快手GRU-InterSpeech2018

机器之心原创

作者：思源

随着短视频的兴起，如何使用算法理解视频内容，并对其进行描述与检索就显得非常重要。最近快手多媒体内容理解部的语音组提出了一种能使用下文信息的门控循环单元，该模型能为快手大量的短视频提供语音识别、语音特效和语音评论等优秀的应用。快手提出的该论文已经被 Interspeech 2018 接收为 oral 论文，目前它同样也部署在了快手的各种语音业务中。

本文介绍了快手这一研究成果以及它在实际业务中的应用，同时也介绍了 Interspeech 2018 中比较有意思的主题。本文首先会讨论语音在快手业务中的应用，以及为什么需要高性能门控循环单元以及较低的解码延迟。随后文章会重点讨论快手如何选择 GRU、mGRU 以及更加精简的循环单元 mGRUIP，同时会介绍如何将下文信息嵌入循环单元以处理语音的协同发音问题，这些带下文信息的高效模块在处理快手短视频语音信息中处于核心地位。最后，本文还会介绍快手整个多媒体理解部门所研究的方向与情况。

快手多媒体内容理解部语音组的李杰博士在 Interspeech2018 做 oral 报告。

为什么语音需要新单元

首先语音在快手业务中的应用主要分为两大类。第一类是语音内容分析，主要目的是对每天快手用户产生的海量语音数据进行内容分析，为接下来的信息安全、内容理解、广告与推荐等提供基础服务。涉及到的技术主要包括：语音识别、关键词识别、说话人识别、声学事件检测等。这类业务快手用户可能不太容易感受的到，但对快手而言是很重要的业务。具体的应用，比如，短视频语音识别、短视频音频标签、直播语音识别、直播脏词过滤等。

第二类是语音交互。其目的是提升用户与快手产品交互时的便利性，此外，可以通过语音设计一些新的玩法，提升趣味性。涉及的技术包括语音识别、关键词唤醒等。比如，快手产品中的魔法表情语音特效触发、语音自动生成字幕、语音评论、语音搜索等。

在语音识别领域，设计一个「又快又好」的声学模型一直是从业者不断追求的目标。「快」指的是模型延迟要小，计算要高效。「好」指的是识别准确率要高。本次快手提出的「具备下文语境的门控循环单元声学模型」就具有这样的特点。在语音内容分析和语音交互两类业务中，语音识别相关部分都可以用此模型。

- **论文：Gated Recurrent Unit Based Acoustic Modeling with Future Context**
- 论文地址：https://arxiv.org/abs/1805.07024

带下文语境的门控循环单元

正因为快手需要快速与准确地处理语音信息，所以快手的李杰博士等研究者提出了一种能利用下文信息的门控循环单元。这里需要注意的是，利用下文信息在语音识别和关键词识别等任务中非常重要。正如快手所述，很多时候语音识别不能仅考虑当前话语的信息，我们还需要一定长度的后文信息才能降低口音和连读等协同发音的影响。

为了利用下文信息，我们首先想到的可能就是 BiLSTM，它广泛应用于机器翻译和其它需要下文信息的序列任务中。但是在语音识别中，双向 LSTM 的延迟非常大，它也做不到实时解

码。例如在使用 BiLSTM 实现语音建模的过程中，模型的延迟是整句话，也就是说在识别第 5 个词时，我们需要等整句话结束并将信息由句末传递到第 5 个词，这样结合前向信息与反向信息才能完成第 5 个词的识别。这种延迟是非常大的，通常也是不可忍受的，没有人希望模型在整句话都说完才开始计算。

整个延迟的控制在语音识别中都处于核心地位，因此正式来说，模型延迟指在解码当前帧时，模型需要等待多久才能对当前帧进行预测。而模型等的时间就应该是识别当前帧所需要的未来信息，这个延迟是一定存在的，只要在可接受的范围内就完全没问题。快手多媒体内容理解部语音组李杰博士表示一般最简单的方法就是在输入特征的时候，除了输入当前特征以外，还要把未来的比如说一百毫秒以内的特征都输入进去。因此在真正使用，并解码的当前时刻 T 的时候，我们必须要等待一百毫秒。

其实有很多方法都能在声学建模中利用下文信息，例如时延神经网络（TDNN）和控制延迟的 LC-BiLSTM 网络等。其中 TDNN 是一种前馈神经网络架构，它可以在上下文执行时间卷积而高效地建模长期依赖性关系。而 LC-BiLSTM 尝试控制解码延迟，希望不再需要等整个句子完成再解码，但这些模型的延迟仍然非常高，达不到实际的需求。

为了降低延迟并提高计算效率，快手的研究者在该论文中以 GRU 为基础进行了修正并添加了上下文模块。总的而言，他们采用了只包含更新门的最小门控循环单元（mGRU），并进一步添加线性输入映射层以作为「瓶颈层」，从而提出大大提升运算效率的门控循环单元 mGRUIP。使用 mGRUIP 再加上能建模下文信息的模块，就能得到高性能与低模型延迟的声学建模方法。

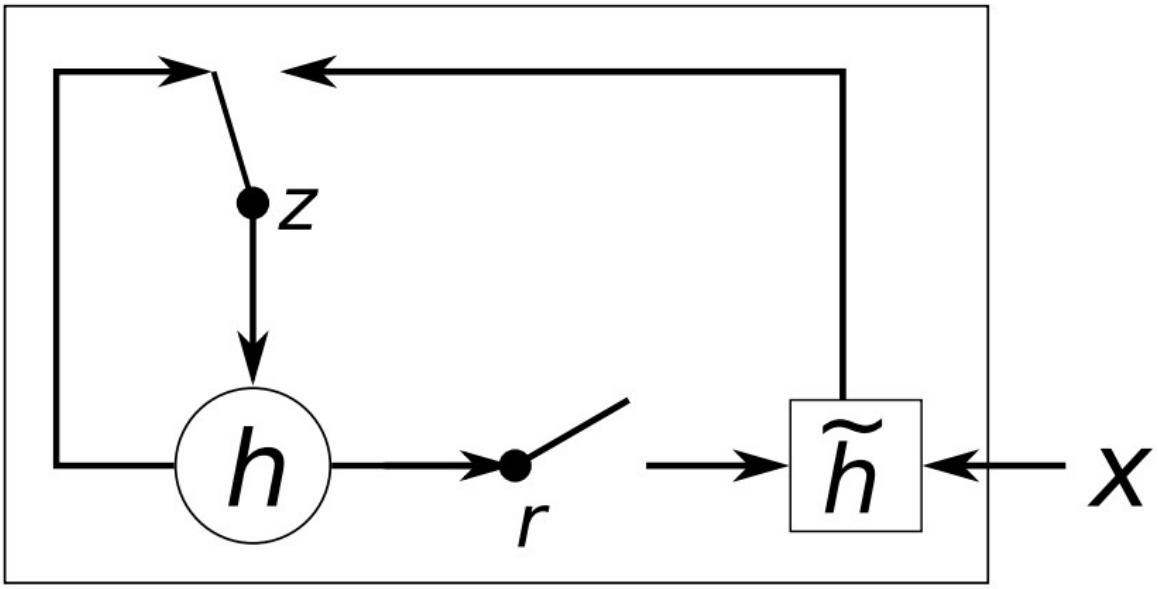
李杰博士表示一般来说，「建模下文信息」总会带来一定的延迟，「建模下文信息」与「低延迟」经常会相互矛盾。这篇论文提出的模型是在两者之间找到了一个比较好的平衡点。模型中的 input projection 形成了一个 bottleneck，而快手在这个 bottleneck 上设计了下文语境建模模块，从而实现了在低延迟的条件下，对下文语境进行有效建模。

从 GRU 到 mGRUIP

为了构建计算效率更高的单元，快手从 GRU、mGRU 到 mGRUIP 探索了新型门控单元。GRU 背后的原理与 LSTM 非常相似，即用门控机制控制输入、记忆等信息而在当前时间步做出预测。GRU 只有两个门，即一个重置门（reset gate）和一个更新门（update gate）。这两个门控机制的特殊之处在于，它们能够保存长期序列中的信息，且不会随时间

而清除或因为与预测不相关而移除。

从直观上来说，重置门决定了如何将新的输入信息与前面的记忆相结合，更新门定义了前面记忆保存到当前时间步的量。在 Kyunghyun Cho 等人第一次提出 GRU 的论文中，他们用下图展示了门控循环单元的结构：



上图的更新 z 将选择隐藏状态 h 是否更新为新的 h tilde，重置门 r 将决定前面的隐藏状态是否需要遗忘。以下图左的方程式展示了 GRU 的具体运算过程：

- **GRU**

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$
$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$
$$\widetilde{h_t} = \tanh(W_h x_t + U_h(h_{t-1} * r_t) + b_h)$$
$$h_t = z_t * h_{t-1} + (1 - z_t) * \widetilde{h_t}$$

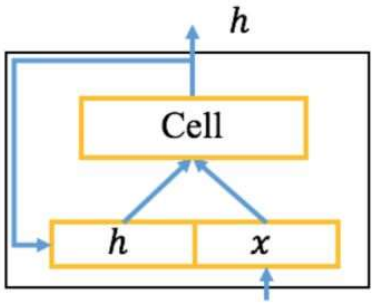
- **Minimal GRU (mGRU)**
  - Removing reset gate
  - Replace tanh with ReLU

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$
$$\widetilde{h_t} = \mathrm{ReLU}(\mathrm{BN}(W_h x_t + U_h h_{t-1}) + b_h)$$
$$h_t = z_t * h_{t-1} + (1 - z_t) * \widetilde{h_t}$$

- Much lower computation cost than GRU and vanilla LSTM

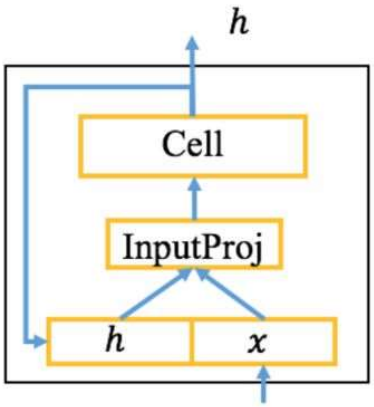
其中 z_t 表示第 t 个时间步的更新门，它会根据当前时间步的信息 X_t 与前一时间步的记忆 h_t-1 计算到底需要保留多少以前的记忆。而 r_t 表示重置门，它同样会通过 Sigmoid 函数判断当前信息与多少以前的记忆能形成新的记忆。而上图右侧所展示的 mGRU 进一步减少了门控的数量，它移除了重置门，并将双曲正切函数换为 ReLU 激活函数。此外，mGRU 相当于令 GRU 中的重置门恒等于 1。

通过上图的左右对比，很明显我们会发现 mGRU 的计算要简单地多，但是如果网络的每一层神经元都非常多，那么 mGRU 的计算量还是非常大，且随着神经元数量的增加计算成线性增长。这就限制了 mGRU 在大型网络和大规模场景中的应用。因此李杰等研究者进一步提出了带输入映射的 mGRUIP，它相当于给输入增加了一个瓶颈层，先将高维特征压缩为低维，然后在低维特征上发生实际的运算，再恢复到应有的高维特征。



1. mGRU

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$
$$\widetilde{h_t} = \mathrm{ReLU}(\mathrm{BN}(W_h x_t + U_h h_{t-1}) + b_h)$$
$$h_t = z_t * h_{t-1} + (1 - z_t) * \widetilde{h_t}$$



2. mGRUIP

$$v_t = W_v[x_t; h_{t-1}]$$
$$z_t = \sigma(W_z v_t + b_z)$$
$$\widetilde{h_t} = \mathrm{ReLU}(\mathrm{BN}(W_h v_t) + b_h)$$
$$h_t = z_t * h_{t-1} + (1 - z_t) * \widetilde{h_t}$$

上图展示了 mGRU 到 mGRUIP 的演变，其中 mGRUIP 会先将当前输入 x_t 与前一时间步的记忆（或输出，h_t-1）拼接在一起，然后再通过矩阵 W_v 将拼接的高维特征压缩为低维向量 v_t，这里就相当于瓶颈层。然后通过批归一化 BN 和激活函数 ReLU 计算出当前需要记忆的信息 h_t tilde，再结合以前需要保留的记忆就能给出当前最终的输出。

mGRUIP 显著地减少了 mGRU 的参数量，它们之间的参数量之比即 InputProj 层的单元数比上隐藏层的单元数。例如我们可以将 InputProj 层的单元数（或 v_t 向量的维度）设置为

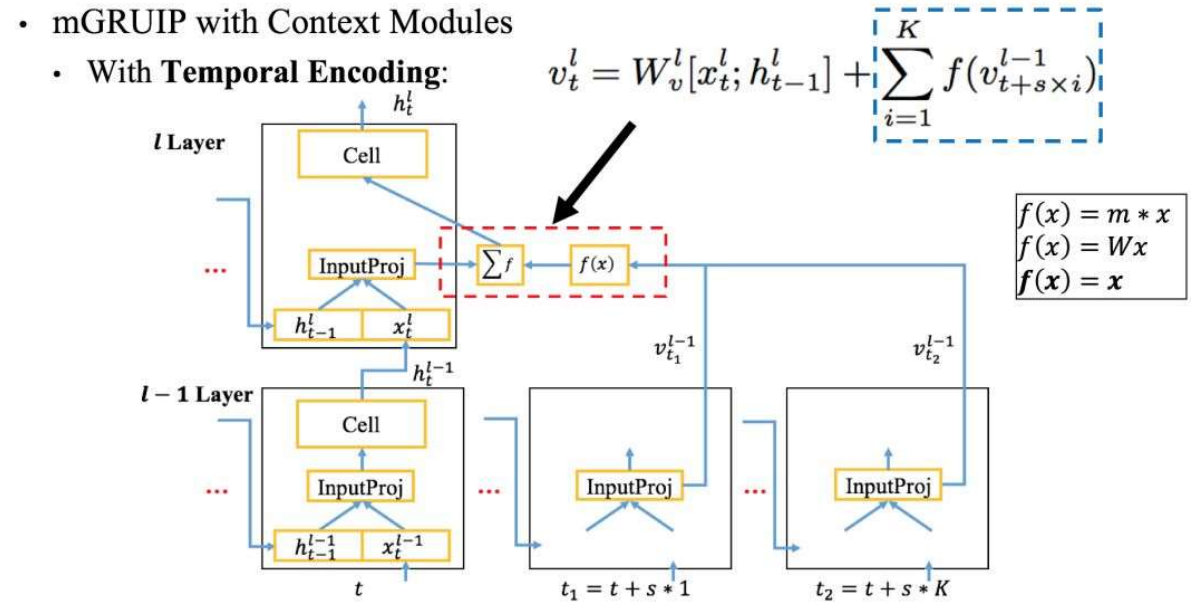
256，而神经网络隐藏层的单元数设置为 2048，那么同样一层循环单元，mGRUIP 比 mGRU 的参数量少了 8 倍。

很多读者可能会疑惑既然等大小的两层网络参数量相差这么多，那么它们之间的表征能力是不是也有差别，mGRUIP 是不是在性能上会有损失。李杰表示他们经过实验发现，这种降维不仅不会降低 GRU 模型的表达能力，反而可以提升模型的性能。不仅本文的 GRU 如此，其他人所做的关于 LSTM 的工作也有类似的发现。在 LSTM 中增加线性输出层，或者输入层，大部分情况下，不仅没有性能损失，反而有一定的收益。可能的原因在于，语音连续帧之间具有较多的冗余信息，这种线性层可以进行一定程度的压缩，降低冗余。

mGRUIP 与上下文模块

完成高效的门控循环单元后，接下来我们需要基于这种单元构建利用下文信息的方法。在快手的论文中，他们提出了两种上下文模块，即时间编码与时间卷积。

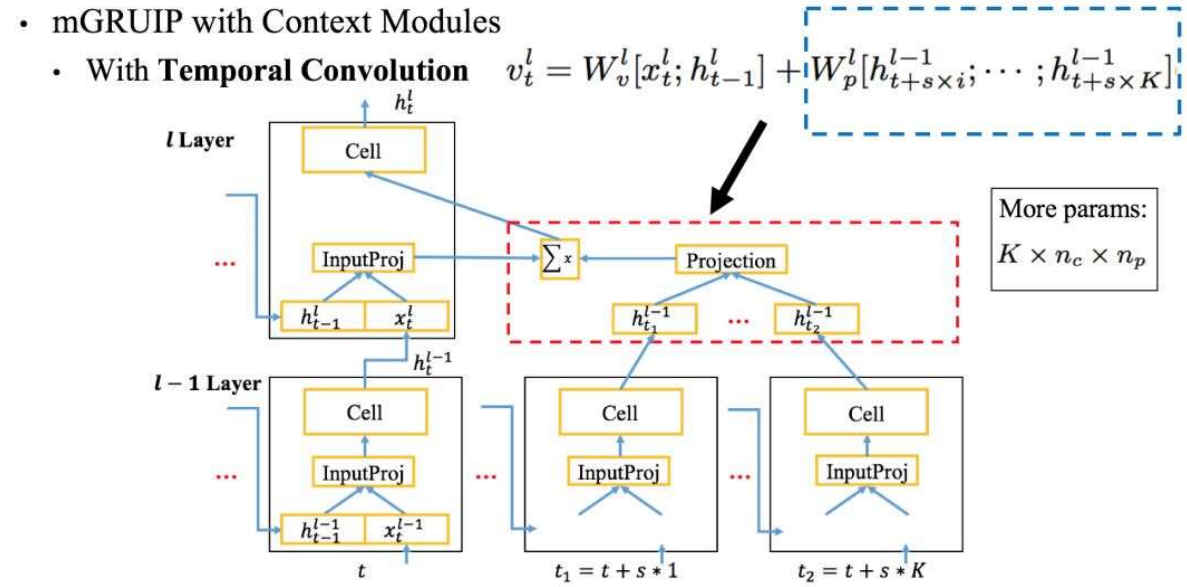
在时间编码中，未来帧的语境信息会编码为定长的表征并添加到输入映射层中。如下向量 v 的表达式为添加了时间编码的输入映射层，其中蓝色虚线框表示为时间编码，且 l 表示层级、K 表示利用未来语境的数量、s 为未来每一个语境移动到下一个语境的步幅。在向量 v 的表达式中，左侧 W_v[x_t; h_t-1] 为 mGRUIP 计算输入映射层的表达式，而右侧时间编码则表示将前一层涉及下文信息的 InputProj 加和在一起，并与当前层的 InputProj 相加而作为最终的瓶颈层输出。这样就相当于在当前时间步上利用了未来几个时间步的信息，有利于更准确地识别协同发音。



上图展示了带有时间编码的 mGRUIP 计算过程，在 l 层时先利用当前输入与上一层输出计算出不带下文信息的 InputProj，然后从 l-1 层取当前时间步往后的几个时间框，并将它们的 InputProj 向量加和在一起。将该加和向量与当前层的 InputProj 向量相加就能得出带有下文信息的瓶颈层向量，它可以进一步完成 mGRUIP 其它的运算。如上所示转换函数 f(x) 一般可以是数乘、矩阵乘法或者是恒等函数，但快手在实验中发现恒等函数在性能上要更好一些，所以它们选择了 f(x)=x。

李杰等研究者还采用了第二种方法为 mGRUIP 引入下文信息，即时间卷积。前面时间编码会使用低层级的输入映射向量表征下文信息，而时间卷积会从低层级的输出状态向量中抽取下文信息，并通过输入映射压缩下文信息的维度。如下 v 向量的计算式为整个模块的计算过程，其中左侧同样为 mGRUIP 计算 InputProj 的标准方法，右侧蓝色虚线框表示时间卷积。

简单而言，时间卷积即将所需要的前层输出拼接在一起，并通过 W_p 构建表征下文信息的输入映射层。其中所需要的前层输出表示模型需要等多少帧语音信息，例如需要等 10 帧，那么前一层当前往后 10 个时间步的输出会拼接在一起。此外，这两种方式的延迟都是逐层叠加的，也就是说每一层需要等 10 毫秒，那么 5 层就需要等 50 毫秒。



如上所示为带时间卷积的 mGRUIP 具体过程，它会利用 l-1 层的 t_1 和 t_2 等时间步输出的隐藏单元状态，并在第 l 层拼接在一起。然后将下文信息压缩为 Projection 向量并与 l 层当前时间步的 InputProj 相加而成为带下文信息的瓶颈层向量。

至此，整个模型就完成了构建，快手在两个语音识别任务上测试了该模型，即 309 小时的 Swichboard 电话语音任务和 1400 小时的国内普通话语音输入任务。mGRUIP 在参数量上显著地小于 LSTM 与 mGRU，且在词错率和性能上比它们更优秀。此外，带有上下文模块的 mGRUIP 在延迟控制和模型性能上都有非常优秀的表现，感兴趣的读者可查看原论文。

Interspeech 2018 与快手研究

这篇论文也被语音顶会 Interspeech 2018 接收为 Oral 论文，李杰同样在大会上对这种能使用下文信息的门控循环单元给出了详细的介绍。前面我们已经了解了该模型的主要思想与过程，但是在 Interspeech 2018 还有非常多优秀的研究与趋势。李杰表示：「从今年的大会看，主流的声学模型依然是基于 RNN 结构，只不过大家所做的工作、所解的问题会更加细致。比如，对于 RNN 模型低延迟条件下，下文语境建模问题，除了我们在关注，Yoshua Bengio 他们也有一篇工作聚焦在该问题上。此外，如何提升 RNN 声学模型的噪声鲁棒性、低资源多语言声学模型建模、说话人和领域声学模型自适应、新的 RNN 结构等问题，也受到了很多关注。」

除此之外，李杰表示端到端模型依然是大家研究的热点。主要的技术方向有三个，第一，CTC；第二，基于 RNN 的带注意力机制的编解码模型；第三，也是今年 Interspeech 新出现的，基于 self-attention 的无 RNN 结构的编解码模型。

其实除了 Interspeech 接收的这篇 Oral 论文，快手还有很多不同方向的研究，包括计算机视觉、自然语言处理和情感计算等等。因为快手平台每天都有大量的短视频上传，因此如何分层有序地提取视频信息、理解视频内容就显得尤为重要。针对该问题，快手多媒体内容理解部门通过感知和推理两个阶段来解读一个视频，首先感知获取视频的客观内容信息，进而推理获取视频的高层语义信息。

在感知阶段，除了上文所述的语音处理，快手还会从另外三个维度来分析理解视频内容，包括人脸、图像和音乐。

- 对于语音信息，快手不仅进行语音识别，还需要实现说话人识别、情绪年龄等语音属性信息分析。
- 对于人脸信息，快手会对视频中的人脸进行检测、跟踪、识别，并分析其年龄、性别、3D 形状和表情等信息。
- 对于图像信息，快手会通过分类、物体检测等算法分析场景、物体，通过图像质量分析

算法对图像的主观质量进行评估，通过 OCR 分析图像中包含的文字信息等。

- 对于音乐信息，快手需要进行音乐识别、歌声/伴奏分离、歌声美化打分等分析，对音乐信息进行结构化。

从以上四个方面，快手能抽取足够的视频语义信息，并为推理阶段提供信息基础。推理阶段可以将视频看做一个整体，进行分类、描述、检索。此外，高级视频信息也可以整理并存储到快手知识图谱中，这样融合感知内容和知识图谱，就可以完成对视频高层语义及情感的识别。因此，感知与推理，基本上也就是快手多媒体理解部门最为关注的两大方面。