Transformer Transducer: One Model Unifying Streaming And Non-Streaming Speech Recognition

变压器换能器：一种统一流与非流语音识别的模型

*Anshuman Tripathi、Jaeyoung Kim、Qian Zhang、Han Lu、Hasim Sak*

{anshumant、jaeykim、zhaqian、luha、hasim}@谷歌网站

美国谷歌公司

# 摘要

In this paper we present a Transformer-Transducer model architecture and a training technique to unify streaming and non-streaming speech recognition models into one model. The model is composed of a stack of transformer layers for audio encoding with no lookahead or right context and an additional stack of transformer layers on top trained with variable right context. In inference time, the context length for the variable context layers can be changed to trade off the latency and the accuracy of the model. We also show that we can run this model in a Y-model architecture with the top layers running in parallel in low latency and high latency modes. This allows us to have streaming speech recognition results with limited latency and delayed speech recognition results with large improvements in accuracy (20% relative improvement for voice-search task). We show that with limited right context (1-2 seconds of audio) and small additional latency (50-100 milliseconds) at the end of decoding, we can achieve similar accuracy with models using unlimited audio right context. We also present optimizations for audio and label encoders to speed up the inference in streaming and non-streaming speech decoding.

Index Terms— Transformer, RNN-T, sequence-to-sequence, encoder-decoder, end-to-end, speech recognition

本文提出了一种变压器-换能器模型结构和一种训练技术，将流式和非流式语音识别模型统一为一个模型。该模型由一个用于音频编码的变换层堆栈和一个附加的变换层堆栈组成，变换层堆栈没有前向上下文或右上下文，而变换层堆栈在顶部使用可变右上下文进行训练。在推理时，可以改变可变上下文层的上下文长度来权衡模型的延迟和准确性。我们还表明，我们可以在Y-model架构中运行该模型，顶层以低延迟和高延迟模式并行运行。这使我们能够以有限的延迟获得流式语音识别结果，并以较大的精度提高延迟语音识别结果（语音搜索任务相对提高20%）。我们证明，在有限的右上下文（1-2秒的音频）和解码结束时较小的额外延迟（50-100毫秒）的情况下，我们可以达到与使用无限音频右上下文的模型相似的精度。我们还提出了优化的音频和标签编码器，以加快推理流和非流语音解码。

*索引项*-变压器，RNN-T，序列到序列，编解码器，端到端，语音识别

# 1.    导言

过去的研究表明，在神经网络模型中访问未来的音频上下文对当前音频帧进行编码可以显著提高语音识别的准确性[1，2，3，4]。双向LSTMs利用未来的音频上下文，但是模型只能在整个音频可用时运行。在过去的几年中，采用自我注意机制的模型在序列建模任务中取得了最新的成果[5,6]。变压器模型通过在序列中每个输入的左右上下文窗口上运行自关注机制来编码输入序列。在语音识别模型中，未来的音频上下文可以通过指定有限的右上下文进行编码[4]。由于正确的上下文是有限的（与双向LSTMs不同），这使得具有未来音频上下文的转换器模型能够以流式方式识别具有一定延迟的语音。这使得transformer模型特别适用于能够提供更高流延迟以获得更好识别质量的应用程序。

即使在低延迟流式语音识别系统（如语音搜索、听写）中，系统的可用性也取决于最终识别结果的准确性。采用n-best或格点重评分技术可以提高最终结果的准确性。对于长形音频，由于n-best假设缺乏多样性，重定模型引入的额外延迟是没有限制的，并且精确度的提高也是有限的。为了解决这个问题，在本文中，我们提出了使用两个转换器模型并行解码语音-一个较小的右上下文用于低延迟流识别结果，另一个较大的右上下文用于最终结果。与重新排序不同的是，这种方法适用于短话语和长话语。我们发现，相对于语音搜索任务的低延迟结果，它可以提高大约20%的相对字错误率。由于受限的右上下文，较大的右上下文解码引入的额外延迟也是有界的。

我们进一步证明，我们可以通过训练一个可以在低延迟或高延迟模式下解码语音的模型来统一低延迟和高延迟模型。我们通过在训练期间改变变压器层（可变上下文层）的正确上下文来实现这一点。通过仅使用最后几层作为可变上下文层（称为Y-model结构），并应用本文提出的解码优化，我们证明了使用一个ASR模型对高延迟和低延迟模式进行有效的并行解码是可能的。

# 2.    变压器传感器

## 2.1.    建筑

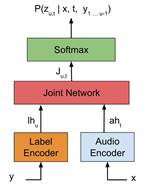


图1。变压器-传感器结构。

Transformer Transducer[4]是一种模型架构，可以使用基于Transformer的音频编码器和标签编码器，使用端到端RNN-T损耗[7]进行训练。如图1所示，T-T模型预测了每个时间步上标签空间的概率分布。排列的概率（z | x）可以分解为*第*

*,*标签（z1:（−1）i）），(1)

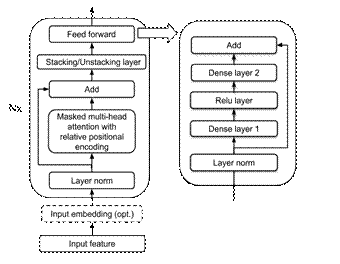


图2。变压器编码器结构。

其中标签（z1:（i−1））是中非空白标签的序列。在T-T架构中（z | x）通过音频编码器、标签编码器和联合网络进行参数化。模型定义（zi | x，ti，标签（z1:（i−1）））如下：*z轴*1： （一−1）*第第*

接头=线性（音频编码器（x））+*我*

（2） 线性（LabelEncoder（标签（z1:（i−1）））

*第*（z | x，t标签（z1:（i−1）））=*我我,*

（3） Softmax（线性（tanh（接头））），

其中每个线性函数是不同的单层前馈神经网络，AudioEncodert（x）是当时的音频编码器输出，LabelEncoder（Labels（z1:（i−1）））是给定先前非空白标签序列的标签编码器输出。*我钛*

关于每个变压器层的更多细节如图2所示。它包括归一化层、带相对位置编码的屏蔽多头注意层、残差连接层、叠加/去叠加层和前馈层。剩余连接通过归一化输入应用到注意层或前馈层的输出。叠加/去叠加层可以用来改变每个变换层的帧速率，这有助于加快训练和推理。为了进一步优化，标签编码器也可以是bigram标签嵌入模型。

## 2.2.    变语境训练

在transformer结构的自注意块中，我们计算整个输入序列上的自注意，然后根据层的左右上下文对其进行屏蔽。在可变上下文训练中，我们保持层的左上下文不变，并从给定的分布中采样右上下文长度。正确的语境长度决定了自我注意的面具。在我们的实验中，我们发现对每一层随机抽取正确的上下文长度会导致不稳定的训练，并且我们必须限制正确的上下文配置的数量。在培训期间，我们将右上下文配置指定为编码器中每个转换器层的右上下文列表，例如：[0]×15+[8]×5指定前15层的右上下文为0，最后5层的右上下文为8帧。在培训过程中，我们从所有这些可用配置中统一抽取一个正确的上下文配置。我们在第4.3.2节中展示了以这种方式训练的模型可以与用于训练的任何正确的上下文配置一起使用。

## 2.3.    Y型

可变上下文训练允许我们训练变压器层，这些层能够在推理时有效地使用不同于输入的正确上下文。我们应用这种技术来训练一个模型，该模型的输入后面紧跟着几个初始层，这些初始层使用零右上下文进行训练，最后几层作为可变上下文层进行训练。利用该模型，我们可以对不同的未来音频上下文（取决于推理时使用的正确上下文）进行语音识别。如果应用程序能够承受更高的延迟以获得更高质量的结果，这将非常有用。我们还可以通过运行两个使用相同模型的并行解码器来使用此模型进行低延迟语音识别，一个没有或非常小的右上下文，另一个具有较大的右上下文。因为我们保持共享层的正确上下文不变（为零），所以我们只需要为具有可变上下文的最终层重新计算激活。我们称之为Y模型，因为我们有并行解码分支运行在不同的正确上下文中。我们称右上下文较小的解码分支为“低延迟分支”，右上下文较高的解码分支为“高延迟分支”

### 2.3.1.    推理与识别潜伏期

在推理过程中，我们将基于低延迟分支的部分识别结果流化（以获得更好的模型响应性），当话语结束时，我们将识别结果替换为来自高延迟分支的结果（以获得更好的ASR质量）。根据用于高延迟分支的正确上下文（在我们的实验中为2.4秒），在有限量的解码中，它总是在低延迟分支之后。当用户话语结束时，我们只需要对剩余的lookahead（例如2.4秒）运行高延迟分支解码并显示最终结果。因为我们可以访问所有剩余的音频上下文，所以我们可以通过批处理所有可用的上下文来非常快速地处理它（无需等待流式处理）。在话语结束时显示这些最终结果的额外延迟包括用于计算在延迟分支和解码中具有可变上下文的音频编码器层的激活的运行时。在第4.3.2节中，我们展示了对于2.4秒的先行音频上下文，这可以非常快速地完成，因此导致非常小的额外延迟。

## 2.4.    约束对齐训练

Transformer Transducer最小化RNN-T损耗，该损耗包含每个标签序列的所有对齐路径。尽管优化标签概率有利于提高ASR性能，但由于在优化RNN-T损失时没有限制预测延迟的机制，模型可能具有较高的对齐延迟。特别是对于没有正确上下文的流模型，由于该模型试图通过展望未来帧来提高预测精度，因此会出现高对齐延迟。

约束对齐训练最初是在[8]中提出的。它通过从参考对准中屏蔽高延迟对准路径来限制RNN-T丢失。为了找到参考校准，我们使用了全神贯注的非流变压器-传感器模型作为参考校准模型，因为它几乎没有延迟。在参考对齐的基础上，我们将约束窗口放置在单词边界上，并将约束窗口外的任何单词标签路径从RNN-T丢失中屏蔽出来。我们在Y模型上应用了约束对齐训练，其评估结果见第4.3.2节。

# 3.    推理优化

在本节中，我们将讨论在流式和非流式应用程序中加速Transformer编码器以及用于语音解码的标签编码器所做的优化和实现。这是使Y模型的并行译码可行所必需的。

## 3.1.    流变压器音频编码器

流式音频编码是将音频帧和前一状态作为输入，并输出相应的编码特征和后一状态的过程。与基于RNN/LSTM的音频编码器不同，Transformer编码器在编码音频时没有时间依赖性。换言之，为了编码第-帧，基于RNN/LSTM的编码器需要在编码第-帧之前首先从第0帧编码到第-1帧，而变压器编码器可以并行地同时编码所有帧。在并行“批步”推理中，我们提到了对一批时间步进行编码的方法。在后面的部分中，我们将从推理速度的角度讨论批处理更多时间帧的有效性。*我我我我*

## 3.2.    非流变压器音频编码器

在非流式推理中，理想情况下，我们可以完全按照训练时运行的方式运行Transformer编码器。然而，在实际应用中，由于注意矩阵计算中的（T2）内存消耗，转换器编码器的内存非常昂贵。如[4]所示，有限上下文转换器编码器提供了与无限上下文编码器相同的质量。基于这种模型结构，我们可以一次计算几个块的注意矩阵，每个块只关注一组有限的s，这使得内存消耗是恒定的。我们将这种推理方法称为“查询切片”。*O查询查询钥匙*

## 3.3.    解码器优化

在变压器-传感器模型中，我们使用标签编码器作为自回归模型来编码预测的标签历史。标签编码器输出与音频编码器输出相结合，在softmax函数之前使用密集层。译码速度与标签编码网络的计算复杂度密切相关，因为它是针对每个假设运行的。对于RNN-T模型，已经证明标签历史上下文长度可以显著减少，而不会影响模型的准确性[9]。在本文中，我们尝试了两种标签编码模型。第一种是具有有限标签上下文长度的转换器模型，如[4]所述。另一个是一个嵌入模型，带有一个bigram标签上下文。嵌入模型为每个可能的二元标签上下文学习一个维数的权重向量，其中是音频和标签编码器输出的维数。参数总数为\*d，其中是标签的词汇表大小。在T-T模型中，学习的权值向量被简单地用作二元标记上下文的嵌入。因为这是一个基于标签bigrams的简单嵌入查找，所以标签编码器的运行时间非常快。*d级d级不*2 *不*

为了进一步加速推理，我们维护了一个缓存来存储和重用计算出的标签编码器输出，用于目前为止在解码过程中看到的有限标签上下文，因为在解码过程中，模型将为不同的假设多次编码同一标签历史。

|  |  |  |
| --- | --- | --- |
| 标签编码器 | WER公司 | RTF公司 |
| 40字形上下文转换器 | 4.8 | 0.3 |
| 3字形上下文转换器 | 4.8 | 0.02 |
| 2个图形集emb查找 | 4.9 | 0.01 |

表1。不同标签编码器的WER和RT系数。

# 4.    实验和结果

## 4.1.    数据

在我们的实验中，我们使用了来自voicesearch应用程序的3万小时的语音数据。我们使用的测试集由14K个语音搜索语句组成，持续时间小于5.5秒。训练集和测试集都是匿名的，都是手写的。输入的语音波形是用一个32毫秒的窗口加上10毫秒的位移来框定的。我们使用128维logmel能量特征，将其中的4个特征叠加，再以3的因子进行子采样，得到30毫秒的特征。在训练期间，我们对声学特征进行specaugment[10]。

## 4.2.    有限上下文解码

表1显示了不同标签编码器架构的WER和基准测试结果。从结果中我们可以看到，对于标签编码器，当上下文减少时，WER没有太大的差异。类似的结果在文献[9]中已有报道。对于3个图形的有限上下文的情况，我们看到解码器的速度比标签编码器有40个标签的情况有了很大的提高。这是因为标签编码器输出缓存，如第3.3节所述。对于label context 2，它甚至更快，因为模型本身只是一个查找表。

## 4.3.    Y模型结果

我们给出了两种Y结构模型的结果：

### 4.3.1.    Y型1

音频编码器有20L变压器，所有层都使用可变上下文进行训练，具有以下可能的正确上下文配置：[0]×19+[4]、[2]×20、[4]×20和[32]×20。在所有这些模式下，模型的训练输出延迟为4帧。当用240ms的前瞻性评估模型时，我们使用正确的上下文配置[0]×19+[4]，当用1.2s的前瞻性评估模型时，我们使用[2]×20，对于2.4秒的前瞻性，我们使用配置[4]×20

### 4.3.2.    Y型2

音频编码器有20L变压器，前15层训练时没有正确的上下文，后5层训练时有可变上下文，可能有以下正确的上下文配置：[0]×19+[4]、[0]×15+[8]×5和[0]×15+[16]×5。在所有这些模式下，模型的训练输出延迟为4帧。当用240ms的前瞻性评估模型时，我们使用正确的上下文配置[0]×19+[4]，当用1.2s的前瞻性评估模型时，我们使用[0]×15+[8]×5，对于2.4秒的前瞻性，我们使用配置[0]×15+[16]×5。

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 先行 | WER公司 | 对准延迟 |
| 全上下文模型 | 34秒 | 4.8 | 60秒 |
| 左上下文模型 | 240毫秒 | 6.1 | 982秒 |
| Y模型1 | 240毫秒  1.2秒 | 6.1  5.2 | 883秒  780毫秒 |
|  | 2.4秒 | 5.1 | 764秒 |
| Y型2 | 240毫秒  1.2秒 | 5.3  5.0 | 767秒  - |
|  | 2.4秒 | 5.0 | 742秒 |

表2。我们有不同的模特和造型。

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 先行 | WER公司 | 对准延迟 |
| Y型2 | 240毫秒  2.4秒 | 6.5  4.9 | 119毫秒74毫秒 |

表3。具有不同模型的WER和具有约束对齐训练的lookaheads。

表2显示了没有约束对齐训练的基线。我们可以看到，具有2.4秒高延迟分支的Y架构非常接近全注意模型的性能。在低延迟模式下的相同模型比没有正确上下文的最佳流模型具有更好的准确性。我们还显示了所有模型评估的词对齐延迟。由于Y-model2的低延迟分支仍然将输出延迟767毫秒，因此它可以获得更好的性能，因为它仍然通过延迟字向前看240毫秒+767毫秒。另一个有趣的观察是，Y模型的2.4秒先行模式也会延迟单词对齐，尽管完全注意模型不会延迟。这是意料之中的，因为延迟字有助于低延迟模式，而不影响高延迟模式。

表3显示了具有约束RNN-T丢失的Y模型的结果，我们可以看到现在预测的字要早得多，但是这导致低延迟分支的质量下降。较高延迟模式的质量仍然是相同的，因为它不受字延迟的太多影响。

对齐延迟定义为参考非流模型和约束对齐Y模型之间的平均字对齐差：

(4)

其中，是来自参考模型的单词的对齐时间，是来自Y模型的单词的对齐时间，N是单词总数。对于Y模型的低延迟模式，我们在最后一层使用额外的4个右上下文来减少WER丢失。此外，Y模型有4个输出延迟。因此，除了对齐延迟，Y模型对于低延迟模式还有额外的240毫秒。*钛*裁判*伊思钛*是的*伊思*

表3显示了约束对齐模型。与表2中的无约束Y模型2相比，对齐延迟显著改善：对于低延迟和高延迟模式，分别为767毫秒到119毫秒和742毫秒到74毫秒。此外，约束对齐高延迟模式没有显示任何WER损失，因为高延迟模式已经有足够的正确上下文，并且限制其预测延迟不会影响其性能。然而，对于

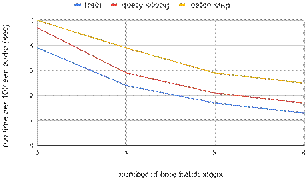


图3。基于TPU的8种不同模式的批处理推理速度。

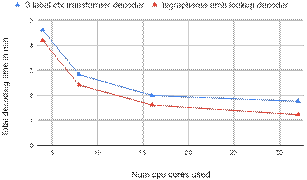


图4。CPU上查询切片模式的推理速度，36分钟音频。

低延迟模式，由于前瞻帧减少而降级。

对于Y模型，更精确的高延迟模式总是用来纠正来自低延迟模式的任何错误。因此，降低对齐延迟对于低延迟模式更为重要，而保持高质量性能是高延迟模式的主要关注点。约束对齐训练非常适合这种情况。

## 4.4.    推理基准

在图3中，我们根据编码器一次运行的时间步数，在一个TPU[11]内核上用不同的模式对100秒音频进行编码。我们可以看到，对于所有模式，当我们在一次推理运行中编码更多的时间步时，推理速度更快，因为它允许更好的并行化。我们还可以看到，训练模式比查询切片模式快，批处理步骤模式是其中最慢的，因为处理查询块和状态的额外开销。在高延迟场景中，编码器一次编码120帧，编码时间甚至更快。分别以训练模式、查询切片模式和批处理模式对100秒音频进行编码需要0.3秒、0.6秒和1.8秒。

在图4中，我们在桌面CPU上测试了36分钟长的音频的查询切片推理模式。我们可以看到，使用8个CPU核（即小于8%的实时系数）可以在不到3分钟的时间内识别出该音频。采用嵌入双字元的查找解码器，可以进一步提高识别时间。

# 5.    结论

我们描述了一种训练变压器-换能器模型的方法，该方法允许训练一个可以在低延迟和高延迟模式下解码语音的模型。我们还提出了倒数第二层用可变右上下文训练的Y模型作为统一低延迟和高延迟解码模式的有效解决方案。低延迟模式可用于流式识别结果，而最终的识别结果可从高延迟模式获得。我们表明，在未来2.4秒的小音频上下文中，我们可以获得与具有完整音频上下文的模型非常相似的精度。我们还表明，通过音频帧的并行编码和解码器的优化，从高延迟模式解码结束时的额外延迟可以非常小。

# 6.    参考文献

[1] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673–2681, 1997.

[2] Alex Graves, Santiago Fern´andez, and J¨urgen Schmidhuber, “Bidirectional lstm networks for improved phoneme classification and recognition,” in Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005, Włodzisław Duch, Janusz Kacprzyk, Erkki Oja, and Sławomir Zadro˙zny, Eds., Berlin, Heidelberg, 2005, pp. 799–804, Springer Berlin Heidelberg.

[3] H. Sak, A. Senior, K. Rao, O. ˙Irsoy, A. Graves, F. Beaufays, and J. Schalkwyk, “Learning acoustic frame labeling for speech recognition with recurrent neural networks,” in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 4280–4284.

[4] Qian Zhang, Han Lu, Hasim Sak, Anshuman Tripathi, Erik McDermott, Stephen Koo, and Shankar Kumar, “Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss,” 2020.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in Advances in neural information processing systems, 2017, pp. 5998–6008.

[6] Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov, “Transformer-xl: Attentive language models beyond a fixed-length context,” in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, p. 2978–2988.

[7] Alex Graves, “Sequence transduction with recurrent neural networks,” in Proceedings of the 29th International Conference on Machine Learning, 2012.

[8] A. Senior, H. Sak, F. de Chaumont Quitry, T. Sainath, and K. Rao, “Acoustic modelling with cd-ctc-smbr lstm rnns,” in 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2015, pp. 604–609.

[9] M. Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, “Rnn-transducer with stateless prediction network,” in ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 7049– 7053.

[10] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” arXiv preprint arXiv:1904.08779, 2019.

[11] Jouppi et al, “In-datacenter performance analysis of a tensor processing unit,” in Proceedings of the 44th Annual International Symposium on Computer Architecture, New York, NY, USA, 2017, ISCA ’17, p. 1–12, Association for Computing Machinery.