团队编号： 2022XJTUTFCX0365



第三十四届“腾飞杯”创新创业大赛创新赛道

参赛作品

**作品名称 ：** 基于深度学习方法的和弦生成模型

类别：

☑自然科学类学术论文

□哲学社会科学类社会调查报告和学术论文

□科技发明制作类

**作品名称：基于深度学习方法的和弦生成模型**

**项目分类：信息技术**

**申请人：李书宇**

**指导教师：李晨**

摘 要

深度学习方法在音乐领域已得到较为广泛的应用，并取得了不错的效果。和弦作为音乐中一个重要的概念，其生成是音乐创作中的基本任务之一。机器学习在音乐方面的应用对于人类对机器智能的理解、音乐的理解都有重要意义，同时人工智能技术也能丰富人们在音乐方面的研究。本文基于Transformer模型构建了和弦生成模型，进一步深入和弦的内部组成结构，以组成音的方式拆分和弦，希望生成更加精细和灵活的和弦。同时结合色彩和声理论构建了符合乐理规律的评估方法和匹配方法，希望通过较简单的模型得到较好的输出效果，并利用模板匹配的方法提升模型的实际效果和可拓展性。应用深度学习方法的人工智能模型强调数据驱动，希望机器能够从海量的数据中自行挖掘、提取和学习有用的信息，并利用这些信息完成所期望的任务。但是音乐是一种复杂且抽象的艺术，其数据庞大且多样，难以用简单的模型达到很好的拟合效果，而复杂的模型需要大量的投入，也难以进入人们的日常生活。本文向深度学习构建的模型中引入乐理规则，使得简单的模型也能够通过规则得到较合理的输出，模型准确性和实际效果都较未引入规则的模型有明显提升。并且模板库可根据需求进行调整，从而在不改变模型的情况下改变模型的输出以适应不同的任务情境，模型具有很好的灵活性和可拓展性。

**关 键 词：人工智能；和弦；生成；神经网络**

目 录

[1 绪论 1](#_Toc30312)

[2 研究背景 1](#_Toc27243)

[2.1 和弦生成 1](#_Toc8347)

[2.2 相关研究 2](#_Toc2708)

[2.3 BLSTM 2](#_Toc7544)

[2.4 色彩和声理论 2](#_Toc20239)

[3 研究方法 4](#_Toc9590)

[3.1问题描述 4](#_Toc8943)

[3.2数据处理 4](#_Toc8679)

[3.2.1旋律和和弦的表示法 4](#_Toc4746)

[3.2.2数据归一化 5](#_Toc15596)

[3.3 模型构建 5](#_Toc14869)

[3.4 评估方法 7](#_Toc14875)

[3.4.1 和弦的多标签评估法 7](#_Toc20027)

[3.4.2 和弦的色彩度匹配评估法 7](#_Toc9523)

[4 实验 8](#_Toc13944)

[4.1 数据集 8](#_Toc3986)

[4.1.1 数据集构建 8](#_Toc12090)

[4.1.2 数据对齐 9](#_Toc6735)

[4.2 模型训练 9](#_Toc23061)

[4.2.1 损失函数 9](#_Toc14882)

[4.2.2 模型参数选择 9](#_Toc5262)

[4.3 色彩度输出匹配 10](#_Toc19325)

[4.3.1 基于直觉的简单转换方法 10](#_Toc29257)

[4.3.2 基于色彩度和弦匹配的转换方法 11](#_Toc134)

[4.4 模型评估 11](#_Toc23968)

[5 结论与展望 13](#_Toc16338)

[5.1 模型优势与缺陷 13](#_Toc14823)

[5.1.1 模型优势 13](#_Toc534)

[5.1.2 模型缺陷 13](#_Toc31984)

[5.2 研究展望 13](#_Toc28192)

[参考文献 14](#_Toc8585)

[附 录 15](#_Toc11734)

[致 谢 16](#_Toc8967)

1 绪论

近年来，利用深度学习方法的人工智能模型被广泛应用于艺术领域，并取得了相当不错的效果。而音乐是艺术领域中一个重要的分支，和弦的编排则是音乐创作中的一个重要环节。机器学习在音乐方面的应用对于人类对机器智能的理解、音乐的理解都有重要意义，同时人工智能技术也能丰富人们在音乐方面的研究。

在乐理中，音乐可以被拆分为旋律和和弦两个部分，二者彼此呼应、紧密相连。而给一段旋律配和弦，即为一段给定旋律，编写其相应的和弦，是音乐创作中一项非常基础而重要的工作。在和声学中，为旋律配和弦有着一套较为完备，有既定逻辑的体系。因而对于专业的音乐创作者而言，给一段旋律配上比较合理的和弦进行是较为容易的。但对于不具备和声学知识的人而言，这会是一个相当困难的任务。

不同于美术，音乐是时间上的艺术，具有时序逻辑。每个音符在时间上是一个离散的点，同时音乐中的一系列音符又构成一个序列。因此我们需要构建一个能够充分学习音乐时序特征的神经网络模型，从而根据旋律的时序特征，生成期望的和弦序列。

在先前的研究工作中，Tsushima,Nakamura,Yoshii等人基于HMM构建了旋律到和弦的树形生成模型[1]。Lim,Rhyu,Lee等人利用BLSTM构建了由旋律生成和弦的模型[2]。这些工作都取得了不错的效果。但是，先前的这些和弦生成模型均把和弦看作一个简单的标签，仅从数学的角度构建模型拟合旋律到和弦的映射关系，并没有结合乐理考虑旋律和和弦。同时，这些工作对和弦种类不具有可拓展性，如Lim等人的工作中将和弦分类为24种，但是如果考虑更多种类的和弦就需要重新构建和训练模型。本文提出的方法将结合乐理构建和弦生成模型，深入和弦的底层结构，从而可以利用较简单的模型和较少的数据得到较好的输出效果，同时通过色彩度与模板匹配的方法使模型对和弦种类具有很好的可拓展性。

2 研究背景

2.1 和弦生成

对音乐创作者来说，给旋律配和弦是一个较为困难的过程，这既需要熟知和弦进展和又要考虑音调的和谐。虽然编配和弦在作曲中非常重要，但其过程却很困难，尤其是对于那些缺乏音乐研究相关的经验和领域知识的新手而言。

和弦的生成，可以对应自然语言处理里的语言生成，因为它们就是用 symbolic token 符号表示的语言。[3]大多数现代音乐创作作品都是基于MIDI事件的方法。他们将乐曲转换为MIDI事件标记序列，并使用自然语言处理方法处理标记序列。基于MIDI事件的方法可以更好地学习不同音乐事件之间的时间相关性，从而显示出更稳健的生成性能[4]。

2.2 相关研究

目前比较相关的机器学习任务是自动和弦生成。E. C. Lee and M. W. Park和S. D. You and P. Liu应用隐马尔可夫模型（HMM）在该任务上取得了较好的成果[5,6]。单个HMM与12个旋律半音向量一起使用作为观察，而相应的和弦作为隐藏状态[1]。

但是使用HMM的模型均存在明显的缺陷。这是因为HMM假设当前节点观测值与其相邻节点的输入和输出无关，仅取决于当前节点输入。这些缺点在旋律任务的和弦生成中也可以观察到，因为西方音调音乐的和弦进行和旋律序列中存在长期依赖性[7]。

2.3 BLSTM

循环神经网络（RNN）在设计的时候将某一节点的输出和前一个节点关联起来，因而能够很好地处理包含时序信息的输入数据。长短期记忆（LSTM）是一种特殊的循环神经网络，保留了RNN善于处理时序信息的优势的同时，改善了RNN训练过程中常见的梯度消失和梯度爆炸问题。目前，LSTM已经在语音识别、文本生成和视频标记、机器翻译等领域有着广泛的应用。而在和弦生成领域，相比较于传统的HMM等方法，由于和弦进行和旋律序列中存在长期依赖性，LSTM的递归层更适合于建模旋律和和弦之间的关系。因此，LSTM通常能生成更多样的和弦序列。

结合常识和现有音乐研究，我们不难发现得出一个结论：旋律序列及对应和弦序列和前后节点输入均有关，我们最好在训练时同时提供顺序上下文的信息。因此，我们应用了双向LSTM网络（Bi-directional Long Short-Term Memory，BLSTM），以便网络不仅可以向前而且也可以向后反映音乐上下文[2]。

2.4 色彩和声理论

一个和弦进行到另一个和弦，其“意外程度”、“新鲜感”可量化为一个数值，称之为色彩度，此即和弦进行的色彩度理论[8]。两和弦间的色彩度描述了两个和弦的“差异”。可以认为，两个色彩度相同的和弦在乐理上的作用相似，听者听该和弦感受到的情感色彩相似。

两和弦间色彩度的计算采用如下的方法：

首先给12平均律中的每一个音按照“五度圈”的方式赋相应的色彩值，如下图所示，从而得到一个相应的色彩值向量。

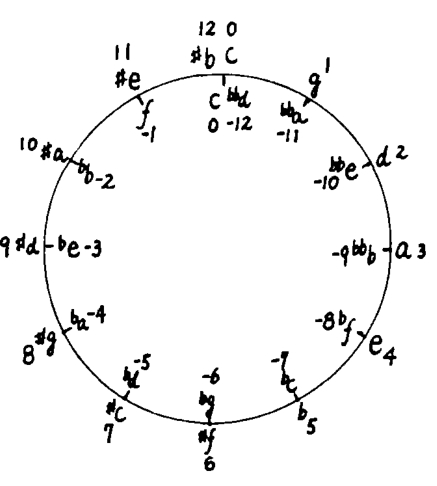


图2-1 五度圈中的音的色彩值[8]

记色彩值映射函数为f，前和弦及其各个组成音为A=()；后和弦及其各个组成音为B=()。

称A与B的毛色彩度为

相应的真色彩度为

其中归一化函数

这里一般取。

3 研究方法

3.1问题描述

在乐理中，音乐可以被拆分为旋律和和弦两部分，二者彼此呼应、紧密相连。在给定一段旋律后，依照乐理给出的逻辑规律，便可以为这段旋律编配合适的和弦序列。基于上述分析，由旋律生成和弦的过程便可以抽象为，提取旋律的时序特征，再由此时序特征生成对应的和弦序列。

在音乐中，一段旋律的音符构成比一个和弦要更加复杂，旋律的音符组合方式要远大于和弦，因而可以将和弦理解成旋律的标签，和弦生成的过程类似于将旋律按照时序的方式进行分类并生成相应的标签。上文所提到的这些研究工作[1,2]便是基于这种考虑。

然而，和弦不同于传统意义上的用于分类的标签，它是由更加底层的音符排列组合而成，可以向音符的层面继续划分。因此，和弦的复杂程度取决于音符排布的复杂程度，和弦的种类也并非只有简单的几十或几百个。如果想要穷尽和弦的所有种类，生成更加复杂丰富的和弦序列和音乐，就需要深入和弦的底层结构，从音符出发构建和弦。但这样就需要复杂的模型和大量的数据，在少量数据的情况下难以达到预期的效果。本文希望结合乐理构建一种方法，采用较简单的模型和较少的数据，取得较好的和弦生成效果。

3.2数据处理

3.2.1旋律和和弦的表示法

根据乐理知识[9]，在纵向的音高层面，一个八度内所有的音可以按照音高均等地分为12份（依次记为C,C#,D,D#,E,F,F#,G,G#,A,A#,B），每一份都是一个独立的音符，因而我们可以用12维向量表示旋律音和和弦音。向量的每一个维度表示这个旋律或和弦中是否包含这一维度的音符。同时，在音乐横向的时序结构中，可以以小节进行划分，小节内的音组合成一个完整的结构，小节之间彼此关联。基于上述知识，本文以小节为基本单位划分音乐，认为每个小节由一段旋律和一个和弦构成，旋律和和弦分别由一个12维向量表示。



图 3-1 一个小节中的旋律

表 3-1 图3-1的向量表示

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C | C# | D | D# | E | F | F# | G | G# | A | A# | B |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

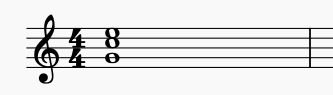


图 3-2 一个小节中的和弦

表 3-2 图3-2的向量表示

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C | C# | D | D# | E | F | F# | G | G# | A | A# | B |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

3.2.2数据归一化

在音乐中存在调的概念，不同调的音乐其旋律和和弦的音符排布方式也会有区别。因此，为了统一不同调的音乐，我们以C大调为基准，将其他调的音乐通过移调操作，全部归一至C大调，从而实现数据的归一化。

同时，音乐的节奏也受到拍号的影响，不同拍号的音乐，其每个小节内的音符拍子数不同。而通过以小节为基本单位划分和处理音乐数据，将每个小节的旋律和和弦统一成12维向量，可以规避拍号的影响。

3.3 模型构建

基于十二平均律，一段旋律的乐音组成可以用一个12维的0-1向量来简单描述：将十二平均律中的每一个音依次与12维向量中的每一位对应，则向量中的一维表示该段旋律中是否包含与之对应的音，包含则该维为1，否则为0。同理，一个和弦的组成也可以由一个12维的0-1向量来表示，其包含的音便是向量中非零维对应的音。

根据对问题的分析，可以认为和弦的组成与其修饰段的主旋律的乐音组成有较强的相关性。因此对于和弦生成问题，一段旋律的有效信息可以用一组向量序列来描述：首先基于和弦的平均作用区间长度，将旋律划分为若干段。再对于划分出来的每一小段，使用一个12维向量来描述该段中乐音的组成。和弦生成问题便可以形式化地描述为一个向量生成问题：给定一个乐曲集合，其中的每一个元素都是一个形如的二元组序列，表示一首曲子中的若干段旋律向量以及其对应的和弦向量。现要求根据中满足的匹配规律，对于给定的旋律向量，生成其对应的和弦向量。

然而由于旋律的连贯性和整体性，这种匹配关系不能简单地从一对向量的维度进行总结，而是要从曲子的维度，也就是将向量放在向量序列中考虑其匹配关系。由于该种匹配关系难以通过比较分析提炼成代数形式，考虑使用深度学习的方法来解决该问题。

Transformer模型是一种完全基于自注意力机制的深度学习模型。每个小节的12维向量在经过位置编码后送入Transformer Encoder中，利用自注意力机制得到该小节关于其他小节的注意力信息，这种处理方式也符合实际音乐创作中综合考虑前后小节的信息来编配该小节和弦的做法。通过这种方式，就可以提取到每一段音乐的序列信息，随后再将每小节的信息通过全连接层构成的Decoder进行解码，得到每小节和弦的12维向量表示。

构建的和弦生成网络结构如图所示。其中每小节的旋律向量先经过位置编码层，将位置信息加入到向量中。随后将含有位置信息的向量输入Transformer Encoder中，利用自注意力机制提取序列信息。最后再由全连接层构成的Decoder解码输出所需的和弦向量。输入层为12维0-1向量序列，输出层输出一个12维向量表示每一维对应音的出现概率，并选用Sigmoid函数作为激活函数将每维的数值映射至0-1的范围。其他层的激活函数均选用ReLU函数。

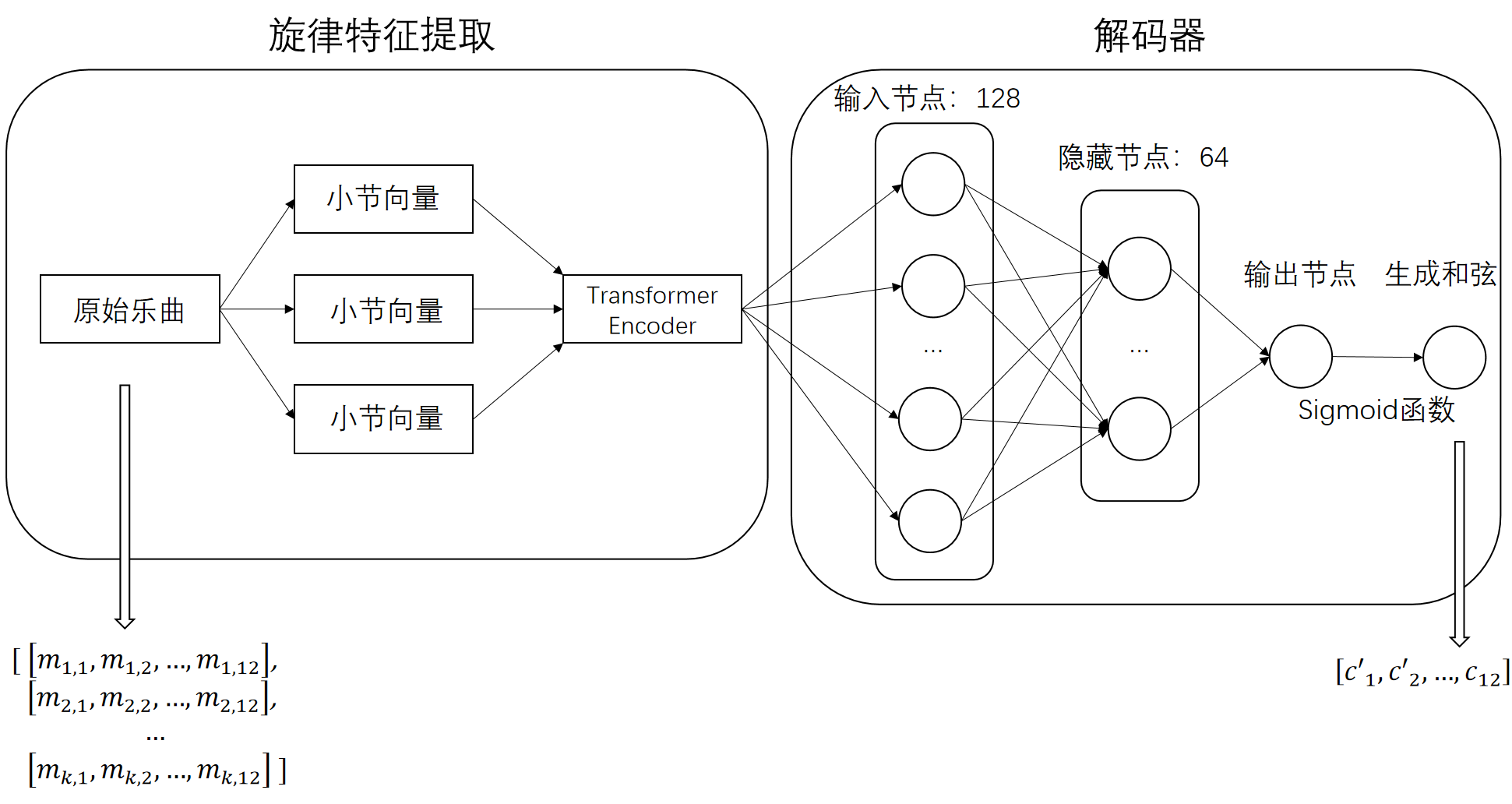


图 3-3 模型结构

3.4 评估方法

如前所述，一段旋律的和弦标签和模型的输出均被处理为12维0-1向量（1表示和弦中有该音，0表示和弦中没有该音）。

3.4.1 和弦的多标签评估法

一种简单的评估方式是采用多标签评估的方法，直接对两个向量进行比较。

设两个向量分别为，定义函数。则我们定义两个和弦的简单相似度为两个和弦向量的某一维度相同的概率：

其优点是计算速度快，便于在一定程度上迅速而直观地反映实验结果。

然而，考虑到乐理中和弦的结构与特性，相比于和弦中没有某个音，我们更加关注和弦中存在哪些音，因而向量中0与1的作用并不等同，两个向量的多标签相似比难以准确描述模型的结果。因此我们基于和弦色彩度理论，给出了新的评估方法：

3.4.2 和弦的色彩度匹配评估法

将输出的和弦与所有的标准模板（本文采用所有大小三和弦作为模板库）按照2.4节所述方法计算真色彩度，从而得出它们之间的距离，将输出和弦与其距

离最近的模板三和弦进行匹配；将标签和弦亦与标准模板大小三和弦进行匹配。如果两者匹配到了相同的三和弦，则认为模型计算的结果和标签相同，否则认为其不相同。这种评估方式基于乐理，相较于之前的评估方法更加准确和方便，更加符合人耳对和弦的听感。并且我们可以以色差为基础去评估输出和弦与标签和弦的分数，这种评估方式相较于之前的非零即一的评估方式更加合理。同时，和弦模板库可以根据需要进行扩充，如添加增和弦、减和弦、七和弦等和弦类型，这将会使匹配和评估更加准确。

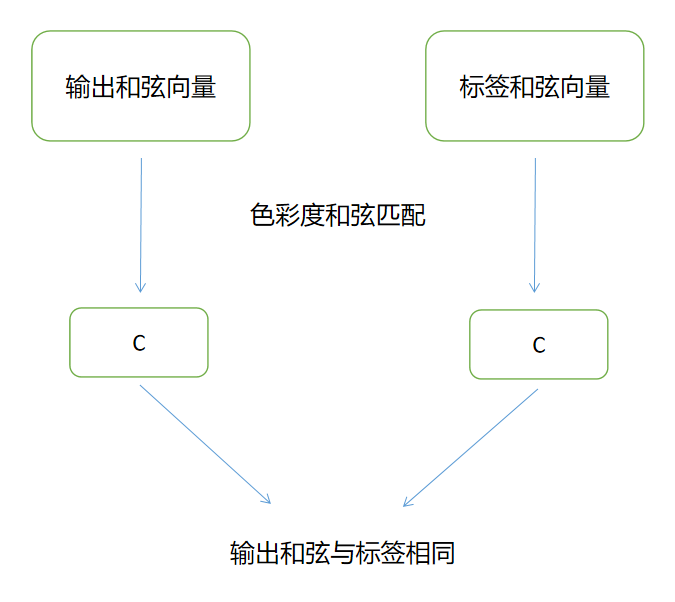


图 3-4 色彩度匹配评估法

4 实验

4.1 数据集

4.1.1 数据集构建

数据集收集自公共乐谱网站Wikifonia.org，共收集了19859个midi格式的音乐文件，这些音乐文件包括了摇滚、流行、乡村、爵士、民谣、R&B，儿童歌曲等多种音乐类型。这些文件已打包好以供下载[[1]](#footnote-0)。将下载好的数据集按照3.2所述方式进行归一化和预处理，得到处理好的向量数据集，之后按照8:1:1划分训练集、验证集、测试集。

4.1.2 数据对齐

由于每个歌曲的小节数不尽相同，我们通过预处理得到的向量表长度不一致，因此需要将数据进行截断或填充，便于批量训练模型。经过统计，在数据集中绝大多数歌曲的小节数在10-20之间。我们测试了以8,16,32小节为基准的对齐长度，最后选择了效果最好的16小节作为数据集的对齐长度。短于16小节的音乐片段在末尾被填充0向量进行补齐，长于16小节的音乐片段在第16小节后进行截断。

4.2 模型训练

4.2.1 损失函数

在模型训练过程中需要使用损失函数度量模型输出与目标之间的距离，从而针对此距离进行优化。如前所述，模型输出为12维向量，每个维度为0-1之间的概率值，而目标为一个12维0-1向量。交叉熵损失函数可以度量概率分布之间的相似程度，因此可选用交叉熵损失函数作为训练的损失函数。然而，在训练中，采用该损失函数时效果并不理想，模型倾向于将和弦预测为一个全零向量。通过分析数据集可以发现，绝大多数和弦都只有3到4个组成音，即每个12维和弦向量仅有3到4个非零值。因此数据集是稀疏的，可以认为和弦的预测是一个稀疏多标签分类任务。由于数据集的稀疏性，0和1并不平衡，0在数据集中占有更高的权重，因而当模型的复杂程度不足以完全拟合出0和1的分布情况时，模型会更倾向于输出0，这就导致模型输出大量零向量。

为了解决上述问题，我们尝试了下面两种方法。

其一，输出全零向量是模型试图泛化地拟合所有维度的数据造成的。因而模型会更加关注数据的整体分布，即数据的多数维度均为0，而忽视了数据的局部为1的分布。基于以上考虑，我们去掉了模型的Dropout层，降低模型的泛化能力以解决这一问题。

其二，可以重新分配数据集中0和1的权重，提升模型在稀疏数据中对1的关注，从而平衡数据集中0-1分布不均的问题。基于以上考虑，我们重新选择了损失函数，选择BCEWithLogitsLoss作为新的损失函数，并且设置pos\_weight的值为4，即将1的原始权重乘4作为平衡后的训练数据。

4.2.2 模型参数选择

按照3.3所述结构构建模型，采用两种损失函数的模型参数及训练参数如表所示。

表 4-1 交叉熵损失函数模型及训练参数

|  |  |
| --- | --- |
| 项目 | 参数 |
| 批量大小 | 64 |
| 对齐长度 | 16 |
| 输入维度 | 12 |
| 输出维度 | 12 |
| Transformer Encoder隐藏层大小 | 128 |
| Transformer Encoder层数 | 6 |
| 多头注意力 | 4 |
| 解码器隐藏层大小 | 128 |
| dropout | 0 |
| 学习率 | 0.001 |
| 损失函数 | CrossEntropyLoss |
| 优化方法 | Adam |

表 4-2 BCE损失函数模型及训练参数

|  |  |
| --- | --- |
| 项目 | 参数 |
| 批量大小 | 64 |
| 对齐长度 | 16 |
| 输入维度 | 12 |
| 输出维度 | 12 |
| Transformer Encoder隐藏层大小 | 128 |
| Transformer Encoder层数 | 6 |
| 多头注意力 | 4 |
| 解码器隐藏层大小 | 128 |
| dropout | 0.2 |
| 学习率 | 0.001 |
| 损失函数 | BCEWithLogitsLoss |
| pos\_weight | 4 |
| 优化方法 | Adam |

4.3 色彩度输出匹配

如前文所述，模型对和弦的预测值为12维概率向量，需要一种算法将该概率向量转换为12维0-1向量，即一个由组成音表示的和弦向量。

4.3.1 基于直觉的简单转换方法

一种简单的方法是设置lim参数作为阈限，将概率高于该阈限维度取为1，即和弦包括该音，反之取为0。然而在实际音乐中，一个和弦的组成音一般在4个左右，因此需要限制模型预测和弦向量的最大非零维数，记为参数lim\_num。本文设置lim=0.5, lim\_num=3。

然而，由上述方法转换得到的和弦向量在听感上效果不佳。这是因为在实际音乐中和弦的组成音排布有着一定规律，不符合规律的排布方式，即使是一个音的偏差或者增减，都会对和弦听感造成很大的影响。因而如果模型不足以精确拟合所有的和弦组成音分布，那么仅仅依靠这种简单的转换，其输出不一定符合和弦组成音的排布规律。因此我们结合乐理，基于色彩度和弦匹配构建了转换方法。

4.3.2 基于色彩度和弦匹配的转换方法

在3.4.2中介绍了色彩度匹配的方法，即利用色彩度作为评分，将输出和弦向量与预先设置的模板库中距离最近的模板进行匹配。我们利用这一方法，将4.3.1中得到的和弦向量映射为模板库中的模板，使得模型输出始终满足我们的预先设定，从而得到听感较好的和弦。在本文的工作中，我们采用全部大小三和弦作为模板库。此外，该方法具有很好的灵活性和可拓展性，在保持模型结构、权重不变的情况下，只需要改变模板库，便可以得到不同的预期和弦输出，如向模板库中增加新的和弦种类，便可以得到更加丰富的和弦输出；或设置模板库中各模板的权重，就可以得到有着不同风格的和弦输出。

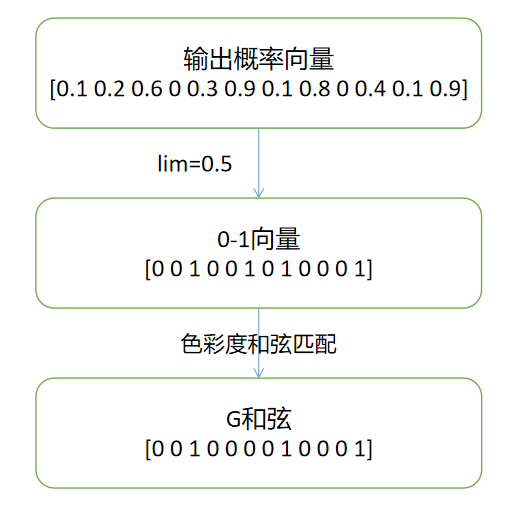


图 4-1 色彩度和弦匹配转换法

4.4 模型评估

我们将模型在测试集中，使用两种评估方法进行评估。

表4-3 交叉熵损失函数模型正确率

|  |  |
| --- | --- |
| 损失函数：CrossEntropyLoss | |
| 评估方法 | 正确率 |
| 多标签法 | 0.73 |
| 色彩度匹配法 | 0.25 |

表4-4 BCE损失函数模型正确率

|  |  |
| --- | --- |
| 损失函数：BCEWithLogitsLoss | |
| 评估方法 | 正确率 |
| 多标签法 | 0.77 |
| 色彩度匹配法 | 0.27 |

在模型输出的基础上，使用色彩度和弦匹配法对输出进行转换，并使用多标签法计算转换后的和弦向量的正确率。

表4-5 转换前后和弦向量正确率对比

|  |  |
| --- | --- |
| 损失函数：CrossEntropyLoss | |
| 和弦向量 | 正确率 |
| 原始输出向量 | 0.73 |
| 转换向量 | 0.76 |
| 损失函数：BCEWithLogitsLoss | |
| 和弦向量 | 正确率 |
| 原始输出向量 | 0.77 |
| 转换向量 | 0.79 |

由上述正确率比较可见，带权重的BCE损失函数更适合作为该模型的损失函数。同时，通过色彩度匹配的方法对和弦向量进行转换可以有效地提升模型输出的准确度。并且可以预见，通过增加模板库中的模板，匹配更加精细和丰富的模板库，能够进一步提升模型预测的正确率。

5 结论与展望

本文基于BLSTM模型构建了由旋律生成和弦的模型，并基于色彩和声理论构建了针对和弦生成的更加符合乐理的评估方法和匹配方法。色彩度和弦匹配转换方法能够有效地提升模型准确度。

5.1 模型优势与缺陷

5.1.1 模型优势

本文提出的模型具有以下优势：

1）深入和弦的组成结构，能够更加精细地拟合和弦；

2）结合乐理，提出更加符合实际的评估方法和匹配方法；

3）具有很好的可拓展性，在模型结构不变的情况下可以通过改变模板库实现不同情况下的和弦输出。

5.1.2 模型缺陷

本文提出的模型尚存在以下不足：

1）仅考虑每小节的第一个和弦，无法应对同一小节中使用多个和弦的情况；

2）没有考虑每小节旋律音的长度、位置、数量等信息，输入数据提供给模型的信息有限，模型可能无法很好地进行预测。

解决以上问题后，模型的准确度和泛化能力将得到进一步提升。

5.2 研究展望

深度学习方法可以很好地应用在音乐领域，辅助人们的音乐学习、音乐创作，并帮助人们理解音乐。但音乐是一种复杂且抽象的艺术，其数据庞大且多样，难以用简单的模型达到很好的拟合效果，而复杂的模型需要大量的投入，也难以进入人们的日常生活。因而将乐理中的规则与深度学习的方法相结合，用自上而下的规则辅助模型自下而上的学习过程，将会是一个有意义的研究方向。

参考文献

[1] Tsushima H, Nakamura E, Yoshii K. Bayesian melody harmonization based on a tree-structured generative model of chord sequences and melodies[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2020, 28: 1644-1655.

[2] Lim H, Rhyu S, Lee K. Chord generation from symbolic melody using BLSTM networks[J]. arXiv preprint arXiv:1712.01011, 2017.

[3] 谭旭，AI音乐，技术与艺术的碰撞[R/OL]，2021-07-22，

https://www.msra.cn/zh-cn/news/features/xu-tan-ai-music

[4] Zhang C, Ren Y, Zhang K, et al. SDMuse: Stochastic Differential Music Editing and Generation via Hybrid Representation[J]. arXiv preprint arXiv:2211.00222, 2022.

[5] E. C. Lee and M. W. Park: “Music Chord Recommendation of Self Composed Melodic Lines for Making Instrumental Sound,” Multimedia Tools and Applications, pp. 1-17, 2016.

[6] S. D. You and P. Liu: “Automatic Chord Generation System Using Basic Music Theory and Genetic Algorithm,” Proceedings of the IEEE Conference on Consumer Electronics (ICCE), pp. 1-2, 2016.

[7] Mittal G, Engel J, Hawthorne C, et al. Symbolic music generation with diffusion models[J]. arXiv preprint arXiv:2103.16091, 2021.

[8] 华萃康.论和弦之间的色彩对比(上)——剖析“色彩度”“色差”的实质及主要缺点,并改进其计算方法[J].南京艺术学院学报(音乐与表演版),1993(02):8-16.

[9] 李重光. 音乐理论基础[M]. 人民音乐出版社, 1962.

附 录

附录1 部分生成效果展示



图 1 样例1（上方为旋律输入，下方为和弦输出，下同）



图 2 样例2

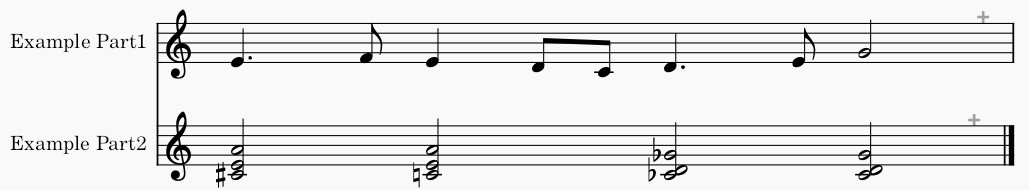


图 3 样例3

致 谢

在项目思路的提出、模型构建、代码编写和论文写作的各个阶段，我们遇到过诸多困难。幸运的是，在此过程中我们也得到了老师与同学的帮助，从他们身上学到了许多，我们在此向他们致以诚挚的感谢。

感谢李晨老师帮助我们明确了研究方向和思路，在我们构建模型的过程中提供指导。

感谢刘洋河学长、黄俊豪同学在色彩和声理论、和弦模板匹配方面的帮助。

感谢李泽爽同学、张家宁同学在模型构建和数据处理方面的帮助。

感谢曹佳恺学长在代码编写和论文写作方面的指导。

如果没有这些老师和同学的帮助，我们的工作将困难重重，再次感谢他们为我们的工作能够顺利进行提供的帮助。

1. https://drive.google.com/file/d/1LH7v59EsS4rwhy9HKaEdE9frcuHZTabQ/view [↑](#footnote-ref-0)