听器等边缘侧应用中。算法上采用在时域上进行频段划分以减少因 FFT 造成的硬件消耗,电路上采用时分复用最小计算单元并利用流水线等技术进行设计。实验结果表明,

本文提出的语音增强加速器可以有效地抑制噪声,提高信噪比、短时客观语音可懂度和语音质量感知评估。软硬件的输出相关性大于 97%,且当时钟频率为 10 MHz 时,系统的总延时为 9.2 ms,功耗为 32.61 mW。该加速器的面积与功耗仍然很大,后续的研究将致力于 ASIC 芯片的研究,以进一步降低芯片面积与功耗,以便适用于小体积的可穿戴边缘侧产品。

参考文献:

[1] Loizou P C.语音增强:理论与实践:Theory and Practice[M]. 第 1 版. 高毅,肖莉,邓方,等译. 成都:电子科技大学出版社,2012:1-5.

[2] Berouti M,Schwartz R,Makhoul J. Enhancement of Speech Corrupted by Acoustic Noise [C]//1979 IEEE International Conference on Acoustics,Speech,and Signal Processing,Washington,DC,USA,1979,4:208-211.

[3] Kamath S,Loizou P. A Multi-Band Spectral Subtraction Method for Enhancing Speech Corrupted by Colored Noise[C]//2002 IEEE International Conference on Acoustics,Speech,and Signal Processing,Orlando,FL,USA,2002:IV-4164.

[4] Yi H,Loizou P C. Speech Enhancement Based on Wavelet Thresholding the Multitaper Spectrum[J]. IEEE Transactions on Speech and Audio Proceessing,2004,12(1):59-67.

[5] Dendrinos M, Bakamidis S, Carayannis G. Speech Enhancement from Noise:A Regenerative Approach[J]. Speech Communication, 1991,10(1):45-57.

[6] Ephraim Y, Van Trees H L. A Signal Subspace Approach for Speech Enhancement[C]//1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, Minneapolis, MN, USA,1993:355-358.

[7] 鲍长春,项扬. 基于深度神经网络的单通道语音增强方法回顾[J]. 信号处理,2019,35(12):1931-1941.

[8] Wang Y,Narayanan A,Wang D L. On Training Targets for Supervised Speech Separation[J]. IEEE/ACM Transactions on Audio, Speech,and Language Processing,2014,22(12):1849-1858.

[9] Xu Y,Du J,Dai L R,et al. A Regression Approach to Speech Enhancement Based on Deep Neural Networks [J]. IEEE/ACM Transactions on Audio,Speech,and Language Processing,2015,23(1):7-19.

[10] Kounovsky T,Malek J. Single Channel Speech Enhancement Using Convolutional Neural Network [C]//2017 IEEE International Workship of Electronics,Control,Measurement,Signals and Their Application to Mechatronics,Donostia,Spain,IEEE,2017:1-5.

[11] Xiang Y, Bao C. A Parallel-Data-Free Speech Enhancement Method Using Multi-Objective Learning Cycle-Consistent Generative Adversarial Network[J]. IEEE/ACM Transactions on Audio,Speech,and Language Processing,2020,28:1826-1838.

[12] Valin J M. A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement [C]//2018 IEEE 20th International Workshop on Multimedia Signal Processing(MMSP), Vancouuer,BC,Canada,2018:1-5.

[13] Parviainen M, Pertilä P, Virtanen T, et al. Time-Frequency Masking Strategies for Single-Channel Low-Latency Speech Enhancement Using Neural Networks[C]//2018 IEEE 16th International Workshop on Acoustic Signal Enhancement(IWAENC),Tokyo,Japan,2018:51-55.

[14] Bhat G S,Shankar N,Reddy C K A,et al. A Real-Time Convolutional Neural Network Based Speech Enhancement for Hearing Impaired Listeners Using Smartphone[J]. IEEE Access,2019,7:78421-78433.

[15] Chen J,Ran X. Deep Learning with Edge Computing:A Review [J]. Proceedings of the IEEE,2019,107(8):1655-1674.

[16] Nurvitadhi E, Sim J, Sheffield D, et al. Accelerating Recurrent Neural Networks in Analytics Servers:Comparison of FPGA,CPU, GPU,and ASIC[C]//2016 26th International Conference on Field Programmable Logic and Applications (FPL), Lausanne Switzerland,IEEE,2016:1-4.

[17] Wang S,Li Z,Ding C W,et al. C-LSTM:Enabling Efficient LSTM Using Structured Compression Techniques on FPGAs[C]//2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays(FPGA),Monterey,CA,USA,2018:11-20.

[18] Zhu W T,Sun Y H,Shen Z Y,et al. Design of High Performance RNN Accelerator Based on Network Compression[C]//2020 IEEE 2nd International Conference on Circuits and Systems(ICCS), Chengdu,China,2020:6-10.

[19] Chang A X M,Culurciello E. Hardware Accelerators for Recurrent Neural Networks on FPGA[C]//2017 IEEE International Symposium on Circuits and Systems(ISCAS), Baltimore, MD, USA, 2017:1-4.

[20] Conti F, Cavigelli L, Paulin G, et al. Chipmunk: A Systolically Scalable 0.9 mm 2, 3.08 Gop/s/mW@ 1.2 mW Accelerator for Near-Sensor Recurrent Neural Network Inference[C]//2018 IEEE Custom Integrated Circuits Conference(CICC), San Diego, CA, USA,2018:1-4.

[21] Kadetotad D, Berisha V, Chakrabarti C, et al. A 8.93-TOPS/W LSTM Recurrent Neural Network Accelerator Featuring Hierarchical Coarse-Grain Sparsity with all Parameters Stored on-Chip[J]. IEEE Solid-State Circuits Letters,2019,2(9):119-122.

[22] Gao C,Rios-Navarro A,Chen X,et al. EdgeDRNN:Enabling Low-Latency Recurrent Neural Network Edge Inference[C]//2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems(AICAS),Genova,Italy,2020:41-45.

[23] 黄翔东,王兆华. 一种设计频率特性有间断滤波器的新方法. 天津大学学报,2006,39(5):614-620.

[24] Elman J L. Finding Structure in Time[J]. Cognitive Science, 1990,14(2):179-211.

[25] Cho K,Merrienboer B V,Gulcehre C,et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation [C]//2014 Conference on Empirical Methods in Natural Language Processing,Doha,Qatar,2014:1724-1734.

[26] Srivastava R K,Greff K,Schmidhuber J. Training Very Deep Networks[C]//NIPS 2015: The 28th International Conference on Neural Information Processing Systems,Montereal,Canada,2015:

2377-2385.

[27] Wang D,Zhang X W. THCHS-30:A Free Chinese Speech Corpus. Computer Science[EB/OL]. (2015-12-10). http://arxiv.org/abs/1512.01882v1.

[28] Varga A,Steeneken H. Assessment for Automatic Speech Recognition:II. NOISEX-92:A Database and an Experiment to Study the Effect of Additive Noise on Speech Recognition Systems[J]. Speech Communication,1993,12(3):247-251.

[29] Taal C H,Hendriks R C,Heusdens R,et al. An Algorithm for Intelligibility Prediction of Time-Frequency Weighted Noisy Speech[J]. IEEE Transactions on Audio,Speech,and Language Processing,2011,19(7):2125-2136.

[30] Rix A W,Beerends J G,Hollier M P,et al. Perceptual Evaluation of Speech Quality(PESQ)—A New Method for Speech Quality Assessment of Telephone Networks and Codecs[C]//2001 IEEE International Conference on Acoustics, Speech, and Signal Processing,Salt Lake City,UT,USA,2001:749-752.

[31] Cohen I. Optimal Speech Enhancement Under Signal Presence Uncertainty Using Log-Spectral Amplitude Estimator[J]. IEEE Signal Processing Letters,2002,9(4):113-116.

郎　标(1995—),男,天津大学微电子学院硕士研究生,研究课题为基于循环神经网络的语音增强加速器;

陈　霏(1981—),男,清华大学微纳电子系博士,深圳清华大学研究院副教授。主要研究方向为集成电路设计,应用领域为面向医疗健康的智能电路与系统、助听器械等,特色为将 IC 设计与深度学习算法和低功耗无线通信技术相融合。担任 IEEE 期刊 T-BioCAS,T-MTT,TCAS-I、II 等审稿人,chenfei@tsinghua-sz.org。