arXiv:2301.05295v2 [eess.AS] 1 Feb 2023

通过自然语言处理生成摇滚吉他六线谱

Josue Casco-Rodriguez莱
斯大学，美国得克萨斯州休
斯敦jc135@rice.edu

摘要

深度学习近年来推动了图像和文本生成式建模的发展，并使其更加普及 [1, 2], ，与此同时，另有研究正在探索生成更复杂数据形式（如音频）的可能性 [3, 4]。然而，高维度、长程依赖关系以及缺乏标准化数据集，使得音频和音乐的生成式建模目前极具挑战性。我们提出将音乐建模为一系列离散音符，并在此基础上运用自回归自然语言处理技术，实现成功的生成式建模。以往的研究曾在乐谱和 MIDI 等数据上采用类似流程 [5, 6], ；我们旨在将此类方法拓展至研究尚不充分的吉他六线谱领域。具体而言，据我们所知，我们首次将一种特定流派——重摇滚——建模为吉他六线谱。与其他吉他六线谱生成研究不同，我们提供了一个可免费使用的公开演示：https://huggingface.co/spaces/josuelmet/Metal MusicInterpolator。[1]

1 引言

音乐与图像和语言一样，是一种基本的艺术形式，也是人类经验的典型组成部分。尽管文本到图像模型 [7, 1, 8, 9, 10] 在视觉艺术生成与建模方面取得了突破性进展，但音乐创作领域尚未实现类似突破；不过，近期的一些研究，例如 OpenAI 的 Jukebox [4], ，已在高级音乐生成方面取得进展。此类突破迟迟未能出现，很大程度上是因为音乐难以建模，需要对数据进行序列建模，而这些数据媒介不如图像或文字那样广为人知或直观。

探究机器如何学习理解或生成音乐，可以从了解人们如何学习处理音乐入手。虽然音乐以连续时间音频信号的形式存在，但人们最有效地将音乐理解和分析为由离散频率构成的模式（例如音符“A” = 440 Hz），这些音符按离散量化的时间间隔演奏（例如四分音符、二分音符和全音符）。因此，只要有足够的、以离散形式表示的音乐样本数据集，就可以利用用于理解离散数据序列的序列建模技术来进行音乐建模。

虽然乐谱和乐器数字接口（MIDI）文件是音乐离散压缩表示的传统形式，但一种研究较少的突出音乐表示形式是吉他六线谱（见图 1 和图 2）。吉他六线谱看起来与传统乐谱相似，但它通过品格和琴弦索引来表示音符，演奏者必须将手指放在相应的位置才能奏出特定音符。弦乐器的独特之处在于，大多数可演奏的音符都可以通过不止一种品格与琴弦组合奏出。

我们构建了一个新数据集，其中包含来自某一特定音乐流派（重摇滚）的吉他六线谱压缩表示，

[1]用于训练最终演示模型的源代码可在此处获取：https://github.com/Josuelmet/Metal-Music-Interpolator。

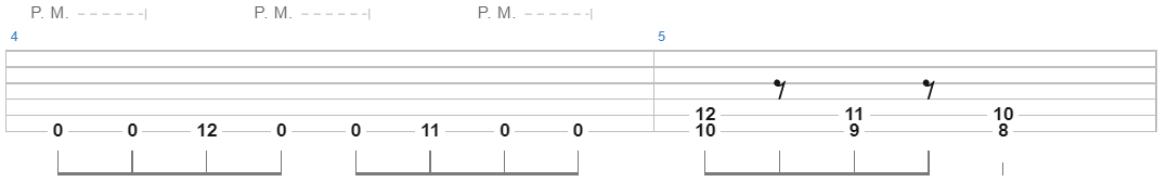


图 1：一段以 4/4 拍写成的吉他六线谱，包含两个小节。每个音符表示的不是其音高，而是演奏者应在某根琴弦上按下的具体品位编号，以奏出该音符。某音符在其所在琴弦上的品位编号，等于该音符的音高与该琴弦可发出的最低音高之间的半音数。请注意，某些音符还包含有关其力度的信息：例如，掌根闷音符号表示某些音符应以半闷音的方式演奏。

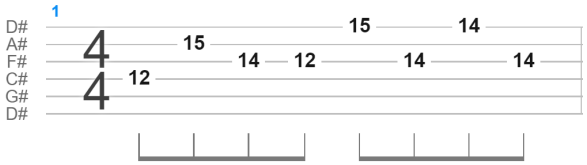


图 2：另一段以 4/4 拍写成的吉他六线谱，这次是枪与玫瑰的*Sweet Child O' Mine* 的标志性首小节。左侧纵向排列的六个音高，分别是六根吉他弦可发出的最低音高。

并提出一种神经网络架构，可利用序列建模（例如长短期记忆网络或自然语言处理模型中的序列建模）生成新的吉他六线谱序列，生成时以现有吉他六线谱中的一段简短片段为条件。具体而言，所提出的自回归模型旨在根据前面的词元$\mathbf{x}^{N+1}$ 估计最有可能出现的新六线谱词元$\mathbf{x}_1, \mathbf{x}_2$，……，$\mathbf{x}_N$（即估计条件概率 p（$\mathbf{x}_{N+1}|\mathbf{x}_1$，……，$\mathbf{x}_N$）），从而实现迭代生成：根据一个$N$词元序列生成一个 M词元序列，适用于 $M > N$。

2 背景

2.1 序列模型

循环网络。序列建模是机器学习和统计学中一个由来已久的问题，其早期的重要尝试之一是循环神经网络 [11]。循环神经网络能够利用某种记忆机制，对理论上无界的上下文中的序列进行建模，但在实际应用中，它们及其近期变体 [12] 仍难以做到这一点，部分原因在于梯度传播问题 [13]。

Transformer。与此同时，Transformer [14] 通过用一种前馈注意力操作取代循环操作，规避了循环神经网络的问题；该操作会将序列中的每个元素与其他所有元素进行比较。由于注意力机制本身不具备记忆能力，且其对长度为 N [16] 的序列具有有限的上下文范围 [15], 和 $O(N^2)$ 运行时间，这种方法乍看之下似乎处于劣势。然而，结合位置嵌入以及通过向量化实现的词元降维等创新后，变换器架构在序列建模和图像建模方面取得了巨大进展[2, 1, 17]。

自注意力。 任何 Transformer 架构的核心都是自注意力机制。给定一个向量值输入序列 $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^{N \times D_x}$，其中每个元素的维度均为 D_x，且 Transformer 前馈维度为 D，自注意力头通过以下方式将 $\mathbf{X}$ 转换为输出序列 $\hat{}\mathbf{V}$ ： [16]

1. 使用权重 $\mathbf{W}_Q$,$\mathbf{W}_K \in \mathbb{R}^{D\times D_x}$ 和 $\mathbf{W}_V \in \mathbb{R}^{D_v\times D_x}$，通过以下线性变换将 $\mathbf{X}$ 投影为三个不同的矩阵——查询矩阵、键矩阵和值矩阵 $\mathbf{Q}$,$\mathbf{K}$, 和 $\mathbf{V}$:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q^T$$
$$\mathbf{K} = \mathbf{X}\mathbf{W}_K^T$$
$$\mathbf{V} = \mathbf{X}\mathbf{W}_V^T$$

2. 将查询、键和值矩阵表示为$\mathbf{Q} = [\mathbf{q}_1, ..., \mathbf{q}_N]^T, \mathbf{K} = [\mathbf{k}_1, ..., \mathbf{k}_N]^T$, 以及$\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_N]^T$，其中，向量$\mathbf{q}_i, \mathbf{k}_i, \mathbf{v}_i$对应于 $i \in \{1, 2, ..., N\}$，分别是查询向量、键向量和值向量。

每个输出序列向量$\hat{}\mathbf{v}_i$ 的计算方法是：将每个值向量$\mathbf{v}_j$ 乘以一个分数，该分数由查询向量$\mathbf{q}_i$ 与键向量$\mathbf{k}_j$ 之间的相似度确定:

$$\hat{\mathbf{v}}_i = \sum_{j=1}^{N} \text{softmax}\left(\frac{\mathbf{q}_i^T \mathbf{k}_j}{\sqrt{D}}\right)\mathbf{v}_j$$

因此，计算$\hat{}\mathbf{V} = [\hat{}\mathbf{v}_1, ..., \hat{}\mathbf{v}_N]^T$可以简单地表示为:

$$\hat{\mathbf{V}} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D}}\right)\mathbf{V} = \mathbf{A}\mathbf{V},$$

其中，注意力矩阵 $\mathbf{A}$ 通过对矩阵的每一行应用 softmax 操作计算得出 $\mathbf{Q}\mathbf{K}^T/\sqrt{D}$ [16]。

2.2 相关工作

音乐/音频生成。 我们对音乐序列建模的核心贡献，是公开提供重摇滚吉他六线谱建模。此前的多项研究以及正在进行的工作，均探索了连续和离散数据模态下的音乐生成。例如，近期的 SaShiMi [3] 和 Jukebox [4] 架构分别在连续波形和离散音符领域进行音频和音乐生成。扩散模型 [18] 的出现也催生了一种结合频谱图与 MIDI 音乐生成的新模型 [19]。

吉他谱文献。 吉他谱分析领域规模不大，但正在不断发展，已有多项研究应对诸如基于图的独奏分析 [20]，转录 [21]，数据集收集和序列建模 [22, 23, 24] 等挑战。AnimeTab[22] 和 DadaGP [23]，对我们的工作尤为重要，因为它们也选择采用基于 Transformer 的方法，以统计方式生成吉他六线谱序列。与 DadaGP 不同，我们的模型采用的词元表示简单得多，也更易于理解；它是在一个特定流派的数据上训练的，并且提供了公开可用的演示。我们的模型或许与 AnimeTab 有些相似；AnimeTab 在本研究开发期间发表，预计很快会发布演示。不过，我们的模型已经提供演示，并且训练所用的流派是重摇滚，而非动漫/电子游戏音乐。

3 方法

3.1 数据处理

初始预处理。 统计推断方法的成功往往取决于用于训练的数据质量——数据预处理对模型成功的重要性，与模型本身不相上下。我们的数据预处理流程首先收集大量吉他六线谱格式的歌曲，确保它们能够准确代表一种音乐子流派[2]。对于每个六线谱文件，先将其中的每首歌曲转换为 4/4 拍，以便于处理；随后通过 PyGuitarPro[3] 将其转换为 Python 对象，以便查询。接着，将每首歌曲中的每条音轨（即每种乐器或人声，不包括

[2]完整歌曲列表： https://github.com/Josuelmet/Metal-Music-Interpolator/blob/main/songs/README.md [3] https://github.com/Perlence/PyGuitarPro

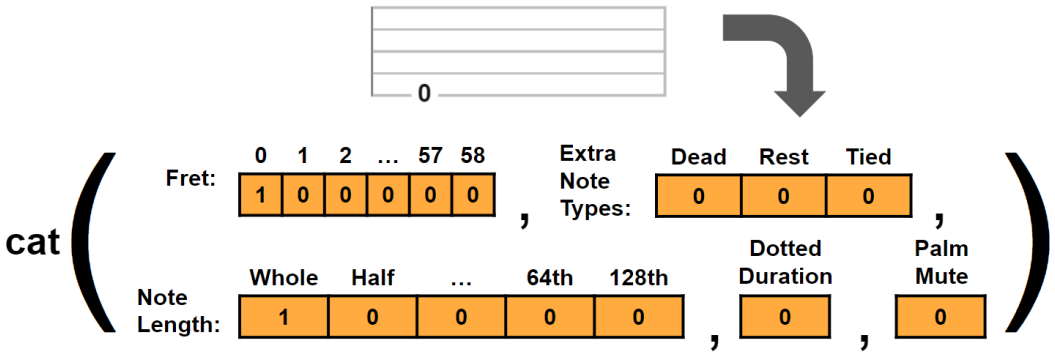


图3：以吉他/贝斯最低音弦0品上的全音符为例，说明音符嵌入方案。品格值以独热编码表示为0，音符时值以独热编码表示为全音符；由于该音符既不是附点音符，也不是闷音音符、闷音止音符或休止符，因此所有标志位均未设为1。

鼓）转换为一维列表，其中包含歌曲中的每个音符；每个音符均以一个元组表示，包含音符的音高（并用特殊标记表示延音、闷音[4], 和休止音符）、时值、适用时的和弦属性（所表示的和弦包括四度、减五度和纯五和弦），以及两个标记，用于指示该音符是否附点以及是否被闷音。请注意，每个音符的音高并非以其音乐音高（例如“A4”或“C3”）表示，而是以吉他（或贝斯）上演奏者应按下的品格表示，从而奏出该音符。将所有歌曲的音符表示为元组后，再通过可逆字典映射将每个元组转换为整数。

嵌入初始化。经过初步预处理后，每首歌曲都表示为一组序列，其中每个序列代表一种人声或乐器，并包含用于表示各个音符的整数。虽然朴素的序列模型可以尝试对这些标量序列进行推断，但现代序列模型成功地采用了另一种方法：将序列中的各个词元或元素表示为向量，从而以更具表现力和信息量的方式表示词元模态。与以往的研究不同，[23], 我们选择了简单而有效的初始词元向量化方法，如图3所示。每个经过初始向量化的词元嵌入在训练前有72个维度：前59个维度用于对音高值与相应乐器可演奏的最低音高之间相差的半音数进行独热编码（这相当于吉他或贝斯上的品格数）；接下来的3个维度是标志位，分别表示音符是否为闷音止音符、休止符或连音符；再接下来的8个维度对音符时值进行独热编码（例如全音符、二分音符和四分音符）；最后两个维度是标志位，分别表示音符时值是否带附点，以及音符是否以闷音方式演奏。词元嵌入在训练过程中会经过训练和迭代优化，但我们发现，手工设计的初始化方案优于默认的随机词元初始化。与其他成功的Transformer架构一样，每个词元还带有位置嵌入，并与向量化的词元嵌入一同输入Transformer模型。

3.2 模型架构

除了以 PyGuitarPro 为唯一切入点，将吉他谱文件查询为 Python 对象，并据此实现完整的数据预处理流程外，我们还必须手动实现 Transformer 架构。由于与训练传统语言模型所使用的序列数量和参数数量相比，我们手头的数据集相对较小，因此选择了 mini-GPT 模型 [2]。由于我们没有使用现成的 Transformer 模型，还必须实现因果掩码机制，以确保 Transformer 无法使用其试图预测的词元之后的任何词元信息。在采用 90-10 的测试集-验证集划分进行大量超参数调优后，最终的超参数值列于表 1。最终的 mini-GPT 架构由一个嵌入层（如前所述）、依次排列的三个 Transformer 块组成，

[4]闷音是指被大幅制音、以至于失去明确音高感的音符。

超参数	值
N（序列长度）	100
输出维度（唯一词元数量）	1629
初始嵌入维度	72
Transformer 块数量	3
Transformer 前馈维度	512
Transformer dropout 率	30%
每个 Transformer 的注意力头数量	10
初始学习率	0.003
β_1 (Adam 参数)	0.96
批量大小	512
训练轮数（由早停法确定）	~ 120

表1：最终 mini-GPT 模型的超参数。

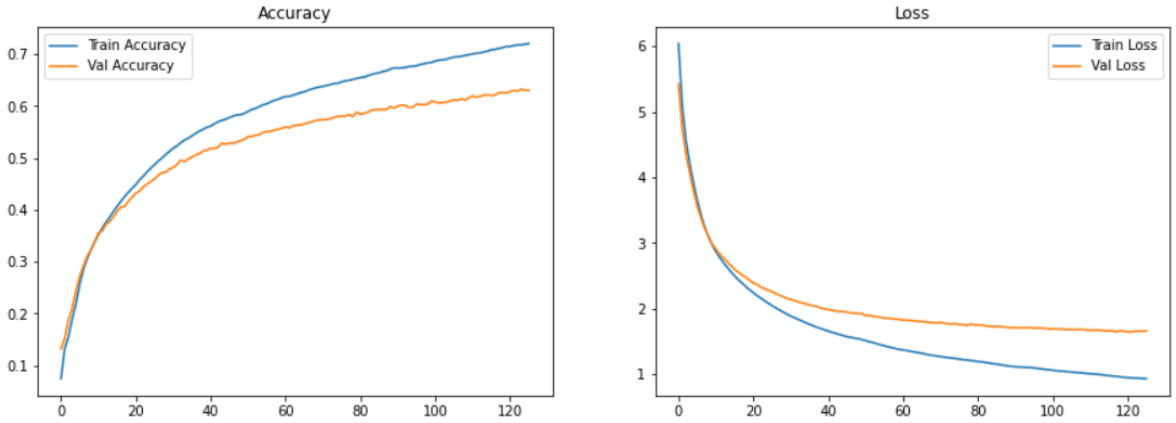


图4：最终 mini-GPT 模型在 90-10 训练集-验证集划分下的训练和验证准确率与损失。

以及最后一个前馈层，该层会返回一个 1629 维向量，用以表示下一个词元值的条件概率 $p(\mathbf{x}_{N+1}|\mathbf{x}_1, \ldots, \mathbf{x}_N)$。

4 结果

性能。 使用表格中列出的超参数 1 ，并根据验证损失进行早停后，我们的 mini-GPT 模型取得了超过 70% 的训练准确率和超过 60% 的验证准确率，如图所示。 4。虽然降低 Transformer 丢弃率可以提高训练准确率，但通过实证我们发现，这样做会因过拟合而损害验证集上的泛化能力。借助所提供的交互式演示，我们可以对模型进行定性评估[5]。据我们所知，我们的工作不仅是首个提供公开访问生成演示的工作，而且模型的成功也进一步验证了自然语言处理技术在音频和音乐生成领域的应用价值及其必要性。

未来的工作。 对我们的模型进行直接且有趣的扩展，包括在嵌入中更明确地纳入节奏信息 [24] ，或将其作为单独的嵌入；这种方法将更有助于捕捉和理解落在强拍、弱起拍和后拍上的音符之间的差异。另一种扩展是同时对多个音轨或声部进行建模，从而能够建模丰富的和弦或鼓点序列。对于基于音符的和弦或鼓点建模，某种专门的平移不变性或许是关键，以确保自回归模型能够理解：多个音符同时出现时，它们在序列中的顺序并不重要。采用更新颖的注意力机制 [16] 或许还能进一步提升性能。扩散

[5] https://huggingface.co/spaces/josuelmet/Metal Music Interpolator

模型也可以用于生成吉他六线谱序列，但它们能否胜过自回归语言模型尚未得到证明；近期研究表明，Transformer 与扩散模型可以结合起来，在音乐生成方面取得最先进的成果 [19]。

参考文献

[1] A. Ramesh、P. Dhariwal、A. Nichol、C. Chu 和 M. Chen，《利用 CLIP 潜变量进行层次化文本条件图像生成》，*arXiv* 预印本 *arXiv:2204.06125*，2022年。[2] T. Brown、B. Mann、N. Ryder、M. Subbiah、J. D. Kaplan、P. Dhariwal、A. Neelakantan、P. Shyam、G. Sastry、A. Askell、S. Agarwal、A. Herbert-Voss、G. Krueger、T. Henighan、R. Child、A. Ramesh、D. Ziegler、J. Wu、C. Winter、C. Hesse、M. Chen、E. Sigler、M. Litwin、S. Gray、B. Chess、J. Clark、C. Berner、S. McCandlish、A. Radford、I. Sutskever 和 D. Amodei，《语言模型是少样本学习者》，载于神经信息处理系统进展，第33卷，2020年。[3] K. Goel、A. Gu、C. Donahue 和 C. Re，《这是原始音频！利用状态空间模型生成音频》，载于第*39*届国际机器学习会议论文集，第162卷机器学习研究论文集，2022年。[4] P. Dhariwal、H. Jun、C. Payne、J. W. Kim、A. Radford 和 I. Sutskever，《Jukebox：一种音乐生成模型》，*arXiv* 预印本 *arXiv:2005.00341*，2020年。[5] C. Payne，《MuseNet》，*OpenAI*博客，2019年。[6] G. Mittal、J. Engel、C. Hawthorne 和 I. Simon，《利用扩散模型进行符号音乐生成》，载于国际音乐信息检索学会会议，第22卷，2021年。[7] R. Rombach、A. Blattmann、D. Lorenz、P. Esser 和 B. Ommer，《利用潜扩散模型进行高分辨率图像合成》，载于*IEEE/CVF*计算机视觉与模式识别会议（*CVPR*）论文集，2022年。[8] H. Chang、H. Zhang、J. Barber、A. Maschinot、J. Lezama、L. Jiang、M.-H. Yang、K. Murphy、W. T. Freeman、M. Rubinstein、Y. Li 和 D. Krishnan，《Muse：通过掩码生成式Transformer实现文本到图像生成》，*arXiv* 预印本 *arXiv:2301.00704*，2023年。[9]N. Ruiz、Y. Li、V. Jampani、Y. Pritch、M. Rubinstein 和 K. Aberman，《Dreambooth：微调文本到图像扩散模型，实现主体驱动的生成》，*arXiv* 预印本 *arXiv:2208.12242*，2022年。[10] L. Luzi、A. Siahkoohi、P. M. Mayer、J. Casco-Rodriguez 和 R. G. Baraniuk，《Boomerang：使用扩散模型在图像流形上进行局部采样》，*arXiv* 预印本 *arXiv:2210.12100*，2022年。[11] A. Sherstinsky，《循环神经网络（RNN）和长短期记忆（LSTM）网络基础》，《*Physica D*：非线性现象》，第404卷，2020年。[12] S. Hochreiter 和 J. Schmidhuber，《长短期记忆》，《神经计算》，第9卷，第8期，第1735–1780页，1997年。[13] B. Kiani、R. Balestriero、Y. LeCun 和 S. Lloyd，《projUNN：一种使用酉矩阵训练深度网络的高效方法》，载于神经信息处理系统进展，第35卷，2022年。[14] A. Vaswani、N. Shazeer、N. Parmar、J. Uszkoreit、L. Jones、A. N. Gomez、Ł. Kaiser 和 I. Polo-sukhin，《《Attention Is All You Need》》，载于神经信息处理系统进展，第30卷，2017年。[15] A. Gu、K. Goel 和 C. R´e，《利用结构化状态空间高效建模长序列》，载于国际学习表征会议，2022年。[16] T. Nguyen、V. Suliafu、S. Osher、L. Chen 和 B. Wang，《FMMformer：通过分解近场与远场注意力实现高效且灵活的Transformer》，载于神经信息处理系统进展，第34卷，2021年。

[17] A. Dosovitskiy、L. Beyer、A. Kolesnikov、D. Weissenborn、X. Zhai、T. Unterthiner、M. Dehghani、M. Minderer、G. Heigold、S. Gelly、J. Uszkoreit 和 N. Houlsby，《一张图像胜过16×16个词：用于大规模图像识别的Transformer》，载于国际学习表征会议，2021年。[18] J. Ho、A. Jain 和 P. Abbeel，《去噪扩散概率模型》，载于神经信息处理系统进展，第33卷，2020年。[19] C. Hawthorne、I. Simon、A. Roberts、N. Zeghidour、J. Gardner、E. Manilow 和 J. Engel，《利用频谱图扩散合成多乐器音乐》，载于国际音乐信息检索学会会议，第23卷，2022年。[20] S. Ferretti，《作为网络的吉他独奏》，载于*IEEE*国际多媒体与博览会会议，2016年。[21] Y.-H. Chen、W.-Y. Hsiao、T.-K. Hsieh、J.-S. R. Jang 和 Y.-H. Yang，《迈向复调电吉他音乐的自动转录：一种新数据集和多损失Transformer模型》，载于*IEEE*国际声学、语音与信号处理会议，2022年。[22] Y. Zhou、Y. Ju 和 L. Xie，《AnimeTab：一个新的动漫与游戏音乐吉他六线谱数据集》，2022年。[23] P. Sarmento、A. Kumar、C. Carr、Z. Zukowski、M. Barthet 和 Y.-H. Yang，《DadaGP：面向序列模型的令牌化GuitarPro歌曲数据集》，载于国际音乐信息检索学会会议，第22卷，2021年。[24] Y.-H. Chen、Y.-H. Huang、W.-Y. Hsiao 和 Y.-H. Yang，《利用Transformer和律动建模自动创作吉他六线谱》，载于国际音乐信息检索学会会议，第21卷，2020年。