ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context

ContextNet：使用全局上下文的改进卷积神经网络用于自动语音识别

*Wei Han, Zhengdong Zhang, Yu Zhang, Jiahui Yu, Chung-Cheng Chiu, James Qin, Anmol Gulati, Ruoming Pang, Yonghui Wu*

Google Inc.

# 摘要

Convolutional neural networks (CNN) have shown promising results for end-to-end speech recognition, albeit still behind RNN/transformer based models in performance. In this paper, we study how to bridge this gap and go beyond with a novel CNN-RNN-transducer architecture, which we call ContextNet. ContextNet features a fully convolutional encoder that incorporates global context information into convolution layers by adding squeeze-and-excitation modules. In addition, we propose a simple scaling method that scales the widths of ContextNet that achieves good trade-off between computation and accuracy.

We demonstrate that on the widely used Librispeech benchmark, ContextNet achieves a word error rate (WER) of 2.1%/4.6% without external language model (LM), 1.9%/4.1% with LM and 2.9%/7.0% with only 10M parameters on the clean/noisy LibriSpeech test sets. This compares to the best previously published model of 2.0%/4.6% with LM and 3.9%/11.3% with 20M parameters. The superiority of the proposed ContextNet model is also verified on a much larger internal dataset.

**Index Terms**: speech recognition, convolutional neural networks

卷积神经网络（CNN）在端到端语音识别中显示出了良好的效果，尽管在性能上仍然落后于基于RNN/transformer的模型。在本文中，我们研究如何通过一种新的CNN-RNN-transducer结构，我们称之为ContextNet，弥补并超越了这一差距。ContextNet是一个完全卷积编码器，通过添加压缩和激励（squeeze-and-excitation）模块，将全局上下文信息合并到卷积层中。此外，我们提出了一种简单的缩放方法来缩放ContextNet的宽度，在计算量和精度之间取得了良好的平衡。

我们证明，在广泛使用的Librispeech基准上，ContextNet在没有外部语言模型（LM）的情况下，在clean/noisy测试集上，实现了2.1%/4.6%的字错误率，在使用LM时达到了1.9%/4.1%，在10M参数模型上达到2.9%/7.0%的字错误率。而相较与之前发布的最佳模型相，它们使用LM时为2.0%/4.6%，20M参数模型为3.9%/11.3%。在一个更大的内部数据集上，也验证了我们提出的ContextNet模型的优越性。

**关键词**：语音识别，卷积神经网络

2020.05.07 arxiv

# 1.     介绍

基于卷积神经网络（CNN）的端到端（E2E）语音识别模型正引起越来越多的关注[1，2，3[[1]](#footnote-1)，4[[2]](#footnote-2)]。其中，Jasper模型[3]最近在LibriSpeech test-clean[5]上通过外部神经语言模型实现了接近最新水平的单词错误率（WER）2.95%。Jasper模型的主要特点是一个基于深度卷积的编码器，它由一维卷积和跳跃连接组成的层堆积而成。深度可分离卷积[6]被用来进一步提高CNN模型的速度和精度[7[[3]](#footnote-3)，4]。基于CNN的模型的关键优势在于其参数效率；然而，最佳CNN模型QuartzNet[4]实现的WER仍然落后于基于RNN/transformer的模型[8、9[[4]](#footnote-4)、10[[5]](#footnote-5)、11[[6]](#footnote-6)]。

基于RNN/Transformer[9，10，11]的模型和CNN模型之间的一个主要区别是上下文的长度。在双向RNN模型中，理论上一个单元可以访问整个序列的信息；在Transformer模型中，注意机制显式地允许具有两个遥远时间戳的节点相互参与。然而，具有有限内核大小的朴素卷积只覆盖时域中的一个小窗口，因此上下文很小，并且不包含全局信息。在本文中，我们认为，全局信息的缺乏是导致基于CNN的ASR模型与基于RNN/Transformer的模型之间的WER差距的主要原因。

为了增强CNN模型中的全局上下文，我们从[12[[7]](#footnote-7)]中引入的squeeze-and-excitation（SE）层中得到启发，并提出了一种新的用于ASR的CNN模型，我们称之为ContextNet。SE层将一系列局部特征向量压缩成单个全局上下文向量，将该上下文广播回每个局部特征向量，并通过乘法将两者合并。当我们在原始卷积层之后放置SE层时，我们授予卷积输出对全局信息的访问权。从经验上看，我们观察到在ContextNet中添加squeeze-and-excitation层会最大程度地减少LibriSpeech test-clean的WER。

以前关于混合ASR的工作已经成功地将上下文引入声学模型，方法是堆叠大量的层，或者使用单独训练的全局向量来表示说话人和环境信息[13，14，15，16]。在[17]中，SE被用于RNN的无监督自适应。在本文中，我们证明了SE对于CNN编码器也是有效的。

ContextNet的体系结构也受到QuartzNet[4]的设计选择的启发，例如在编码器中使用深度可分离的一维卷积。然而，除了合并SE层之外，在体系结构中还有一些关键的区别。例如，我们使用RNN-T解码器[18、19、20、21]而不是CTC解码器[22]。此外，我们使用Swish激活函数[23]，这有助于轻微但一致地降低WER。总的来说，ContextNet在LibriSpeech test-clean/test-other上的WER为1.9%/4.1%。这比以前基于CNN的架构（如QuartzNet[4]）有很大的改进，它的性能优于基于transformer和LSTM的模型[11、10、24、9、25]。

本文还研究了如何降低ContextNet的计算量，以提高训练和推理的速度。首先，我们采用视觉模型中常用的渐进式下采样方案。具体来说，我们逐步减少编码序列，达到原长度的1/8，显著降低计算量，同时保持编码器的表示能力和整体模型精度。作为一个优点，这种下采样方案允许我们将所有卷积层的内核大小减少到5，而不会显著降低编码器输出节点的有效接收场。

我们可以通过参数全局改变卷积滤波器中通道的数目来扩展ContextNet。图1说明了ContextNet在模型大小和WER之间的权衡，以及与其他方法的比较。显然，我们的缩放模型实现了所有模型中最好的权衡。

综上所述，本文的主要贡献有：（1）提出了一种改进的具有全局上下文的CNN架构，（2） 一种逐步降采样和模型缩放方案实现了更高的精度和模型大小的权衡。

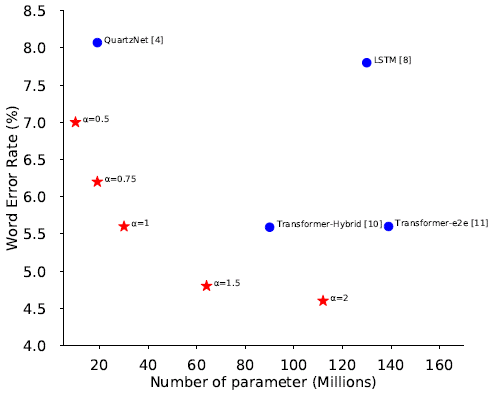


图1：LibriSpeech test-other WER vs model size。上述模型都没有外部LM。对于transformer混合模型，我们使用编码器的大小。ContextNet数字以红色突出显示。*α是第2.2.6节中讨论的模型比例参数。详细结果见表2。*

# 2.     模型

本节介绍ContextNet的体系结构细节。第2.1节讨论ContextNet的高层设计。然后第2.2节介绍了卷积编码器，并讨论了如何在保持模型精度的同时，逐步减少网络中输入语句的时间长度以减少计算量。

## 2.1.    端到端网络：CNN-RNN-Transducer

我们的网络基于RNN-Transducer框架[18，19，20]。该网络由三个部分组成：输入语音上的音频编码器、输入标签上的标签编码器以及将二者结合起来进行解码的联合网络。我们直接使用了文献[20]中基于LSTM的标签编码器和联合网络，提出了一种新的基于CNN的音频编码器。

## 2.2.    编码器设计

让输入序列为x=（x1，…，xT）。编码器将原始信号转换为high level表示h=（h1，…，hT’），其中T’≤T。基于卷积的音频编码器（·）定义为：

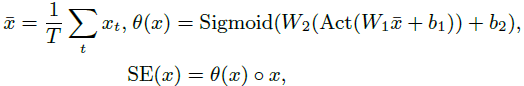


其中每个Ck（·）定义一个卷积块。它包含几个卷积层，每个卷积层后面都是批量标准化[26]和激活函数。它还包括squeeze-and-excitation组件[12]和跳跃连接[27]。

在介绍C（·）的细节之前，我们首先阐述C（·）中的重要模块。

*2.2.1. squeeze-and-excitation*

如图2所示，squeeze-and-excitation [12]函数SE（·）对输入执行全局平均池化，将其转换为全局channelwise weight θ（x），并按元素将每个帧乘以该权重。我们把这个想法应用到1D情景中，



其中◦表示element-wise乘法，w1 w2是权重矩阵，b1 b2是偏差向量。

*2.2.2. 深度可分卷积*

让conv（·）表示编码器中使用的卷积函数。在本文中，我们选择深方向可分离卷积作为conv（·），因为这样的设计已经在各种应用[6，4，28]中展示过，以在不影响精度的情况下获得更好的参数效率。

为了简单起见，我们在网络中的所有深度卷积层上使用相同的内核大小。

*2.2.3. Swish激活功能*

让Act（·）表示编码器中的激活函数。为了选择Act（·），我们对ReLU和swish函数进行了实验[23]，swish定义如下：



其中设置*β*=1，在所有实验中。我们观察到swish函数始终比ReLU工作得更好。

*2.2.4. 卷积块*

在介绍了所有单独的模块之后，我们现在给出了方程（1）中的卷积块（·）。图3展示了（·）的高级体系结构。一个块（·）可以包含几个Conv（·）函数；m设为Conv（·）函数的个数。设BM（·）为批标准化[29]。我们将每一层定义为*f*（x）=Act（BN（Conv（x））。因此，

C（x）=Act（SE（fm（x））+P（x）），其中fm表示函数（·）在输入上的叠加m层，P（·）表示残差上的逐点投影函数。通过稍微重用符号，我们允许第一层和最后一层与其他m−2层不同：如果块需要将输入序列降采样为1半，则最后一层的步长为2，而其余的−1层的步长为1；否则，所有层的步长为1。此外，如果该块有一个输入通道数Din和输出通道数Dout，则第一层*f*将通道Din转换为通道Dout，而其余m-1层将通道数保持为Dout。按照惯例，投影函数P具有与第一层相同的步长。

*2.2.5. 渐进下采样*

我们使用跨步卷积进行时间下采样。更多的下采样层降低了计算成本，但是编码器中过多的下采样可能会对解码器产生负面影响。从经验上看，我们发现渐进式8×下采样方案在速度和精度之间取得了很好的折衷。这些权衡在第3.3节中讨论。

*2.2.6. ContextNet的配置详细信息*

ContextNet有23个卷积块。所有卷积块都有五层卷积，除了和*C*0*，C*22，它们只有一层卷积。表1总结了架构细节。请注意，全局参数*α*控制模型的缩放。当增大*α*使*α >*1时，卷积的通道数增加，使模型在更大的模型尺寸下具有更强的表示能力。

|  |
| --- |
| 图2:1D squeeze-and-excitation模块。输入首先经过卷积层，然后进行批量规范化和激活。然后采用平均池法将conv结果压缩成一维向量，然后由两个具有激活函数的完全连接（FC）层形成的瓶颈结构进行处理。输出通过要映射到（0,1）的Sigmoid函数*，然后使用逐点乘法平铺并应用于conv输出。* |

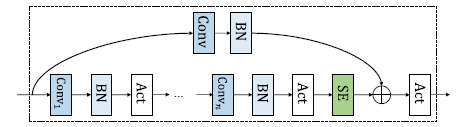
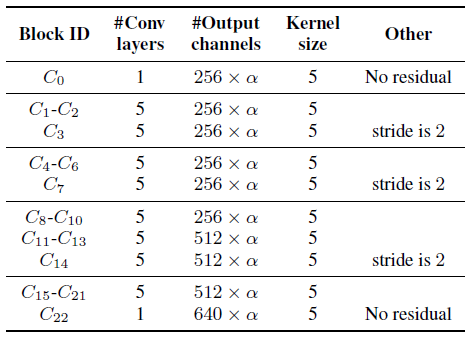


图3：卷积块*Ci包含许多卷积，每一个卷积后面都是批量规范化和激活函数。（SE）块对最后一个卷积层的输出进行操作。在SE模块的输出端采用带投影的跳接。*

表1:ContextNet编码器的配置。*α控制输出通道的数量，从而控制模型的缩放。内核大小是时域中的窗口大小；卷积是跨频率的。如第2.2节所述，如果卷积块的步幅为2，则其最后一个conv层的步幅为2，而其余conv层的步幅为1。*



# 3.     实验

我们在Librispeech[5]数据集上进行了实验，该数据集由970小时的标记语音和一个额外的纯文本语料库组成，用于建立语言模型。我们提取80维滤波器组特征，使用25ms窗口，步长为10ms。

我们使用Adam优化器[30]和transformer学习率方案[31]，搭配15k warm-up steps，峰值学习率为0.0025。在网络中的所有可训练权重中还添加了一个具有10-6权重的l2正则化。我们使用单层LSTM作为解码器，输入维为640。在译码器中引入变分噪声作为正则化。

我们使用SpecAugment[25，8]，mask参数（F=27），使用ten time masks和最大时间掩码比率（ps=0.05），其中时间掩码的最大大小设置为长度的ps倍。不使用时间扭曲。我们使用3层LSTM LM，宽度4096，在LibriSpeech langauge模型语料库上训练，添加LibriSpeech 960h转录文本，用LibriSpeech 960h构建的1k WPM标记。LM在开发集转录文本上有63.9的单词级困惑度。浅层融合的LM权重通过网格搜索在dev集上进行调整。所有模型都是用Lingvo工具箱实现的[32]。

## 3.1.    LibriSpeech结果

我们评估了LibriSpeech上ContextNet的三种不同配置。模型均基于表1，但网络宽度不同，因此模型大小不同。具体来说，我们的*α*在{0.5,1,2}中取值，得到小型、中型和大型ContextNet。我们还建立了自己的LSTM基线作为参考。

表2总结了评估结果以及与以前公布的几个系统的比较。结果表明ContextNet比以前发布的系统有改进。我们的中等模型ContextNet（M）只有31M个参数，但与其他更大的系统[11，33]相比，有相似的WER。大型模型ContextNet（L）在test-clean和test-other上的性能分别比之前的SOTA好13%和18%。我们的缩小模型ContextNet（S）也显示了对以前类似大小的系统的改进[4]，无论是否有语言模型。

## 3.2.    上下文大小的影响

为了验证为ASR的CNN模型添加全局上下文的有效性，我们对squeeze-and-excitation模块如何影响LibriSpeech test-clean/test-other的WER进行了对比研究。表1中的ContextNet去掉了所有的squeeze-and-excitation模块，*α*=1.25作为上下文的基线。

squeeze-and-excitation将整个话语作为context。为了研究不同上下文大小的影响，我们将squeeze-and-excitation模块中的全局平均池化替换为stride-one pooling操作，其中上下文可以由池窗口的大小控制。在这项研究中，我们在所有卷积块上，做窗口大小256、512和1024的比较。

表2:LibriSpeech WER。与以前的模型相比，ContextNet在使用和不使用语言模型的情况下都取得了优异的性能

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | #params(m) | Without lm | | With lm |  |
|  |  | dev clean | dev other | test clean | test other |
| Hybird  transformer[10] | - | - | - | 2.26 | 4.85 |
| CTC  QuartzNet（CNN）[4] | 19 | 3.90 | 11.28 | 2.69 | 7.25 |
| LAS  transformer [9] | - | - | - | 2.6 | 5.7 |
| transformer [33] | 270 | 2.89 | 6.98 | 2.33 | 5.17 |
| LSTM | 360 | 2.6 | 6.0 | 2.2 | 5.2 |
| Transducer  transformer [11] | 139 | 2.4 | 5.6 | 2.0 | 4.6 |
| This Work  ContextNet(S) | 10.8 | 2.9 | 7.0 | 2.3 | 5.5 |
| ContextNet(M) | 31.4 | 2.4 | 5.4 | 2.0 | 4.5 |
| ContextNet(L) | 112.7 | 2.1 | 4.6 | 1.9 | 4.1 |

如表3所示，SE模块比基线有了重大改进。此外，随着上下文窗口长度的增加，好处也会越来越大。这与SE在图像分类模型上的类似研究中的观察结果一致[34]。

表3：上下文窗口大小对WER的影响。所有模型均为*α*=1.25*.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| context | dev clean | dev other | test clean | test other |
| None | 2.6 | 7,0 | 2.6 | 6.9 |
| 256 | 2.1 | 5.4 | 2.3 | 5.5 |
| 512 | 2.1 | 5.1 | 2.3 | 5.2 |
| 1024 | 2.1 | 5.0 | 2.3 | 5.1 |
| global | 2.0 | 4.9 | 2.3 | 4.9 |

## 3.3.    深度、宽度、核大小和下采样

*深度：*我们对卷积块的数量进行遍历，我们的最佳配置在表1中。我们发现在这种配置下，我们可以在一天内训练一个稳定收敛的模型。

*宽度：*我们在所有编码器层上全局缩放网络的宽度（即通道的数量），并研究它如何影响模型性能。具体来说，我们从表1中获取ContextNet模型，遍历*α*，并报告模型大小和LibriSpeech的WER。表5总结了结果；它展示了模型大小和ContextNet的WER之间的良好权衡。

表4：时间缩减和卷积核大小对FLOPS和模型精度的影响。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reduction | Kernel size | GFLOPS[2] | test clean | test other |
| 2倍 | 3 | 2.131 | 2.7 | 6.3 |
|  | 5 | 2.137 | 2.6 | 5.8 |
|  | 11 | 2.156 | 2.4 | 5.4 |
|  | 23 | 2.194 | 2.3 | 5.0 |
| 8倍 | 3 | 1.036 | 2.3 | 5.1 |
|  | 5 | 1.040 | 2.3 | 5.0 |
|  | 11 | 1.050 | 2.3 | 5.0 |
|  | 23 | 1.071 | 2.3 | 5.2 |

表5：网络宽度模型缩放对WER的影响。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *α* | #Params(M) | dev clean | dev other | test clean | test other |
| 0.5 | 10.8 | 2.7 | 7.0 | 2.9 | 7.0 |
| 1 | 31.4 | 2.2 | 5.1 | 2.4 | 5.4 |
| 1.5 | 65.4 | 2.0 | 4.7 | 2.2 | 4.8 |
| 2 | 112.7 | 2.0 | 4.6 | 2.1 | 4.6 |

*下采样和内核大小：*表4总结了在不同的下采样和文件大小选择时，在LibriSpeech上的FLOPS和WER。我们使用同一个模型，只添加一个下采样层作为基线，因此基线只进行2倍的时间缩减。我们在{3,5,11,21}中扫描内核大小，每个内核大小应用于所有深度卷积层。结果表明，渐进式下采样可以显著减少FLOPS。此外，它实际上对模型的准确性有一些好处。此外，随着逐步下采样，增加核大小会降低模型的WER。

## 3.4.    大规模实验

最后，我们证明了所提出的架构在大规模数据集上也是有效的。我们使用了一个类似于[35]的实验设置，其中训练集有公开的Youtube视频，其中有[36]中方法生成的半监督转录文本。我们评估了117个视频，总时长为24.12小时。这套测试集具有多样性和挑战性的声学环境[3]。表6总结了结果。我们可以看到，ContextNet的性能比[35]中以前的最佳体系结构要好，后者是卷积和双向LSTM的组合，相对而言，在参数和FLOPS较少的情况下，性能提高了12%。

表6：在Youtube测试集上对比ContextNet与以前的最佳结果

|  |  |  |  |
| --- | --- | --- | --- |
| Model | #Params(M) | GFLOPS | Youtube wer |
| TDNN[35] | 192 | 3.834 | 9.3 |
| ContextNet | 112 | 2.647 | 8.2 |

# 4.     结论

在这项工作中，我们提出并评估了一个基于CNN的端到端语音识别架构。讨论并比较了几种建模方法。与以前发表的CNN模型相比，该模型在LibriSpeech基准上获得了更好的精度，参数更少。通过限制网络的宽度，该结构可以方便地用于搜索小型ASR模型。对一个更大更具挑战性的数据集的初步研究也证实了我们的发现。

# 5.     参考文献

[1] Y. Zhang, W. Chan, and N. Jaitly, “Very deep convolutional networks for end-to-end speech recognition,” in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017, pp. 4845–4849.

[2] N. Zeghidour, Q. Xu, V. Liptchinsky, N. Usunier, G. Synnaeve, and R. Collobert, “Fully convolutional speech recognition,” arXiv preprint arXiv:1812.06864, 2018.

[3] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde, “Jasper: An end-to-end convolutional neural acoustic model,” arXiv preprint arXiv:1904.03288, 2019.

[4] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, and Y. Zhang, “Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions,” arXiv preprint arXiv:1910.10261, 2019.

[5] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015, pp. 5206–5210.

[6] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.

[7] A. Hannun, A. Lee, Q. Xu, and R. Collobert, “Sequence-tosequence speech recognition with time-depth separable convolutions,” arXiv preprint arXiv:1904.02619, 2019.

[8] D. S. Park, Y. Zhang, C.-C. Chiu, Y. Chen, B. Li, W. Chan, Q. V. Le, and Y.Wu, “Specaugment on large scale datasets,” in ICASSP, 2020.

[9] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang et al., “A comparative study on transformer vs rnn in speech applications,” arXiv preprint arXiv:1909.06317, 2019.

[10] Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang et al., “Transformer-based acoustic modeling for hybrid speech recognition,” arXiv preprint arXiv:1910.09799, 2019.

[11] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, “Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss,” arXiv preprint arXiv:2002.02562, 2020.

[12] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

[13] V. Peddinti, G. Chen, V. Manohar, T. Ko, D. Povey, and S. Khudanpur, “Jhu aspire system: Robust lvcsr with tdnns, ivector adaptation and rnn-lms,” in 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE, 2015, pp. 539– 546.

[14] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, and Q. Liu, “Fast adaptation of deep neural network based on discriminant codes for speech recognition,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, no. 12, pp. 1713–1725, 2014.

[15] M. Karafia´t, L. Burget, P. Mateˇjka, O. Glembek, and J. Cˇ ernocky`, “ivector-based discriminative adaptation for automatic speech recognition,” in 2011 IEEE Workshop on Automatic Speech Recognition & Understanding. IEEE, 2011, pp. 152–157.

[16] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, “Speaker adaptation of neural network acoustic models using i-vectors,” in 2013 IEEE Workshop on Automatic Speech Recognition and Understanding. IEEE, 2013, pp. 55–59.

[17] H. B. Sailor, S. Deena, M. A. Jalal, R. Lileikyte, and T. Hain, “Unsupervised adaptation of acoustic models for asr using utterancelevel embeddings from squeeze and excitation networks,” in 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2019, pp. 980–987.

[18] A. Graves, “Sequence transduction with recurrent neural networks,” arXiv preprint arXiv:1211.3711, 2012.

[19] K. Rao, H. Sak, and R. Prabhavalkar, “Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer,” in 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2017, pp. 193–199.

[20] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S. yiin Chang, K. Rao, and A. Gruenstein, “Streaming end-to-end speech recognition for mobile devices,” in ICASSP, 2019.

[21] T. N. Sainath, Y. He, B. Li, A. Narayanan, R. Pang, A. Bruguier, S.-y. Chang, W. Li, R. Alvarez, Z. Chen, and et al., “A streaming on-device end-to-end model surpassing server-side conventional model quality and latency,” in ICASSP, 2020.

[22] A. Graves, S. Fern´andez, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in Proceedings of the 23rd international conference on Machine learning, 2006, pp. 369–376.

[23] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions,” arXiv preprint arXiv:1710.05941, 2017.

[24] A. Zeyer, P. Bahar, K. Irie, R. Schl¨uter, and H. Ney, “A comparison of transformer and lstm encoder decoder models for asr,” in IEEE Automatic Speech Recognition and Understanding Workshop, Sentosa, Singapore, 2019.

[25] D. S. Park,W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” in Interspeech, 2019.

[26] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016.

[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.

[29] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” arXiv preprint arXiv:1502.03167, 2015.

[30] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in neural information processing systems, 2017, pp. 5998–6008.

[32] J. Shen, P. Nguyen, Y. Wu, Z. Chen, and et al., “Lingvo: a modular and scalable framework for sequence-to-sequence modeling,” 2019.

[33] G. Synnaeve, Q. Xu, J. Kahn, T. Likhomanenko, E. Grave, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, “End-toend asr: from supervised to semi-supervised learning with modern architectures,” 2019.

[34] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, “Gatherexcite: Exploiting feature context in convolutional neural networks,” in Advances in Neural Information Processing Systems, 2018, pp. 9401–9411.

[35] C.-C. Chiu, A. Narayanan, W. Han, R. Prabhavalkar, Y. Zhang, N. Jaitly, R. Pang, T. N. Sainath, P. Nguyen, L. Cao, and Y. Wu, “Rnn-t models fail to generalize to out-of-domain audio: Causes and solutions,” arXiv preprint arXiv:2005.03271, 2020.

[36] H. Liao, E. McDermott, and A. Senior, “Large scale deep neural network acoustic modeling with semi-supervised training data for youtube video transcription,” in 2013 IEEE Workshop on Automatic Speech Recognition and Understanding. IEEE, 2013, pp.368–373.

1. *Jasper: An end-to-end convolutional neural acoustic model*. 2019 [↑](#footnote-ref-1)
2. *Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions*. 2019 [↑](#footnote-ref-2)
3. *Sequence-to-sequence speech recognition with time-depth separable convolutions*. 2019 [↑](#footnote-ref-3)
4. *A comparative study on transformer vs rnn in speech applications*. 2019 [↑](#footnote-ref-4)
5. *Transformer-based acoustic modeling for hybrid speech recognition*. 2019 [↑](#footnote-ref-5)
6. *Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss*. 2020 [↑](#footnote-ref-6)
7. *Squeeze-and-excitation networks*. 2018 [↑](#footnote-ref-7)