AudioLLM - 李沐亲自解说语音大模型训练的底层思路

2025年7月29日 1 分钟阅读

#AI #AudioLLM #李沐 #语音大模型 #训练

> *李沐亲自解说语音大模型AudioLLM训练的底层思路*

引言

今天在Youtube上看了李沐亲自向我们介绍AudioLLM训练底层思维逻辑，非常精彩，我将其中的核心技术细节提炼出来，按照“提出问题 -> 寻找思路 -> 找到的解决方案”的结构进行细致的总结, 力求让你能清晰地理解这个模型是如何炼成的的第一性原理。

AudioLLM训练的底层思路: 旨在将一个强大的预训练文本大语言模型的通用智能，拓展至语音领域，构建一个统一的多模态系统。它首先通过一个特制的语音编码器（Tokenizer），将连续的音频信号转换为离散的数字序列。此过程的关键在于优先保留语音中的“语义”信息而非纯粹的声学细节，从而建立文本与语音在概念层面的深刻链接。为解决海量数据标注难题，它采用一种巧妙的自监督训练框架：一个“生成模型”学习将文本和场景描述转化为语音，同时一个“理解模型”学习反向任务。两个模型互为数据源和监督者，在海量无标注数据上协同进化，最终使统一的大模型不仅能听会说，更能深刻理解和执行复杂的、包含丰富上下文的语音指令。

核心问题一：如何让一个精通文本的“学霸”学会“听说”？

> **提出问题：** 我们已经有了非常智能的文本大语言模型，它们能读会写，智商很高。那么，我们能不能让它在不“变笨”（即文本能力不下降）的前提下，额外掌握听和说的能力？如果直接在训练文本模型时，加入千万小时级别的海量语音数据，会发生什么？
>
> **如何寻找思路：**

目录

文章信息

字数

阅读时间

发布时间

更新时间

标签

#AI #AudioLLM #李沐 #训练

思路1：训练独立的语音模型。 这是传统做法，比如专门的语音识别（ASR）模型或语音合成（TTS）模型。但这种方式的缺点是模型功能单一，难以理解复杂的、带有丰富上下文和情感设定的指令。例如，它很难理解“请用一个急躁的年轻男性的声音，在吵架的场景下，说出这句话”这类复杂的任务。

思路2：改造现有的大语言模型。 将语音能力整合进文本大语言模型。

潜在好处： 可以利用大语言模型强大的文本理解和推理能力。模型能听懂复杂的指令，完成如“写一首歌并唱出来”、“分析一段录音中的场景、人物和情绪”或进行低延迟的实时语音对话等高级任务。所有这些任务都可以统一成一个“系统指令(System) -> 用户输入(User) -> 模型输出(Assistant)”的框架，用一个“祖传配方”解决所有问题。

潜在风险： 模型会变得更大、更复杂；加入语音数据可能会干扰原有的文本能力，导致“智商”下降。

找到的解决方案： 选择**思路2**，即**构建一个统一的语音文本多模态大语言模型**。接受其可能带来的模型变大等挑战，因为其在处理复杂指令和多任务统一方面的优势是革命性的。目标是通过“大力出奇迹”（Scaling Law），用庞大的数据和算力，让模型在保留高水平文本能力的同时，无缝集成强大的语音理解和生成能力。

核心问题二：机器如何“阅读”和“说出”连续的声音信号？

提出问题： 语言模型处理的是文本，文本是由一个个离散的“字”或“词”（Token）组成的。但语音信号是连续的波形，如何将其转换成模型能理解的离散化Token？

如何寻找思路：

基本思路：离散化。 将连续的语音信号切成极小的片段（例如，每秒切成几十个片段）。

建立“声音词典”。 创建一个包含成千上万个基础声音片段的“词典”（Tokenizer）。

编码/解码。 对于任意一段语音，将其切片，然后用“词典”中与之最相似的声音片段的编号来表示它。这样，一段连续的语音就变成了一个由数字编号组成的序列，模型就可以像处理文字一样处理它了。

遇到的关键抉择： 这种转换是一种极高倍率的压缩（视频中提到比MP3压缩375倍），必然会丢失大量信息。那么，在压缩时，我们应该优先保留什么？是声音的物理特性（如音色、音调），还是声音所传达的**语义信息**（说了什么内容）？

找到的解决方案： 优先保留语义信息。团队发现，声学上的特征（比如说话风格）只需要保留一点点，后续可以想办法还原。但语义信息千变万化，是模型理解的核心。只有优先保证语义的准确性，才能让模型将语音Token和文本Token的"意思"尽早关联起来，从而实现语音和文本之间的流畅切换。因此，**研发一个高质量的、能最大程度保留语义的语音Tokenizer是整个技术的关键基石。**

核心问题三：如何准备"教材"，教会模型理解和生成语音？

提出问题： 我们有了模型架构和声音表示方法，但用什么样的数据、通过什么样的方式来训练，才能让模型真正打通语音和文本的任督二脉呢？

如何寻找思路：

数据来源：

思路A： 使用YouTube、B站等平台的现成视频数据。这些数据质量好，数量大。但问题在于，这些平台通常不允许爬取其数据用于模型训练，存在版权风险。

思路B： 购买商业数据或从允许抓取的公开网页中寻找音频文件。这种方式合规，但数据质量参差不齐，可能需要从1亿小时的原始数据中筛选出1000万小时的可用数据，成本和工作量巨大。

数据标注（如何制作"教材"）：

思路A： 将海量语音数据交给GPT-4或Gemini等先进模型，让它们自动分析并生成详细的标注（如场景描述、人物、情绪、对话文本等）。但问题在于，这些模型的服务条款禁止使用其输出来训练竞争模型，且API调用成本对于上亿级别的数据来说是天文数字。

找到的解决方案：

数据来源： 采用**思路B**，通过购买和抓取公开数据，并进行大量清洗来获取训练数据。

数据标注： 独创了一种"**左右互搏**"的自监督训练方法。 有了"声音的语言"，接下来就是如何教模型学习。如果靠人工去给海量的语音数据打上详细的标签（比如"这是一个性格急躁的男人在嘈杂的咖啡馆里愤怒地说……"），成本高到无法想象。于是，他们设计了一种极其巧妙的训练方法，可以称之为 **"双子模型互搏学习法"。** 他们用**同样的模型架构**，训练了两个"学徒"模型：
a. **学徒A（语音理解模型）**：它的任务是"听"。你给它一段语音，它会努力分析并用文字输出对这段语音的全方位描述，包括： - **内容转录：** 里面的人说了什么话。 - **场景分析：** 这是在

室内还是室外？是安静的房间还是嘈杂的街道？ - **人物画像：** 说话的是男人还是女人？大概什么年纪？情绪是开心、悲伤还是愤怒？

b. **学徒B（语音生成模型）**：它的任务是“说”。你给它一段和学徒A输出格式一样的文字描述（包含场景、人物、情绪和说话内容），它就要根据这些复杂的指令，生成一段符合所有要求的、逼真的语音。 **训练过程就像是两位学徒在切磋武艺：** - **第一步：** 让“理解者”学徒A去听一段原始语音，然后写下它的“听后感”（即详细的文字描述）。 - **第二步：** 把这份“听后感”作为作业，交给“生成者”学徒B，让它照着这份作业去“说”出对应的语音。 - **第三步：** 比较学徒B“说”出的语音和原始语音的差距，然后同时告诉两个学徒：“A，你听得还不够准！B，你说得还不够像！” - **第四步：** 反过来，也可以用生成模型的输出来训练理解模型。

就这样，两位学徒在一个体系内，一个负责输入（听），一个负责输出（说），互相出题，互相纠错，共同进步。通过亿万次的“左右互搏”，最终融合成一个既懂听又会说的强大模型。这个过程完全自动化，摆脱了对人工标注的依赖，真正实现了“大力出奇迹”。

核心问题四：为什么模型的“声音克隆”能力如此强大？

提出问题： 视频中展示了模型能惟妙惟肖地模仿任何人的声音，甚至进行跨语言的声音转换。这种强大的声音克隆（Voice Cloning）能力是如何实现的？

如何寻找思路： 模型的声音克隆能力并非单一技术，而是其整体设计理念的自然体现。

海量数据是基础： 模型见过的声音数据足够多，覆盖了各种各样的音色和说话风格，这是模仿的基础。

上下文学习是关键： 在进行声音克隆时，并非简单地提取音色。其操作方式是，将一小段需要被模仿的声音样本（包含音频和对应的文字）作为“上一轮对话”输入给模型。

找到的解决方案（工作原理）： 模型利用了其强大的上下文理解和延续能力。

抓取综合信息： 当模型接收到声音样本时，它不仅仅分析了音色（声学特征），更重要的是，它利用其语言模型的“智商”去理解了这段样本的**场景信息、语意信息和说话人特点**（比如是在严肃演讲还是在轻松聊天）。

匹配与延续： 当你给出新的文字让它生成时，模型会根据刚刚抓取到的综合信息，去匹配和延续这种风格、情绪和场景。

Prompt Engineering： 这也揭示了如何更好地使用它：为了生成特定场景的语音，你最好提供一个来自相似场景的语音样本作为参考。如果找不到，还可以通过修改系统指令（System Prompt）来描述你想要的声音场景，让模型在你提供的声音和指令之间进行融合，创造出最贴切的输出。这正是这种多模态大语言模型进行“提示工程”（Prompt Engineering）的精髓所在。

参考

Youtube:李沐: 肝了6个月的AudioLLM，开源了【100亿模型计划】

> **李飞飞博士的生平与...**
>
> 李飞飞博士的生平与洞见