

**本 科 毕 业 设 计（论文）**



**题目: 音乐旋律的和弦编配算法的研究与实现**

**姓 名 孙一帅**

**学 院 软件学院**

**专 业 软件工程**

**班 级 2016211504**

**学 号 2016212026**

**班内序号 20**

**指导教师 崔毅东**

**2020年5月**

**北 京 邮 电 大 学**

**本科毕业设计（论文）任务书**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 软件学院 | | 专业 | 软件工程 | | 班级 | 2016211504 |
| 学生姓名 | 孙一帅 | | 学号 | 2016212026 | | 班内序号 | 20 |
| 指导教师姓名 | 崔毅东 | | 所在单位 | 软件学院 | | 职称 | 副教授 |
| 设计(论文)题目 | （中文）音乐旋律的和弦编配算法的研究与实现 | | | | | | |
| （英文）The Research and Implementation of the Chord Assembling Algorithms for Melodies | | | | | | |
| 题目分类 | 工程实践类 研究设计类□ 理论分析类□ | | | | | | |
| 题目来源 | 题目是否来源于科研项目 是□ 否 | | | | | | |
| 科研项目名称： | | | | | | |
| 科研项目负责人： | | | | | | |
| 主要任务及目标：  在进行基于人工智能的音乐创作研究过程中，需要为给定的旋律自动编配和弦。目前已有一些和弦编配算法，但其编配结果平淡、缺少变化。研究基于机器学习的和弦编配算法并与已有算法进行对比和测试。  要求同学具备基本的乐理知识，能够演奏一种乐器（乐器不限）。最好了解少量的和声学知识，比如三和弦、和弦级数、和弦走向等。  目标如下：   1. 学习经典的机器学习模型并了解其用法。 2. 训练出一个机器学习模型，能够根据已给出旋律自动添加和弦。 3. 程序最终能够根据旋律走向选择最和谐的和弦进行编排，符合基本乐理原理。 4. 最终程序可以选择不同种和弦的分解型进行不同效果的演奏。 5. 最终音乐中生成的和弦拥有多种和弦变式，具有类似人的音乐创作习惯。 6. 系统拥有简易前端，前端可以方便传入旋律文件、后端进行预测后输出到前端处理好的音乐文件。 7. 对多种旋律文件进行测试，增加程序的稳定性和鲁棒性。 | | | | | | | |
| 主要内容：   1. 学习keras框架的使用方法。 2. 学习Python3基础知识。 3. 学习midi文件、musicxml文件的编码格式及处理。 4. 复习相关乐理知识。 5. 复习声学知识。 6. 了解相关机器学习算法。 7. 设计实现模型。 8. 对模型进行测试、优化。 9. 对模型进行乐理方面的优化和限制。 10. 可视化前端的开发。 | | | | | | | |
| 主要参考文献：   1. <https://github.com/musescore/MuseScore> 2. <https://musescore.org/zh-hans> 3. 尤静波，基础乐理[J]，2018 4. 李航，统计学习方法，2012 5. 周志华，机器学习，2016 6. Yifeng Teng，Anny Zhao，generating nontrivial melodies for music as a service[D]   [7] 王卉，浅谈流行歌曲的钢琴编配，2014  [8] 钢琴伴奏多元混合编配实例解析，2019 | | | | | | | |
| 进度安排：  第1-2周：结合自身兴趣，与导师讨论选题方向，确定选题内容。针对目标选题，确定主攻方向和主要难点，编写任务书和开题报告。  第3-5周：查阅相关论文，明确项目实现的具体流程以及相关细节。  第6-10周：学习并完成对各类音乐文件的字符化处理及转换，学习音乐文件有关的python第三方库；了解有关机器学习的算法，并确定模型的方向。  第11-15周：初步得出模型，并尝试调优。复习相关乐理知识，确定并实现下一步调优方向；进行多种测试，提高程序鲁棒性。  第16-19周：编写前端程序，使程序更加方便使用和测试。  第20-24周：撰写毕业论文，准备答辩。 | | | | | | | |
| 指导教师签字 | |  | | 日期 | 2019年 11月 4日 | | |

北京邮电大学本科毕业设计（论文）答辩决议书

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 软件学院 | | | 专业 | | 软件工程 | | | 班级 | 2016211504 |
| 学生姓名 | 孙一帅 | | | 学号 | | 2016212026 | | | 班内序号 | 20 |
| 毕业设计  （论文）题目 | （中文）音乐旋律的和弦编配算法的研究与实现 | | | | | | | | 百分制成绩 | 87 |
| （英文）The Research and Implemention of the Chord Assembling Algorithms for Specific Melodies | | | | | | | | 五级分制成绩 | 良 |
| 指导教师姓名 | 崔毅东 | | 所在单位 | | | 北京邮电大学 | | | 职称 | 副教授 |
| 指导教师评语：（主要包含选题背景、意义；设计（论文）质量；设计（论文）成果、价值、创见性；论文撰写水平、文本规范程度；学生能力体现、工作量、工作态度；不足和希望等方面）  论文选题面向音乐与计算机交叉学科融合技术，论文工作量饱满，毕业设计成果有一定的参考价值；论文工作完成了音乐旋律的和弦编配算法的设计和开发，设计符合软件工程的基本方法；论文所采用的设计方案和算法较合理，对算法进行了实现和测试。  论文结构较合理，语言叙述符合逻辑，文字通顺，排版规范。该生在毕业设计过程中认真主动，能够按时参加每周工作汇报会议。  论文表明作者已经达到本科毕业设计水平。希望在今后的学习和工作过程中，再接再厉，不断培养并锻炼自己的系统思维方法和良好的工作习惯。 | | | | | | | | | | |
| 指导教师评分(满分40分) | 35 | 签字 | | |  | | 日期 | 2020 年 5 月 26日 | | |
| 答辩小组评语：（主要包含选题背景、意义；设计（论文）质量；设计（论文）成果、价值、创见性；论文撰写水平、文本规范程度；答辩准备、陈述、回答问题情况；不足和希望等方面）    论文选题来源于科研项目，面向计算机和软件技术，论文选题符合专业培养目标；论文工作达到本科毕业设计的基本要求，方案//算法//程序经过实际运行。论文结构合理，词句通顺，文本规范。答辩时表述清晰，回答问题正确。  答辩小组评分： 52 组长职称：副教授 签字：  (满分60分) 成员职称：副教授 签字：  成员职称：副教授 签字：  2020年 5 月 26 日 | | | | | | | | | | |

注：毕业设计（论文）成绩由指导教师评分（满分40分）和答辩小组评分（满分60分）相加，得出百分制成绩，再按

100-90分为“优”、89-80分为“良”、79-70分为“中”、69-60分为“及格”、60分以下为“不及格”的标准折合成五

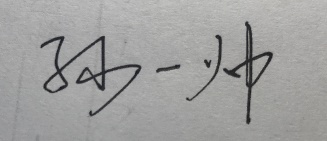
级分制成绩。

**北 京 邮 电 大 学**

**本科毕业设计（论文）诚信声明**

本人声明所呈交的毕业设计（论文），题目《音乐旋律的和弦编配算法的研究与实现》是本人在指导教师的指导下，独立进行研究工作所取得的成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。

申请学位论文与资料若有不实之处，本人承担一切相关责任。



本人签名： 日期： 2020.5.30

音乐旋律的和弦编配算法的研究与实现

摘 要

在进行基于人工智能的音乐创作研究过程中，需要为给定的旋律自动编配和弦。目前已有一些和弦编配算法，但其编配结果平淡、缺少变化。本研究基于机器学习的和弦编配算法并与已有算法进行对比和测试。

利用长短期记忆网络训练出一个机器学习模型，该模型满足音乐编配中对上下文的依赖需求；attention层能够对每一小节所占权重给出一个不错的分配。该模型能够根据已给出旋律得到每一小节的旋律重音，通过旋律重音再进行和弦的选择。

本文系统能够根据旋律重音走向和自定义选择的多种和弦进行规则选择最和谐的和弦进行方式，将旋律音映射到七个本调的基本和弦上，符合基本乐理原理。在模型预测和乐理规则的双重限制下，可以通过修改分解和弦方法来获得不同的风格与听感效果。最重要的一点是，该模型生成的和弦可以拥有自定义和弦变式而不仅仅拘泥于基本和弦，具有类似人的音乐创作习惯，并不局限于系统给定的变式方式。这种结合了机器学习以及传统乐理限制的方式使得单一算法预测和弦出现的偏音、错音问题有了得以有纠正的机会，同时在保证正确率的前提下也保留了机器学习算法的不确定性，规避了只使用乐理限制推测和弦的刻板方式。

本文实现的系统包括完整模型、乐理限制以及简易前端使用界面，并且成功完成了部署和测试，运行结果表明该系统可以方便地进行旋律文件的和弦自动编配自动谱曲、自动播放，并配有一系列的自定义及可视化功能。

**关键词** 机器学习 和弦编配 计算机音乐 Midi

**The Research and Implementation of the Chord Assembling Algorithms for Melodies**

**ABSTRACT**

# In the research process of musical creation based on the artificial intelligence, it is required to automatically arrange chords for a given melody. At present, there are some chord allocation algorithms, but the results are flat and stereotype. This paper mainly studies the chord collocation algorithm based on the comparison and testing between the mechanized chord collocation algorithm and the existing algorithms. A machine learning model is trained by utilizing both short-term and long-term memory network, which can satisfy the context-dependent requirement in music orchestration; The attention layer is likely to assign an adaptable weight to each section. This model has ability to figure out the melody stress of each section according to the given melody, and then carry out the chord selection through the melody stress.

# The system in this paper can choose the most harmonious chord progression according to the melody stress trend and a variety of custom selected chord progression rules, and then map the melody tone to the basic chord of the seven keys, which conforms to basic music theory. Under the double limitation of model prediction and music rationales, different styles and auditory effects can be obtained by modifying and disintegrating chords. The most important point is that the chords generated by this model can have a custom chord variation rather than just stick to the basic chords, which has similar musical creation habits to human beings without the limitation of the given variation mode from the system. This method combined the machine learning with the constraint of the traditional music theory allows a single algorithm to predict the off-center and wrong notes of chords, which retains the uncertainty of the machine learning algorithm on the premise of ensuring accuracy and simultaneously avoids the rigid way of using only music theory limits to predict chords.

# The system implemented in this paper includes a complete model, music theory restrictions and simple front-end interface, and has successfully completed the deployment and testing. The operation results show that the system can easily carry out the automatic chord arrangement, the automatic music composition and the automatic playback of melody files, and is equipped with a series of customization and visualization functions.

# KEY WORDS machine learning chord assembling computer music midi

**目 录**

[第一章 绪论 1](#_Toc39423404)

[1.1 课题背景 1](#_Toc39423405)

[1.1.1 作曲中和弦编配过程 1](#_Toc39423406)

[1.1.2 计算机音乐发展历史 1](#_Toc39423407)

[1.1.3 和声学的相关理论支撑 2](#_Toc39423408)

[1.2 国内外研究情况 2](#_Toc39423409)

[1.3 本章小结 2](#_Toc39423410)

[第二章 相关技术综述 4](#_Toc39423411)

[2.1 开发环境 4](#_Toc39423412)

[2.2 keras 4](#_Toc39423413)

[2.3 music21库 4](#_Toc39423414)

[2.4 midi与musicXML文件格式 4](#_Toc39423415)

[2.5 Django框架 5](#_Toc39423416)

[第三章 基于LSTM的和弦编配模型 6](#_Toc39423417)

[3.1 模型的选择 6](#_Toc39423418)

[3.1.1 LSTM 6](#_Toc39423419)

[3.1.2 Attention机制 7](#_Toc39423420)

[3.2 数据预处理 8](#_Toc39423421)

[3.2.1 数据集的选择 8](#_Toc39423422)

[3.2.2 数据清洗及格式转换 9](#_Toc39423423)

[3.2.3 音符记号与字符之间的转换 10](#_Toc39423424)

[3.2 建立模型 10](#_Toc39423425)

[3.3.1 参数选择 10](#_Toc39423426)

[3.3.2 测试集预测结果分析 11](#_Toc39423427)

[3.3.3 实际演奏分析 11](#_Toc39423428)

[3.4 本章小结 12](#_Toc39423429)

[第四章 和声学规则的限制 13](#_Toc39423430)

[4.1 旋律调性判断 13](#_Toc39423431)

[4.2 和弦行进套路及风格化 13](#_Toc39423432)

[4.2.1 常见的和弦行进规则 14](#_Toc39423433)

[4.2.2 对套路自定义修改实现风格化 18](#_Toc39423434)

[4.3 和弦替代 19](#_Toc39423435)

[4.4 效果分析 20](#_Toc39423436)

[4.5 本章小结 20](#_Toc39423437)

[第五章 系统设计 21](#_Toc39423438)

[5.1 功能性需求 21](#_Toc39423439)

[5.1.1 旋律文件上传功能 22](#_Toc39423440)

[5.1.2 模型选择功能 22](#_Toc39423441)

[5.1.3 后端预测功能 23](#_Toc39423442)

[5.1.4 乐谱展示功能 24](#_Toc39423443)

[5.1.5 在线播放功能 24](#_Toc39423444)

[5.1.6 播放可视化功能 25](#_Toc39423445)

[5.1.7 文件下载功能 26](#_Toc39423446)

[5.2 非功能性需求及其体现 26](#_Toc39423447)

[5.2.1 系统性能 26](#_Toc39423448)

[5.2.2 可靠性 27](#_Toc39423449)

[5.2.3 可移植性 27](#_Toc39423450)

[5.2.4 可拓展性 27](#_Toc39423451)

[5.2.5 易用性 27](#_Toc39423452)

[5.3 系统架构设计 27](#_Toc39423453)

[5.3.1 系统总体功能架构 27](#_Toc39423454)

[5.3.2 系统输入输出 28](#_Toc39423455)

[5.3.3 异常处理设计 29](#_Toc39423456)

[5.4 本章小结 29](#_Toc39423457)

[第六章 系统实现与测试 30](#_Toc39423458)

[6.1 首页选择模块设计实现 30](#_Toc39423459)

[6.2 文件上传及后台预测模块设计实现 30](#_Toc39423460)

[6.2.1 文件上传模块设计实现 30](#_Toc39423461)

[6.2.2 预测模块设计实现 31](#_Toc39423462)

[6.3 乐谱展示模块设计实现 32](#_Toc39423463)

[6.4 midi播放及下载模块设计实现 33](#_Toc39423464)

[6.5 midi可视化模块设计实现 34](#_Toc39423465)

[6.6 测试 34](#_Toc39423466)

[6.6.1 测试环境 34](#_Toc39423467)

[6.6.2 测试用例编写 35](#_Toc39423468)

[6.6.3 功能展示 37](#_Toc39423469)

[6.7 本章小结 40](#_Toc39423470)

[第七章 结论 42](#_Toc39423471)

[7.1 研究结论 42](#_Toc39423472)

[7.1.1模型部分 42](#_Toc39423473)

[7.1.2前端网页部分 42](#_Toc39423474)

[7.2 展望 43](#_Toc39423475)

[参考文献 44](#_Toc39423476)

[致 谢 45](#_Toc39423477)

# 第一章 绪论

## 1.1 课题背景及目的

在作曲过程中，除了旋律贯穿整首音乐外，和弦是乐谱中最重要的一部分。和弦能够成倍地放大旋律的倾向性并具有无法代替的功能[1]，能够丰富乐曲的音域并有极多种类的变式供创作来表达作曲者的感情。在音乐编配过程中，往往和弦与旋律编配的先后可以互相调换，二者并不存在依附关系；但通常旋律来自于一段灵感，而和弦来自于经验与推敲，所以对旋律进行和弦的编配就显得尤为重要。

在进行基于人工智能的音乐创作研究过程中，需要为给定的旋律自动编配和弦。目前已有一些和弦编配算法，但其编配结果平淡、缺少变化。本研究基于机器研究的和弦编配算法并与已有算法进行对比和测试。

我们的最终目的是训练出一个机器学习模型，能够根据已给出旋律自动添加和弦，最终系统能够根据旋律走向选择最和谐的和弦进行编排，符合基本乐理原理。同时，该系统生成的和弦拥有多种和弦变式，具有类似人的音乐创作习惯。该系统具有一个方便使用的前端界面，在选择模型参数上传文件后可以自动生成完整的音乐文件。让所有只拥有旋律的音乐文件可以很简单地进行和弦编配并且试听。

### 1.1.1 作曲中和弦编配过程

在编曲过程中，不同种类音乐有着不同的编曲习惯，如在电子音乐中就会选择堆叠乐器，而在古典音乐中会选择对主乐器进行编排；可是对于和弦编配而言基本所有乐种都是有一个相同的基本流程：

首先确定歌曲的调式，然后在歌曲的开头和结尾配上主和弦，终止前的一小节选择属七和弦或属和弦，属七和弦前可以配下属和弦，然后为每小节配和弦。配和弦时时刻考虑和弦进行是否合理，最好为四度连接，最后可以根据经验与需要使用代理和弦美化和声；一些特殊风格歌曲如爵士可以选用特殊的和弦，如六和弦、十一和弦。当然，和声围绕旋律为中心的编曲方式，允许反常的和声序进和声部进行。

### 1.1.2 计算机音乐发展历史

1837年，英国的库克和惠通斯发明了有线电报，就可以使用数字信号完全代替模拟信号，标志着计算机音乐的诞生[2]。

20世纪80年代，MIDI格式的出现也标志着计算机编曲进入字符化时代。

进入21世纪，人工智能的兴起使得计算机不再局限于编曲工具，利用各种神经网络算法生成音乐或旋律的方法层出不穷，任何人都可以通过简单的操作获得一段听感尚可的音乐；人工智能在音乐领域的作用已经不言而喻。

### 1.1.3 和声学的相关理论支撑

和弦进行是属于和声学的范畴，和声学对音乐的定义是：音乐是符合人情感的时间上的艺术，是由多个稳定到不稳定再到解决的进行构成[2]。常见的进行有自然大调的T-S-D-T（包括其变体）、自然小调的t-d-s-t、和声小调的t-s-D-t等等。理论上讲和弦总数有限，即解空间有限，可以使用枚举+评估的方法得到答案；即使这种方法显然不符合人类作曲思路，但是它为和弦的自动编配提供的理论可能。

现在很多编曲软件都提供类似于“和弦助手”的功能，其在给定旋律的情况下能够自动推算出下一小节合法且悦耳的和弦名称；那么在给定整首曲子旋律的情况下，可以通过机器学习加乐理限制自动提供整首曲子的和弦名称和自动编配，能够为音乐创作人提供灵感甚至代替一部分工作，对于不懂乐理的人也可以在只想到一阵灵感旋律的情况下，得到悦耳合法的一段音乐。

## 1.2 国内外研究情况

现已有对MIDI文件中熵判断来分解旋律与和弦的音符[3]，使用机器学习算法（如马尔可夫模型）来进行和弦识别[4]，扩大了数据集的可选范围。

在和弦自动编配方面，不少公司和机构都进行了丰富的工作，如Google公司的Magenta项目、微软的AI Lab也有对应子项目。在旋律生成部分，有基于和弦的节奏和旋律交叉生成的模型；编曲过程中，有多乐器联合编曲模型以实现多乐器和谐演奏；风格控制方面，有多风格多乐器联合编曲模型[5]。但其都是单纯基于机器学习，而在实际编曲过程中，编曲的好坏与编曲人水平高低很大程度依赖于对于乐理的理解与变通，而研究在此方向的努力较少。

## 1.3 本章小结

在音乐创作过程中，和弦的编配是必不可缺少的一部分；和弦的好坏及特点直接决定一首音乐的听感；而和弦的编配是有规则可循的，这就给我们对旋律进行自动编配提供了可能。

而实际上，尤其在机器学习领域，编曲一直是一个较为热门的话题，国内外对编曲过程中的所有流程都有了十分充足的研究和成果。本文研究试图将机器学习与传统乐理研究进行结合来对旋律进行和弦编配，希望能够提供另一个方向的思路；研究同时完成了一个便于操作的界面可以进行系统测试与乐曲生成。

# 第二章 相关技术综述

## 2.1 开发环境

本系统的开发环境为Windows系统下的Python3.6，使用的开发工具为Pycharm。

HTML版本为HTML5，兼容Chrome、Firefox、Edge浏览器。

本系统最终会打包为Docker镜像，可以方便地在Docker环境中部署及使用。

## 2.2 keras

Keras是使用Python编写的神经网络的API，它能够以TensorFlow作为后端进行运行，Keras重点支持快速实验的开发，可以用最少的时间把想法转换为实验结果。

Keras完全模块化设计，由模块构成序列或图，这些模块可以自由地组装在一起，包括神经网络层、损失函数、优化器、激活函数、正则化方法、初始化方法等，他们都可以通过模块化来构建新模型。

## 2.3 music21库

Music21是一个由MIT开发的具有强大功能的计算音乐学分析python库，相较于pretty\_midi库只能处理midi文件，music21可以处理包括MusicXML，MIDI，ABC等多种格式的音乐文件，并且可以通过代码对音乐文件进行构建和分析。

人们使用music21计算机来回答音乐学中的问题，研究大型音乐数据集，生成音乐实例，教授音乐理论基础，编辑音乐符号，研究音乐和大脑以及创作音乐（无论是算法上还是直接上）。该系统自2008年以来一直不断更新与扩展。

## 2.4 midi与musicXML文件格式

Midi与musicXML格式为常见的两个计算机音乐存储格式。

MIDI(Musical Instrument Digital Interface)，是20世纪80年代为了解决电声乐器之间的通信问题而提出的，是编曲界最广泛的音乐标准格式，其可以记录每个音符的包括起始终止位置、力度、音高等属性；只需在宿主软件添加所需要的音色库就可以模拟出MIDI文件所记录的乐谱，故又被称为“计算机能理解的乐谱”。MIDI格式的出现也标志着计算机编曲进入字符化时代。

MusicXML（Music Extensible Markup Language，音乐扩展标记语言）是一种基于XML的音乐符号文件格式。由Reccordare公司所开发，其被设计用来作为乐谱信息的交换格式，尤其是在不同乐谱显示软件之间切换。它兼容性高、包含属性多、结构简单。

## 2.5 Django框架

Django是用python开发的一个免费开源框架，可以用于快速搭建网站，采用MVC（模型、视图、控制器）模式，如图2-1。

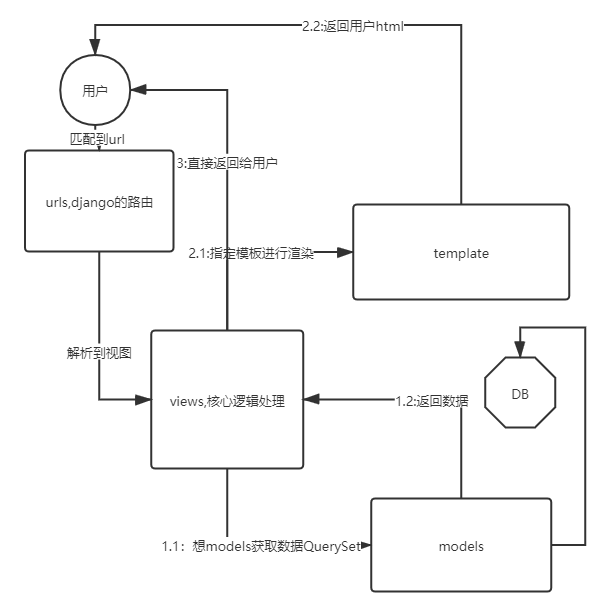


图2-1 Django基本流程

借鉴MVC模式，Django框架的设计分为models、template、views三个部分。

Models：负责业务对象。

View：负责业务逻辑，调用Model与Template。

Template：负责将页面展示给用户。

另外，Django还有一个urls分发器，它控制url页面请求，将其分发给不同view处理。

HTML（Hyper Text Markup Language），是一种用于创建网页的标记语言，由浏览器负责解释；Template可以使用html来进行编写。

CSS可以用来表现HTML文件样式的语言；其不仅可以修饰网页，还可以动态对网页元素进行格式化。

JavaScript是一种即时编译型的语言，它常常被嵌入在HTML文件或被HTML文件调用，其函数功能可以在静态页面实现很多动态功能。

# 第三章 基于LSTM的和弦编配模型

## 3.1 模型的选择

首先对任务进行分析：因为和弦对上下文具有极大的依赖性，所以应当选择具有记忆单元的模型，且该模型可以基于现有小节旋律音符和所有之前的音符，预测本小节的旋律重音。

基于任务的特性，研究选择了适于处理与时间序列高度相关的问题的LSTM模型。

### 3.1.1 LSTM

传统的神经网络并不含有对序列中前部分的记忆，这在处理特殊问题时是一个缺陷。而递归神经网络（Recurrent Neural Network，简称RNN）解决了这个问题，其是具有循环的神经网络，它可以控制信息的存在与遗忘。

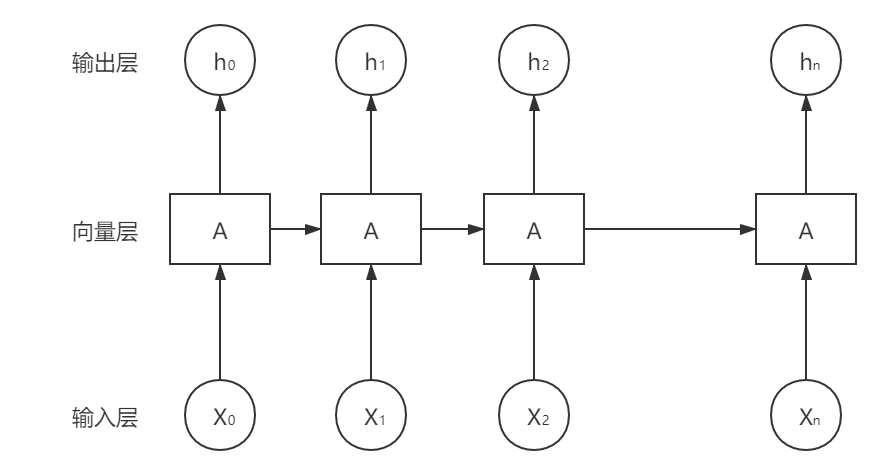


图3-1 RNN模型示意图

递归神经网络特别适用于与序列密切相关的任务，它是神经网络表达这类数据的自然架构。近些年中，RNN在很多应用场景中获得了巨大成功，如：语音识别、文字翻译、智能对话等。而本研究所使用的长短期记忆网络（Long Short-Term Memory，简称LSTM），是一种非常特殊的并且经过改进的递归神经网络：

随着跨度的增大，RNN会很容易出现梯度消失或者梯度爆炸，无法学会如何连接序列中有用的信息；而LSTM继承了RNN的链状结构，改进了它的重复模块，使用了一种新的交互模式。

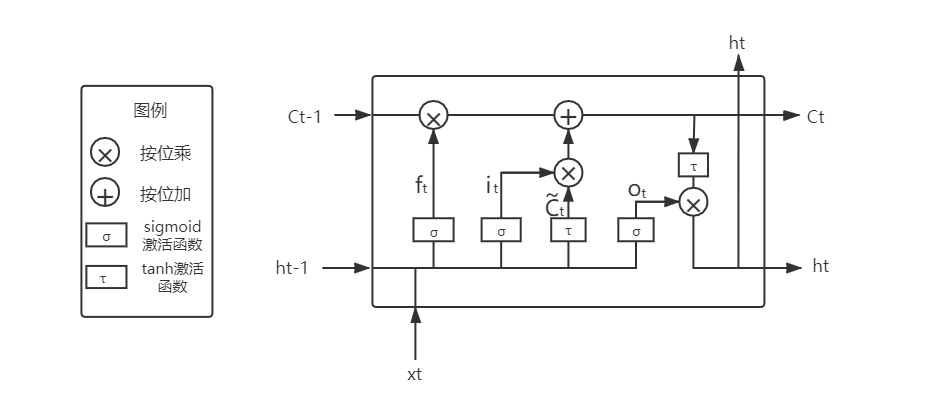


图3-2 LSTM单元示意图

上图中的箭头线段意义如下：

与为单元状态，它沿着如箭头所示的链运行，信息沿此流动。

门由sigmoid激活函数与按位乘法运算组成，其中箭头表示向量传输。LSTM有三个类似的门控制单元状态。

合并箭头表示级联。

有多指向的箭头表示复制。

对于以上示意图，存在下描述表达式：

式（3-1）

式（3-2）

式（3-3）

在LSTM中，核心部分就是三个门的组成。其中式（3-1）为遗忘门层（forget gate layer），输出为0-1之间的数，表示对前状态的保留程度。式（3-2）表示输入门层，决定选择将要更新的值。式（3-3）为输出门，基于单元状态决定输出的内容。

### 3.1.2 Attention机制

LSTM存在一个问题：输入序列全部都会编码为一个固定长度的向量表示，而解码受限于该固定长度的向量表示；当输入序列比较长的时候，模型的性能会变得很差。在研究中仍有另外一个不可忽略的因素：传统LSTM对每一个向量权重是相同的，然而在实际音乐编配中，每一小节和弦的编配结果对其上下文旋律的依赖权重不同，所以我们引入Attention机制。

在文本翻译任务中，使用attention机制的模型每生成一个词时都会在输入序列中找出一个与之最相关的词集合。之后模型根据当前的上下文向量和所有之前生成出来的词来预测下一个目标词[6]。即便使用attention机制后会增加计算量，但可以提升性能和准确性；在我们生成小节旋律重音时，attention机制将源旋律中的位置信息作为更重要的属性，符合音乐的一般特征。

## 3.2 数据预处理

在进行模型的构建过程中，第一步是对需要训练的数据进行必要的处理以供训练。

### 3.2.1 数据集的选择

根据研究任务的特性，数据集应当具有以下特征：

1.具有小节属性。

2.可以定量表示音高、音长。

3.旋律音符与和弦音符可以得以分开。

首先选择了诺丁汉midi数据集（695个midi文件格式的民间音乐集合），使用MIT的music21库对此数据集中的旋律与和弦进行分离。在后期实现过程中，由于midi格式的音符独立性，和弦中的每一个音符在midi格式中都被独立标记而并非如musicXML格式有独立和弦格式。

后经过查找，确定了来自MARG（music&audio research group）的CSV格式的数据集，如图3-3。

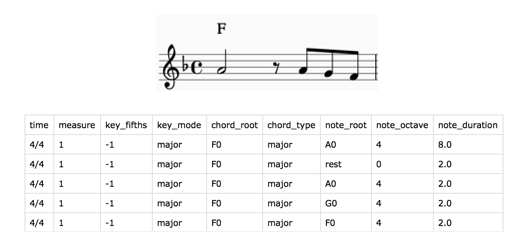
[](http://marg.snu.ac.kr/chord_generation/)

图3-3 MARG数据集样例

此数据集使用了Wikifonia.org提供的关系型数据库，该网站在2013年已经停止了服务，但在停止之前仍有一些数据得以保留。其中包括MusicXML格式的西方音乐旋律，包含摇滚、流行、乡村、爵士、R&B、儿歌等格式。从获得的数据库中，数据集收集了2252个csv文件，他们包含主要旋律，并且每个小节都有一个和弦。

图2-3显示了从单个文件中提取的特征示例，数据集从关系数据库中收集了拍号、小节数、调音键程、和弦以及音符；这些特征通过连接行来表示在矩阵中，这些行分别代表各个音符的音乐特征。在“note\_duration”中，四分音符的持续时间为4.0。

### 3.2.2 数据清洗及格式转换

在对诺丁汉数据集的预处理过程中，主要包含midi文件与csv文件之间的格式转换，这里主要使用MIT的music21库：

1. 由midi转换至csv：读取midi文件，使用库函数对文件中每个音符进行音高比对，最终获得音符音高、音长序列。
2. 由csv转换至midi：对csv文件进行分行读取，写为musicXML乐谱文件格式，再将musicXML文件丢弃一些乐谱元素获得midi文件。

在对MARG数据集的预处理过程中涉及数据清洗：

1. 对于乐曲末尾的和弦不存在的情况选择使用NaN字段对其补全。
2. 对乐曲每小节“note\_duration”和进行统计，若有缺失则用休止符补全。
3. 对于众多繁杂的和弦种类，选择进行简化，分为大小调+旋律音，便于预测，如图3-4。



图3-4 和弦简化示意图

### 3.2.3 音符记号与字符之间的转换

由于调式的关系，所以乐谱记号中的音调不能直接和音高一一对应，根据五度十二律，需要进行调式转换来将所有音符映射到C大调以便训练。

对数据集中的字符格式进行统一化，如：C#与Db、A#与Bb；即使在音程乐理中二者并不完全相同，但在数据预处理中为了便于训练还是对其进行统一。

除此之外，还对数据集进行了向npy格式的转换，训练及预测效率可以获得提高。

## 建立模型

通过对任务的提前评估确定了模型架构，选择了python中的开源框架Keras进行模型的建立训练及预测。

### 3.3.1 参数选择

目标函数选择多分类的对数损失函数，优化器选择Adam算法，通过朴素贝叶斯优化，在已经建立的LSTM+attention层的神经网络模型上自动选择参数，模型图如图3-5、图3-6。

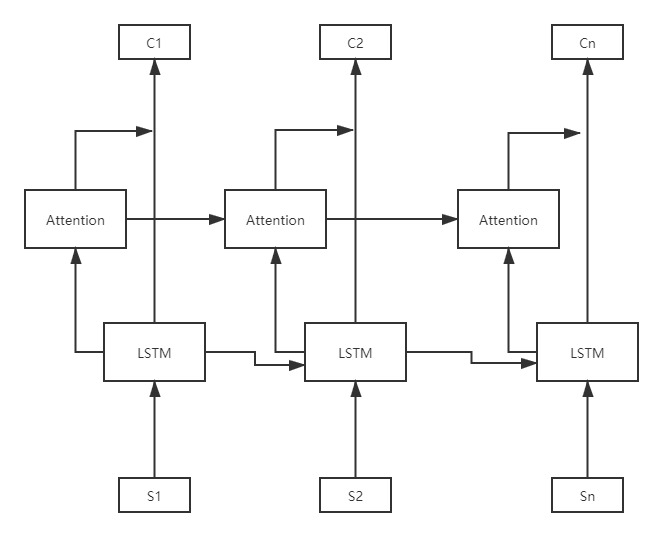


图3-5 模型示意图

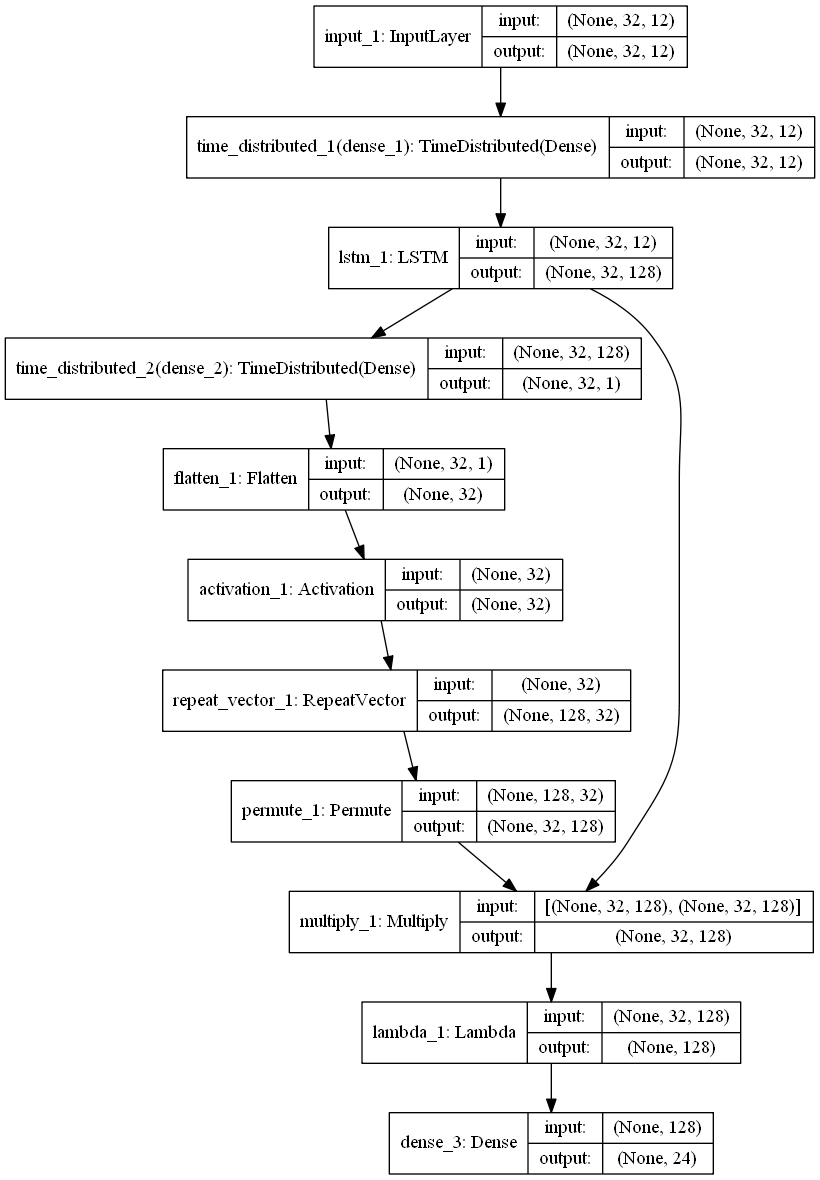


图3-6 模型图

该模型本质上是一个可以自学习上下文权重多分类器，输入为以小节为单元的旋律音符的编码，输出为与小节数相同数目的旋律重音编码；通过每一小节的旋律重音在下一节中获得和弦。

### 3.3.2 测试集预测结果分析

在epoch为1的情况下，使用提前划分的测试集进行测试，模型准确率可以达到73%，考虑到和弦本就是可以有变式及转调，这个准确率是可以接受的；此外，也可以通过增加epoch来提高准确率，经过实测，将epoch设置为100时准确率可以提升至78%左右。

### 3.3.3 实际演奏分析

程序可以通过将传入的midi文件转为csv文件后，对旋律重音进行预测，获得与旋律小节相同长度的旋律重音字符串。根据旋律重音，我们便很容易将柱式和弦对其编配，现在对欢乐颂进行预测，将已经编配的曲子与原曲进行对比，见图3-7 图3-8。

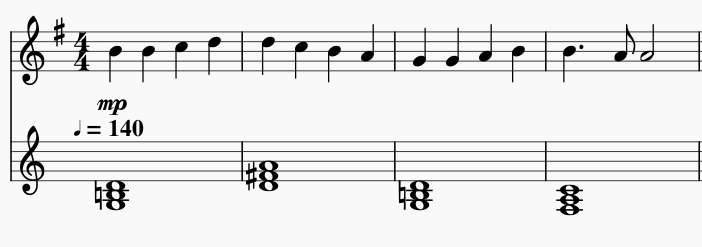


图3-7 生成的《欢乐颂》前四小节和弦



图3-8 原曲《欢乐颂》前四小节和弦

如图所示，其中有分歧的为第二小节与第四小节。在第二小节中，D和弦分解应该是“D F# A”，所以第1转位以F#为根音的话，就变成“F# A D”；在第四小节中，由于训练模型中只有音高没有音长，相当于每个音权重相同，并且每一个小节和弦相同（这是合理的）。若出现多个和弦则会预测错误，预测添加和弦进行规则可以减弱影响。所以，这个和弦生成的系统中模型部分是可用的。

## 3.4 本章小结

本章介绍了基于LSTM的预测旋律重音模型，介绍了数据集的选择及预处理，实现了LSTM+attention的模型搭建并通过朴素贝叶斯进行参数的选择；最终模型可以通过旋律进行旋律重音的预测并检测可用，为下一步生成和弦打下基础。

# 

# 第四章 和声学规则的限制

在上一章中获得了旋律重音字符串，若需要获得和弦还需要音乐人去进行手动编配，这之中主要依赖编曲人的乐理知识以及编曲经验；本章会在乐理方面对旋律重音进行自动编配和弦。

## 4.1 旋律调性判断

音乐人的一项基本训练为视唱练耳：通过不断地进行听音训练来寻求相对音感的能力；第一步就是给定旋律调性[7]。一般情况有一个简单方法：用首调判断。一般在听到一段旋律后，头脑中就会形成一个首调概念，并且立刻得知是大调还是小调；再当旋律进行到首调的1这个音时，用标准音去比较，即可得出是什么大调。

使用计算机判断旋律调性也与此方法相似并且更加简单：在比较音高时无需使用相对音感再与标准音比对，因为在对midi文件处理之时就已经全部标记为固定调，只需要获得预测结果字符串中的第一小节的主音，确定主音音高即可判断调性。

在调音软件《Auto Tone》中，有一项自动功能即判断调性以便在之后调音工作中自动对齐音准，其中原理与此类似。

图4-1为旋律调性判断流程图。

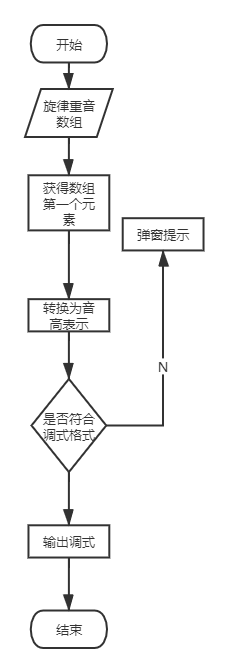


图4-1 旋律调性流程图

## 4.2 和弦行进套路及风格化

在和弦编配的过程中，和弦走向是很重要的一种特性；在人工智能编配和弦的项目中，人们总会感觉到和弦都很和谐但却总有推进不下去的感觉；这就是因为其和弦虽然合法但是并没有一些行进套路，并没有按照为解决到解决、冲突到平稳这样的过程；本系统在和弦选择时就添加了常见的和弦进行规则来推进和弦解决过程。

### 4.2.1 常见的和弦行进规则

在本研究中，主要探讨大调和弦，因为在现代音乐中大小调和弦公用的现象十分普遍，不便用和声学的解释。每个大调有七个基本和弦，如图4-2。

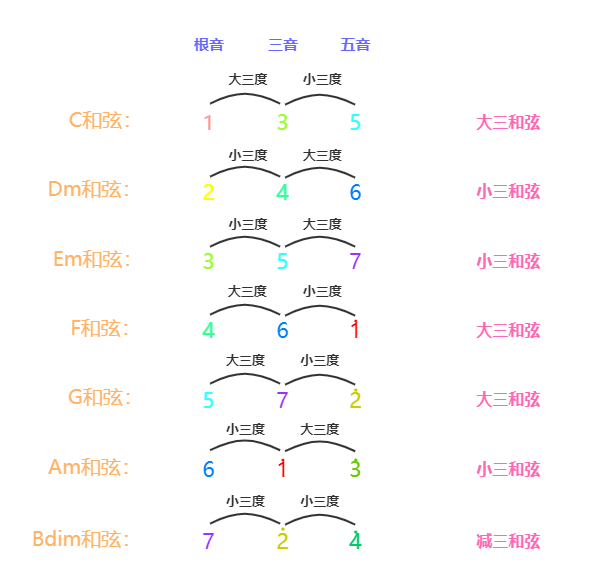


图4-2 C调的各级基础和弦

图3-1所示为C调的 I、II、III、IV、V、VI、VII 级和弦；其中 I、IV、V 被称为正三和弦， II、III、VI 被称为副三和弦。I级和弦也称为主和弦，IV级和弦也被称为下属和弦，V级和弦也成为属和弦，每一级和弦都有其自身功能[8]：

I级和弦：明确调性，和弦感觉洪亮，但多用会感觉生硬。

II级和弦：风格柔和，其在属和弦即V级和弦之间，而V级和弦则自然要回到I级和弦，所以比较容易形成II-V-I的进行，并且经常被用在旋律终结处。

III级和弦：风格柔美悲伤，在部分使用I级和弦时候可以考虑替换为III级和弦来避免生硬的问题。

IV级和弦：正三和弦，属于骨干和弦；和弦明亮空旷；在I级和弦后跟IV级和弦或III级和弦是由旋律风格决定。

V级和弦：对主和弦起支撑作用，对乐曲中止感起重要作用，是流行音乐的明显特色。

VI级和弦：中性和弦（可看作小调的主和弦，歌曲风格忧郁），在大调音乐中起连接作用，它可以在几乎各级和弦中进行连贯；I-VI-IV-V 就是机器常见的进行；部分流行乐曲也仅有这四个和弦。

VII级和弦：在流行乐中使用比较少，减三和弦自然就有向内收缩感，只在特定行进用。

首先我们需要根据调性来确定该调各级和弦名称，比如当计算机确定旋律为C大调即基本调，则确定各级和弦为C、Dm、Em、F、G、Am、Bdim；而当计算机确定旋律为G大调时，各级和弦在编码时分别加上相差的音程即可获得各级和弦在基本调的名称为：G、Am、Bm、C、D、Em、F#dim，其他调式同理，均可由程序自动转换，名称如图4-3。

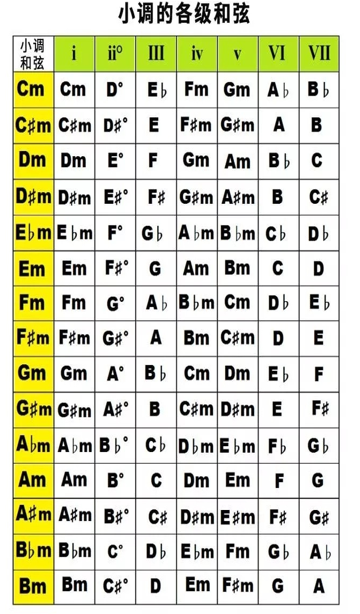
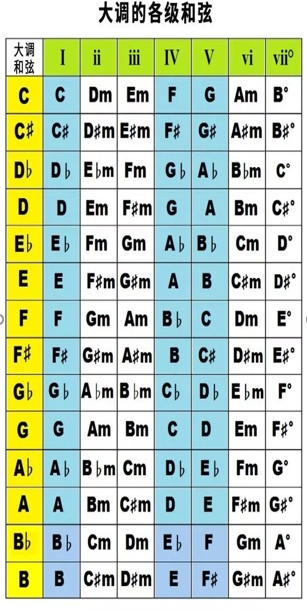


图4-3 所有调式基本和弦名称

将各级和弦与和弦名称一一对应后，便可以在任意旋律上使用“I、II、III、IV、V、VI、VII”来标记和弦而不存在调式问题，就可以很方便地编配和弦了。在编曲软件《Cubase》中，有一项自动功能即根据和弦级数给出关联和弦，其中原理与此类似。

和弦编配的前提是和弦音组中应当包含旋律主音（少数离调和弦或和弦代替在后文讨论），根据上一章中提供的每一小节旋律主音所组成的字符串，很容易地列出所有符合这一基本乐理法则的所有和弦。

首先编配和弦解决部分（一般位于一段旋律的结束），在一段完整的旋律中，主和弦的稳定性最强，和弦解决的一般方法为就近解决到主和弦上，而例如C大调，最不稳定的是F和B音，因为只有这两个音与其后面的音是半音关系，所以倾向性很强烈，所以F一定会解决到G，B一定会解决到C；该规则映射到本系统中，程序检测重音字符串末尾三个字段内七和弦是否合法，若七和弦合法则自动判断是否可以进行和弦解决，解决成功则确定旋律解决部分和弦。

其次是旋律开始部分，第一个和弦基本为主和弦，如C调第一个和弦为C（除直接进入副歌的歌曲），在本系统中，重音字符串第一个音就为主和弦。

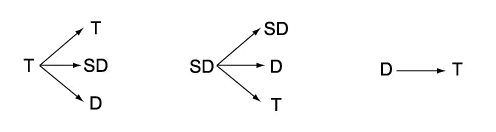


图4-4 调内和弦的进行规则

最后完成旋律主体部分的和弦编配，如图4-4，总体规则基本分为：  
 1. 从主和弦开始：从主和弦可以进行至任何和弦上，即（T->T）

2. 从下属和弦开始：如IV->VII的进行为从下属和弦到属和弦，IV->I型为下属和弦向主和弦的进行；同样的也有下属和弦的连续进行方式。

3. 从属和弦开始：此即为上文中的和弦解决。

在音乐人的编曲习惯中，掌握基本的和弦进行原则后会将T、SD、D按照以上原则进行排列[9]。而在本系统中，内置了多种经典流行古典音乐和弦走向，如图4-5。



图4-5（1） “1645”和弦套路

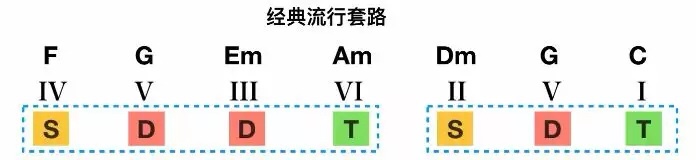


图4-5（2）“4536251”万能和弦

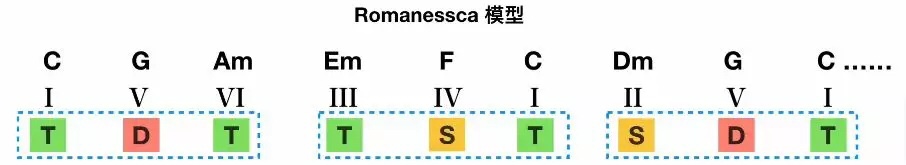


图4-5（3）“Ronanessca”经典古典套路

在得到的每一小节所有合法的和弦级数的数组中，系统使用了动态规划来自动匹配这些规则，如图4-6，尽量多地使和弦走向“套路化”，这也是使和弦听起来有流向的解决办法。

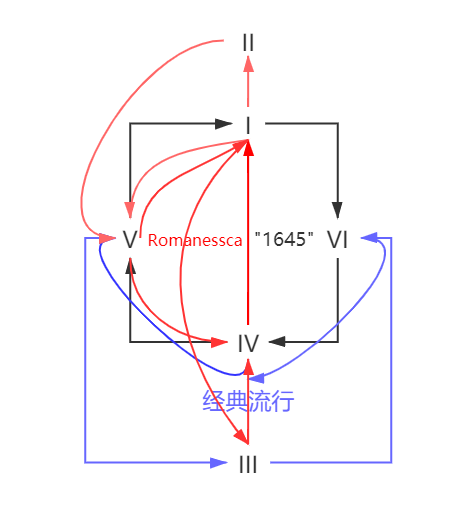


图4-6 系统内置和弦进行

### 4.2.2 对套路自定义修改实现风格化

在系统中所有规则全部写在单独的数据结构如字典等，所以除去在代码中内置的这些规则外，若有需要添加或修改的规则可直接在数据结构中进行修改而无需改动代码。

而另一个方面的自定义则为分解和弦的编配[10]，在上文中，系统得到了每一个小节的和弦级数，很容易地就可以为每一个小节编配柱式和弦；而柱式和弦明显是单一且无趣的，这时候就需要一些分解和弦来增加和弦的表达方式；本系统参考苹果公司的库乐队软件中的“自动编曲”功能，该功能通过选择和弦名称以及速度就可以输出分解和弦，如图4-7。

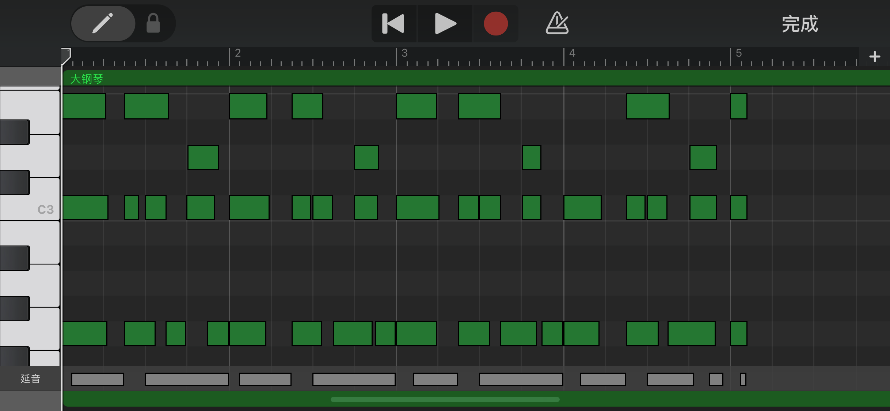


图4-7 库乐队中选择Cmaj和弦输出的midi卷轴

本系统同样内置三种和弦类型选择，只需在命令行选择分解和弦的类型，系统会自动根据旋律的节奏来拉伸或者裁切选择的分解和弦类型，并将此类型直接运用到整首曲子中，过程完全在后台运行；同样的，分解和弦类型以及乐器类型都可以在单独的数据结构中方便修改。

## 4.3 和弦替代

在爵士等音乐即兴演奏或编曲中，用不同的调式音阶代替了原来的和弦，产生了更加丰富的效果

在自然音乐中，音乐和弦多种多样；而在训练集的预处理中本文将所有和弦替换为旋律音方便训练；上文也全部是在调内七个基本和弦上进行编配，不免有些单调。而实际上常见和弦可以被很多其他和弦所代替，在编配基本完成后就可以根据旋律自身特性来进行和弦替代，以下是一些常见替代：

1. 所有的属和弦都可以用它的降五音的属和弦或dim7来替代，如属七和弦可以用降G属七和弦替代。
2. 五级和弦可以用降七级来替代，如基本调中的G和弦可以用Bb来替代。
3. 六级用六级的降五音来替代，如基本调中A和弦可以用Eb和弦来替代。

这些常见的和弦替代在多种乐曲中被广泛替代，和弦本身及其替代并无对错之分，在本系统中可以通过修改和弦级数名称来对和弦进行整体替代，也可在生成乐曲后对midi文件进行二次修改，图4-8为流程图。

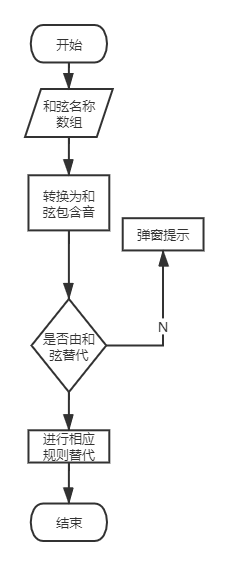


图4-8 和弦替代流程图

## 4.4 效果分析

在命令行中可以简易地选择上传文件的位置、选择模型、选择希望的演奏方式（分解和弦）等，在代码中独立出的数据结构可以方便对和弦进行规则、和弦代替方法进行修改；系统会将输入的midi文件通过第二章所述技术编码为csv文件并传入模型进行预测。预测获得每一小节的旋律重音组成的字符串在经由调性判断后，通过多种和弦进行规则对合法和弦进行约束，直到留下一条确定的和弦级数行进路线。通过和弦级数系统对应相应调性的和弦名称并使用选择的演奏方式进行编配和弦，效果如图4-9。



图4-9 《欢乐颂》选择自动编配分解和弦第一小节效果图

由于模型准确率及存在离调和弦的原因，在和弦编配过后与原曲往往差距不可忽视，但要注意的是，和弦编配属于技巧，它并没有一个最正确的答案。检验一段和弦编配是否合理的唯一办法就是试听，在上述版本《欢乐颂》中，并没有出现明显离调、转调、不和谐的听觉感受；而因为旋律为4/4拍，当选择3/4拍的分解和弦节奏型时，本系统为了乐谱合法性对分解和弦进行了等倍拉伸，会出现重音未对齐的现象，所以在使用时应当选择与旋律匹配或有倍数关系的分解和弦；若没有内置该节奏，则可进行自定义。

## 4.5 本章小结

本章通过模仿音乐人的习惯，实现了由旋律重音向和弦的自动编配：以文件格式转换、模型训练输出为前提，本章主要在乐理方面对和弦进行限制；通过模型输出的旋律重音可以获得每一小节几乎所有的“合法的”和弦，然后依次使用各种规则、代替加以限制，最终输出和弦名称；最后在生成midi文件时选择合适的分解和弦进行编配。最终在上传系统中只需要传入文件、选择参数选项，就获得了一个由旋律、和弦组成的两声部midi文件。

# 

# 第五章 系统设计

在上文中已经展示了模型训练、预测及传统乐理限制部分，在五、六章将会主要描述提高系统易用性上的努力，本章主要介绍系统架构所需技术以及功能性及非功能性设计。

## 5.1 功能性需求

功能需求是系统必须提供的能力和必须遵从的条件。

本软件系统基于对传入midi文件的处理以及对模型生成midi文件的分析展示，需要做到：

1. 后台可以对midi文件进行模型预测、并对和弦名称数组进行输出。
2. 文件上传功能。
3. 参数选择功能。
4. 后台可以根据用户选择的演奏方式进行midi文件生成。
5. 前端页面可以播放后端输出的midi文件。
6. 前端页面可以对midi文件进行可视化以便进行二次调整。
7. 下载midi文件功能。

通过总体用例图，可以更好地把用户、系统以及各功能展现出来，体现总体需求，如图5-1。

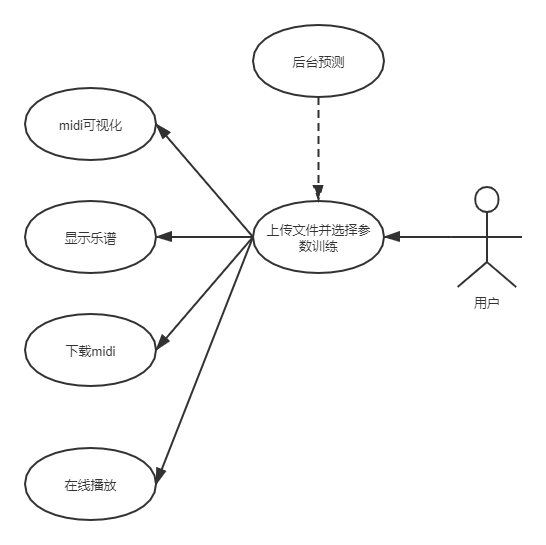


图5-1 系统总体用例图

如上图所示，用户可以在前端界面进行上传文件并且选择自己想要的模型和演奏方式，这些功能都需要服务器将midi文件转为csv/npy文件进行训练，并在乐理规则限制下得出和弦名称。用户可以选择需要展示的乐谱、或者在线播放midi文件并且进行可视化，也可以选择对midi文件进行下载进行二次修改。

### 5.1.1 旋律文件上传功能

表5－1描述了上传旋律文件的用例规约，给出了上传旋律文件的基本事件流程，扩展事件流程，除了说明功能性需求外，还描述了非功能性需求。

表5-1 旋律文件上传用例规约

|  |  |
| --- | --- |
| 用例规约 | |
| 1.用例名称 | 上传旋律文件 |
| 2.简要说明 | 无 |
| 3.基本事件流 | 1.用户进入主界面。  2.用户点击上传文件按钮。  3.弹出选择本地文件窗口。  4.用户选择文件并单击上传。  5.系统对文件进行格式检索。  6.系统上传文件并保存。 |
| 4.扩展事件流 | 步骤5中若检测格式错误，则不可上传。 |
| 5.非功能需求 | 界面美观，操作简单。 |
| 6.前置条件 | 登录网站。 |
| 7.后置条件 | 操作完成后，在主界面显示midi文件名称表示上传完毕 |
| 8.扩展点 | 无 |
| 9.优先级 | 高 |

### 5.1.2 模型选择功能

表5－2描述了用户模型参数选择的用例规约，给出了用户参数模型选择的基本事件流程，扩展事件流程，除了说明功能性需求外，还描述了非功能性需求。

表5-2 模型参数选择用例规约

|  |  |
| --- | --- |
| 用例规约 | |
| 1.用例名称 | 模型参数选择 |
| 2.简要说明 | 无 |
| 3.基本事件流 | 1.用户选择模型文件。  2.用户选择模型权重。  3.用户选择演奏方式。  4.用户单击上传。 |
| 4.扩展事件流 | 步骤5中若检测有未选择选项，则弹窗提醒。 |
| 5.非功能需求 | 界面美观，操作简单。 |
| 6.前置条件 | 用户已经上传midi文件 |
| 7.后置条件 | 操作完成后，系统将选择的参数传入后台。 |
| 8.扩展点 | 无 |
| 9.优先级 | 高 |

### 5.1.3 后端预测功能

表5－3描述了后端预测的用例规约，给出了后端预测的基本事件流程，扩展事件流程，除了说明功能性需求外，还描述了非功能性需求。

表5-3 后端预测用例规约

|  |  |
| --- | --- |
| 用例规约 | |
| 1.用例名称 | 后端预测 |
| 2.简要说明 | 无 |
| 3.基本事件流 | 1.midi文件传入。  2.用户选择模型权重。  3.用户选择演奏方式。  4.用户单击上传。  5.系统将midi文件转为csv。  6.系统将csv文件传入模型进行训练。  7.系统将输出和弦字符串编配为midi文件。 |
| 4.扩展事件流 | 步骤5-7中若出错，则弹窗提醒。 |
| 5.非功能需求 | 界面美观，操作简单。 |
| 6.前置条件 | 输入midi文件及参数全部正确。 |
| 7.后置条件 | 编配完成后将输出的midi文件保存，主界面显示乐谱展示、播放、下载按钮。 |
| 8.扩展点 | 无 |
| 9.优先级 | 高 |

### 5.1.4 乐谱展示功能

表5－4描述了乐谱展示的用例规约，给出了乐谱展示的基本事件流程，扩展事件流程，除了说明功能性需求外，还描述了非功能性需求。

表5-4 乐谱展示用例规约

|  |  |
| --- | --- |
| 用例规约 | |
| 1.用例名称 | 乐谱展示 |
| 2.简要说明 | 无 |
| 3.基本事件流 | 1.用户单击主界面的乐谱按钮。  2.JavaScript解析后台midi文件。  3.跳转乐谱显示界面。 |
| 4.扩展事件流 | 步骤3中若出错，则弹窗提醒。 |
| 5.非功能需求 | 界面美观，操作简单。 |
| 6.前置条件 | Midi文件拥有合法的乐谱格式。 |
| 7.后置条件 | 无。 |
| 8.扩展点 | 无。 |
| 9.优先级 | 中。 |

### 5.1.5 在线播放功能

表5-5描述了在线播放的用例规约，给出了在线播放的基本事件流程，扩展事件流程，除了说明功能性需求外，还描述了非功能性需求。

表5-5 在线播放用例规约

|  |  |
| --- | --- |
| 用例规约 | |
| 1.用例名称 | 在线播放 |
| 2.简要说明 | 无 |
| 3.基本事件流 | 1.用户单击主界面的播放按钮。  2.JavaScript解析后台midi文件。  3.对midi文件进行在线播放。 |
| 4.扩展事件流 | 步骤3中若出错，则弹窗提醒。 |
| 5.非功能需求 | 界面美观，操作简单。 |
| 6.前置条件 | 文件已经成功上传并预测，Midi文件能够正常播放。 |
| 7.后置条件 | 无。 |
| 8.扩展点 | 无。 |
| 9.优先级 | 中。 |

### 5.1.6 播放可视化功能

表5-6描述了播放可视化的用例规约，给出了播放可视化的基本事件流程，扩展事件流程，除了说明功能性需求外，还描述了非功能性需求。

表-6 播放可视化用例规约

|  |  |
| --- | --- |
| 用例规约 | |
| 1.用例名称 | 播放可视化 |
| 2.简要说明 | 无 |
| 3.基本事件流 | 1.用户单击主界面的播放按钮。  2.JavaScript解析后台midi文件。  3.跳转卷轴可视化界面。 |
| 4.扩展事件流 | 步骤3中若出错，则弹窗提醒。 |
| 5.非功能需求 | 界面美观，操作简单。 |
| 6.前置条件 | 文件已经成功上传并预测，Midi文件中不存在音符重叠等错误。 |
| 7.后置条件 | 无。 |
| 8.扩展点 | 无。 |
| 9.优先级 | 中。 |

### 5.1.7 文件下载功能

表5-7描述了文件下载的用例规约，给出了文件下载的基本事件流程，扩展事件流程，除了说明功能性需求外，还描述了非功能性需求。

表5-7 文件下载用例规约

|  |  |
| --- | --- |
| 用例规约 | |
| 1.用例名称 | 文件下载。 |
| 2.简要说明 | 无。 |
| 3.基本事件流 | 1.用户单击主界面的播放按钮。  2.JavaScript解析后台midi文件。  3.跳转卷轴可视化界面。 |
| 4.扩展事件流 | 步骤3中若出错，则弹窗提醒。 |
| 5.非功能需求 | 界面美观，操作简单。 |
| 6.前置条件 | 文件已经成功上传并预测，Midi文件中不存在音符重叠等错误。 |
| 7.后置条件 | 无。 |
| 8.扩展点 | 无。 |
| 9.优先级 | 中。 |

## 5.2 非功能性需求及其体现

除了功能性需求外，还需要在其他场景进行评估。以下通过非功能性需求进行评估，包括系统性能、易用性等、可拓展性、可移植性等。

### 5.2.1 系统性能

因为需要在传入文件并且选择参数完毕后才能进行训练，并写入新的midi文件，所以获得新midi文件需要等待一定时间，所以网页的跳转并不是即时的，限定时间最多应为30秒。

### 5.2.2 可靠性

将项目部署由于服务器之上，在浏览器中输入urls路由中写好的网址即可访问网页；没有注册登录等密码保护操作，可靠性高。

### 5.2.3 可移植性

本系统基于Django框架搭建，部署为Docker镜像，可以任意Docker环境部署，可移植性较强。

### 5.2.4 可拓展性

本系统基于Django框架搭建，此框架基于MVC，耦合程度低；若增加模块对代码改动较小，系统拓展能力强。

### 5.2.5 易用性

系统界面简单、操作容易上手、网页美观简洁、跳转逻辑清晰，便于用户进行操作。

## 5.3 系统架构设计

本系统基于用户使用角度出发，为了用户可以获得更舒适的体验和只需进行更简单的操作，系统应当设计为简洁外观及步骤、功能划分独立，以便开发人员可以快速轻松地修改代码块以方便代码的迭代工作。

### 5.3.1 系统总体功能架构

本系统采用Django框架，基于python语言。神经网络主要使用keras库调用算法，是现在主流的机器学习框架。前端对midi文件的可视化及乐谱的展示使用JavaScript，该语言可以很容易地嵌入在前端HTML语言中。

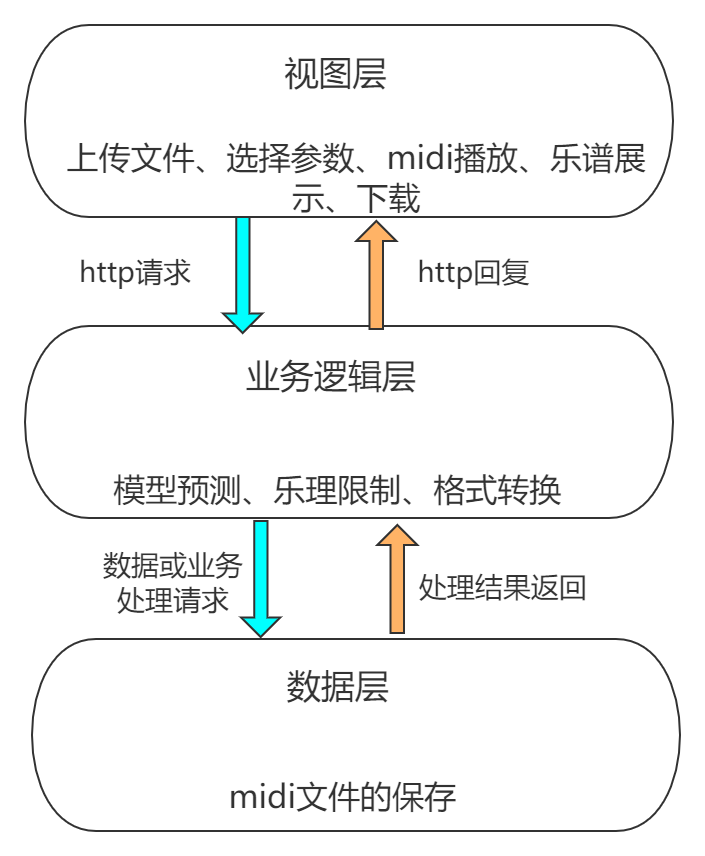


图5-2 系统总体功能架构图

如上图所示，本系统可以抽象为数据层、业务逻辑层和视图层。

在数据层中，服务器本地接收前端传输的midi文件、模型生成的和弦、后端输出的带和弦的midi文件。系统通过路径与文件名对所需文件进行检索并按需传入业务逻辑层。

在业务逻辑层，主要有模型预测、乐理限制、格式转换三个功能模块。在收到http的post请求之后，接收html传入的文件并保存至数据层；接收选择的选项传入LSTM模型，得到重音数组；获得重音数组进行乐理规则的限制，得到和弦名称数组；获得和弦名称数组，进行格式转换得到midi文件并再次存入数据层，最后回复视图层进行功能页面的跳转。

视图层包括上传文件、选择参数、midi播放、乐谱展示、下载等功能。用户可以在主页对文件进行上传、对各种个性化选项进行选择，在收到业务逻辑层的回复后页面显示功能：midi播放、可视化、展示乐谱、下载等。

### 5.3.2 系统输入输出

由于文件传输简单且不存在并发需求，本系统并未单独设置数据库而是直接存在服务器本地，midi文件及中间文件的保存均在指定相对路径下保存、调用及删除。

### 5.3.3 异常处理设计

在前端界面中，对传输文件格式进行检查，对于后缀除mid/midi外的文件不予接收并报错。

在前端界面中，对参数选择进行检查，对于有未选择参数的情况，在post后提示报错。

在后端处理midi文件时，通过music21库对midi文件进行检查，若未能转换为csv文件则在前端页面提示报错。

在后端模型预测时，通过keras库对npy文件进行预测，若输出数组不合法则在前端页面提示报错。

在前端乐谱展示时，若midi文件不符合乐谱编配规则则乐谱不显示。

在前端midi播放时，若MIDI.js库中未有音色则使用默认钢琴音色。

## 5.4 本章小结

本章讲述了系统设计，包括功能性设计、非功能性设计、系统概要设计；明确设计原则及思路，确定整体架构，然后从系统输入输出、异常处理出发，详细描述了系统的设计思路。

# 

# 第六章 系统实现与测试

上文描述了对系统的需求分析，本章是对需求分析、概要设计进行在开发方面的更为详细的描述，对各功能模块进行具体技术方法实现的描述。

## 6.1 首页选择模块设计实现

对于选择模块，前端HTML用于接收选项内容，用户点击button后向view层传输一个post命令，通过url路由调用upload\_file()方法以供接下来使用，与文件一同组成upload\_file()方法的实参，在参数数量不足或者不合法的情况下使用request()方法进行弹窗报错，图6-1为首页选择模块的流程图。

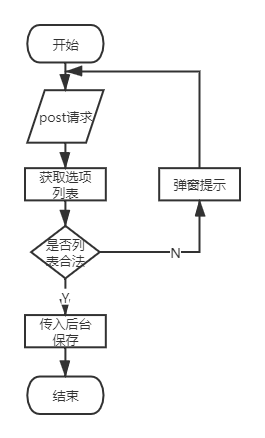


图6-1 首页选择模块流程图

## 6.2 文件上传及后台预测模块设计实现

### 6.2.1 文件上传模块设计实现

对于文件上传模块，前端HTML实现对文件的选择按钮、弹出本地文件选择框、用户选择后将文件名展示在选择按钮旁侧；用户点击button后向view层传输一个post命令，通过url路由调用upload\_file()方法以供接下来使用，与选择过的参数一同组成upload\_file()方法的实参，在文件后缀不合法时，使用request()方法在前端弹窗报错；若文件合法则保存在服务器指定相对路径，图6-2为文件上传模块的流程图。

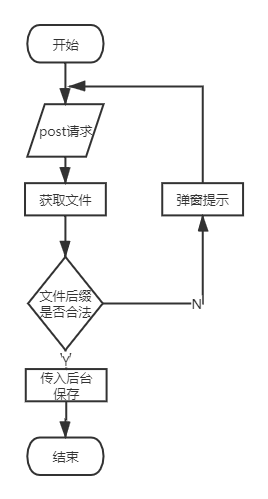


图6-2文件上传模块流程图

### 6.2.2 预测模块设计实现

View层中upload\_file()方法传入midi文件，利用music21库中的parse()函数对midi文件进行读取和解码，将midi音符按照顺序存入csv文件中并按照小节分组。对于得到的csv文件使用make\_npy文件进行格式转换获得npy文件，将该npy文件传入用户选择的模型中进行预测，获得旋律重音数组，系统会判断该数组是否合法（长度是否与小节数一致、编码是否符合下一步读取规范），若非法则进行弹窗提醒用户修改。将获得的npy文件使用make\_chord()函数进行乐理限制，包括和弦级数转换、旋律调性确认、和弦走向匹配、级数名称转换等操作，最终获得和弦名称的数组，系统会判断该数组是否合法（长度是否与小节数一致、编码是否符合下一步读取规范），若非法则进行弹窗提醒用户修改，若合法则传入程序的栈中。图6-3为预测模块流程图。

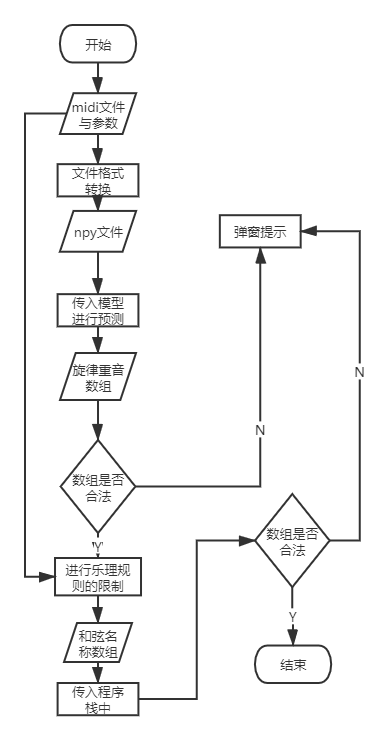


图6-3 预测模块流程图

## 6.3 乐谱展示模块设计实现

由数据层得到生成后的midi文件，将其转为musicXML乐谱文件，在前端界面使用JavaScript对musicXML文件进行音符读取；并依次在HTML的canvas上进行渲染；若格式转换出错或渲染失败则Canvas初始化并弹窗提醒。图6-4为乐谱展示模块流程图。

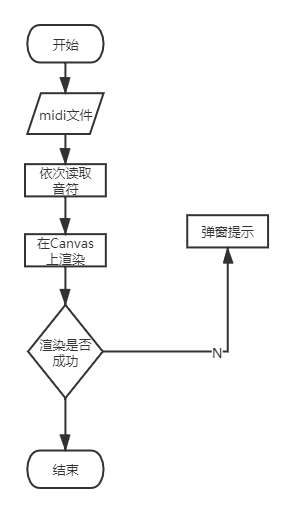


图6-4 乐谱展示模块流程图

## 6.4 midi播放及下载模块设计实现

业务逻辑层接收到predict()方法输入的和弦数组，并在post请求获得选择的演奏方式；通过二者可以生成和弦的midi音频文件，再与旋律文件合并得到最终需要的midi文件并传入前端。使用midi.js对midi文件进行播放，使用HTML中的下载控件选择下载。图6-5为midi播放及下载模块流程图。

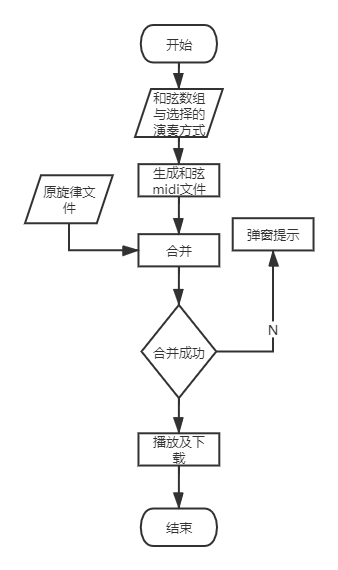


图6-5 midi播放及下载模块流程图

## 6.5 midi可视化模块设计实现

本系统使用magenta.js所内置visualize()函数实现。

## 6.6 Docker镜像部署设计实现

本系统部署在Docker镜像的python3.7环境中，开放8000端口。

## 6.6 测试

软件测试是指在规定的条件下对程序进行操作，发现程序中的错误，评估程序的质量，验证其是否能够满足设计要求，包含正确性、完整性、安全性和质量。

### 6.7.1 测试环境

表6-1 测试环境

|  |  |  |
| --- | --- | --- |
| 处理器 | 内存 | 系统 |
| 英特尔酷睿i7-9500u | 32GB | Windows 10 |

### 6.7.2 测试用例编写

测试用例（Test Case）是指对一项软件产品进行测试的描述，展现了测试的方法、技术、要求和策略。其主要内容包含测试目标、环境、输入输出、期望结果、测试脚本等，下文为最终文档。测试用例可以帮助迅速找到软件中的错误，下表主要描述对各功能块进行的功能测试及用例表。

表6-2主要记录了前端模块的测试用例，对前端模块的主页面、播放页面、乐谱页面进行测试分析。

表6-2 前端测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 用例测试步骤 | 预期结果 | 实际结果 | 备注 |
| 上传文件且选择正确参数并点击提交 | 提交按钮文字更改为请等待后页面刷新。 | 提交按钮文字更改为请等待后页面刷新。 | 无 |
| 上传文件并选择参数不全后点击提交 | 弹窗提醒“参数选择错误” | 弹窗提醒“参数选择错误” | 无 |
| 不进行上传文件便点击提交 | 弹窗提醒“未选择文件” | 弹窗提醒“未选择文件” | 无 |
| 页面跳转后点击乐谱按钮 | 跳转到乐谱界面，展示midi文件所绘制的乐谱，并有放大缩小按钮 | 跳转到乐谱界面，展示midi文件所绘制的乐谱，并有放大缩小按钮 | 无 |
| 页面处于乐谱界面时点击放大缩小按钮 | 乐谱视图随之放大缩小。 | 乐谱视图随之放大缩小。 | 无 |
| 页面跳转后单击播放按钮 | 页面跳转至播放界面，并实现midi钢琴卷轴可视化，有播放暂停速率按键。 | 页面跳转至播放界面，并实现midi钢琴卷轴可视化，有播放暂停速率按键。 | 无 |
| 页面处于播放界面，用户单击播放按钮 | 浏览器播放后台midi文件。 | 浏览器播放后台midi文件。 | 无 |
| 页面处于播放界面，midi在播放，用户单击暂停按钮 | Midi音乐停止播放。 | Midi音乐停止播放。 | 无 |
| 页面处于播放界面，midi在播放，用户在输入栏输入速率 | Midi文件调整当前速率并按此速率继续播放 | Midi文件调整当前速率并按此速率继续播放 | 无 |
| 页面处于跳转后的主界面并单击下载按钮 | 浏览器下载midi文件到本地 | 浏览器下载midi文件到本地 | 无 |

表6-3主要记录了视图层的测试用例，对视图层的模型预测、文件格式转换、乐理知识限制、midi文件生成等进行测试分析。

表6-3 视图层测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 用例测试步骤 | 预期结果 | 实际结果 | 备注 |
| 将midi文件传入midi\_converse() | 输出csv文件，文件中包含小节数、音符音高、持续时间等。 | 输出csv文件，文件中包含小节数、音符音高、持续时间等。 | 无 |
| 将csv文件传入make\_npy()文件 | 输出npy文件。 | 输出npy文件。 | 无 |
| 将csv文件传入converse\_to\_midi() | 输出midi文件，midi文件中音符对应csv文件中描述的音符。 | 输出midi文件，midi文件中音符对应csv文件中描述的音符。 | 无 |
| 将npy文件传入predict()模型进行训练 | 输出数组，元素为旋律重音，长度为旋律小节长度。 | 输出字符串，字符串为旋律重音，长度为旋律小节长度。 | 无 |
| 将重音数组传入level()函数。 | 输出为级数数组，长度与输入数组相同，并且元素仅限“I II III IV V VI VII”。 | 输出为级数数组，长度与输入数组相同，并且元素仅限“I II III IV V VI VII”。 | 无 |
| 将级数数组传入chord()函数 | 输出为和弦数组，长度与输入数组相同，并且元素按照和弦命名规范。 | 输出为和弦数组，长度与输入数组相同，并且元素按照和弦命名规范。 | 无 |
| 将和弦数组传入midi()函数进行创建和弦midi文件 | 输出为midi文件并且内容为输入所有和弦，每个和弦长度一小节。 | 输出为midi文件并且内容为输入所有和弦，每个和弦长度一小节。 | 无 |
| 将旋律Midi文件与和弦midi文件传入add()函数合并 | 输出双声部midi文件，每个声部为原midi文件 | 输出双声部midi文件，每个声部为原midi文件 | 无 |

### 6.7.3 功能展示

运行“Python manage.py runserver”部署服务器，浏览器输入127.0.0.1:8000/play，进入主界面。



在选择文件后单击上传按钮并显示进度。



在后台训练后，主界面刷新，出现功能按钮。



单击播放按钮，跳转播放界面，midi可视化，可以播放、暂停、调整速率。



单击乐谱按钮，跳转乐谱界面，可以显示乐谱并且可以放大缩小。



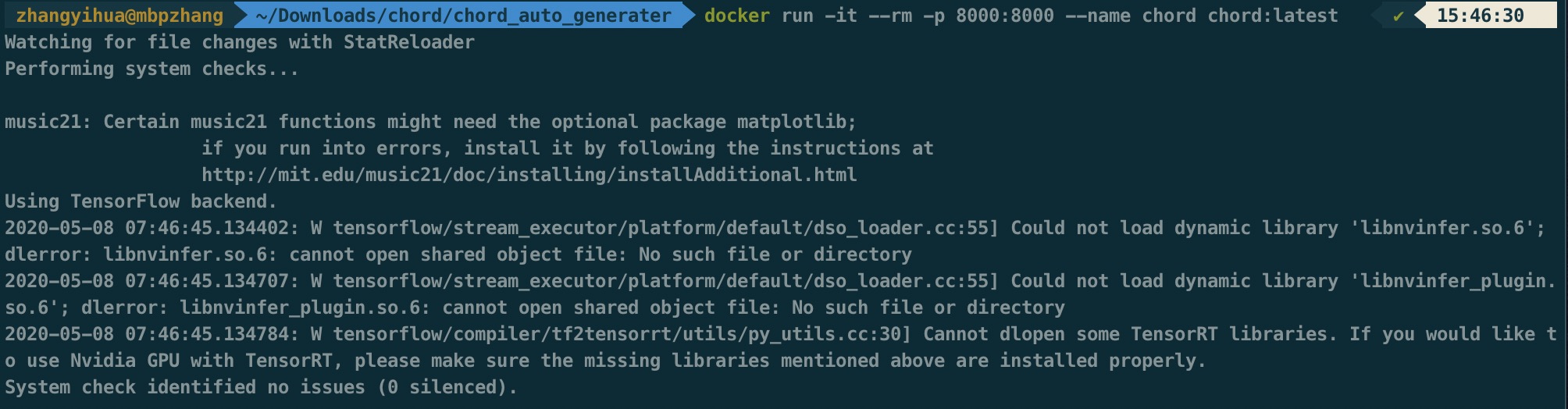
单击下载按钮，文件会自动下载。



对系统进行Docker镜像的文件编写。



对镜像的部署及在本地Docker环境运行。



## 6.7 本章小结

本章由设计各个功能块出发，描述了每个功能的具体开发规范及技术要求，从前端模块、模型模块、乐理模块、文件格式转换模块、可视化模块等等出发，详细描述了本系统功能的主要流程，阐述了每个功能块的逻辑及流程图。

其次从软件测试出发，通过测试用例描述了本系统的测试工作，最终展示了系统的主要功能以及最终成果。

# 

# 第七章 结论

## 7.1 研究结论

本文论述了和弦自动编配算法及系统的设计与实现的详细过程，以软件开发的过程为标准，从算法设计、模型搭建、需求分析、设计与实现、测试与改进几个方面论述了本系统的开发流程。本系统主要分为两大部分：一为算法的模型建立及训练预测，二为实现简易的易于模型使用的前端网页。具体完成内容如下：

### 7.1.1模型部分

* 1. 选择合适的机器学习模型

1. 选择并训练了适合于处理和预测时间序列中间隔和延迟非常长的LSTM模型。
   1. 训练集的选择以及预处理
2. 选择了便于训练的音乐样本，并且实现了训练集的one-hot编码。
3. 实现了音乐文件与字符文件之间的相互转化。
   1. 完成了以下多种和弦进行法则的约束
4. 相邻小节和弦一般不同
5. 一些常见的和弦代替
6. 和弦级数走向
7. 套用了一些古典、流行套路，如：“1645”、Romanessca模型。

### 7.1.2前端网页部分

* 1. 预测结果自动生成音乐文件

1. 将生成的和弦字符串转为midi文件，并与旋律文件合并生成完整音乐文件。
   1. 前端网页的实现
2. 能够向服务器上传midi旋律文件
3. 可以直接在网页选择模型等参数
4. 可以在线播放下载生成的midi音乐
5. 美化界面

本系统采用神经网络算法与乐理知识相结合的方法对和弦进行编配，融合了机器学习的随机性及拟人性和乐理知识的合法性及套路化，使得生成的和弦既不会偏离主流听音感受也不会太过于死板机械化，有着较良好的听感。

本系统可以在用户并不明白编曲原理的情况下，输入一段灵感而发的旋律，约30秒后即可获得匹配好和弦的音频文件。后台模型预测、文件生成、和弦走向编排都在后端运行；前端操作简单易用，只需上传文件选择参数即可获得成品音乐的乐谱、可视化界面和下载成品文件功能。

## 7.2 展望

在人工智能领域，和弦编配一直是一个热门话题，各种算法模型层出不穷，其中大多都能得到良好的听觉效果。但是我们不应该忘记在人类编曲过程中是有规则限制的，这些规则不像公式一样死板，它易于变通但是却是所有编曲的基础，那就是乐理规则。如果单纯使用神经网络算法来进行和弦的编配，在预测中有时会出现准确率很高但听觉效果不好的和弦走向，这时就应该对和弦编配时的规则加以限制，这也是本系统的重点。

本系统中所添加的乐理规则限制还扔较少，模式相对单一，匹配方式及优先级也较为单调，这是为了尽可能对全部限制能够匹配做出的妥协。在以后的研究中，如果能够对乐理规则进行全部添加，并且优先级设置合理，再配以神经网络的辅助，那么和弦自动编配的效果将会得到进一步的提升。

# 

# 参考文献

[1]. 黄振芳, 漫谈三和弦. 音乐世界, 1994(08): 第9页.

[2]. 胡广生, 计算机音乐的起源与发展. 戏剧之家, 2016(19): 第90-91页.

[3]. Teng, Y., A. Zhao and C. Goudeseune, Generating Nontrivial Melodies for Music as a Service.

[4]. 杨战, 基于隐马尔可夫模型的自动和弦识别, 2010, 哈尔滨工业大学. 第 58页.

[5]. 朱洪渊, 基于深度学习的自动作曲编曲研究, 2019, 中国科学技术大学. 第 68页.

[6]. Bahdanau, D., K. Cho and Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate. 2014.

[7]. 熊万敏, 论视唱练耳中的调式感, 2014, 上海师范大学. 第 58页.

[8]. 付丽莎, 浅论视唱练耳教学中五种常用七和弦的教学模式及其拓展, 2015, 上海音乐学院. 第 92页.

[9]. 叶盈, 《快乐钢琴基础教程》和《巴斯蒂安钢琴教程》的教材分析, 2013, 上海音乐学院. 第 78页.

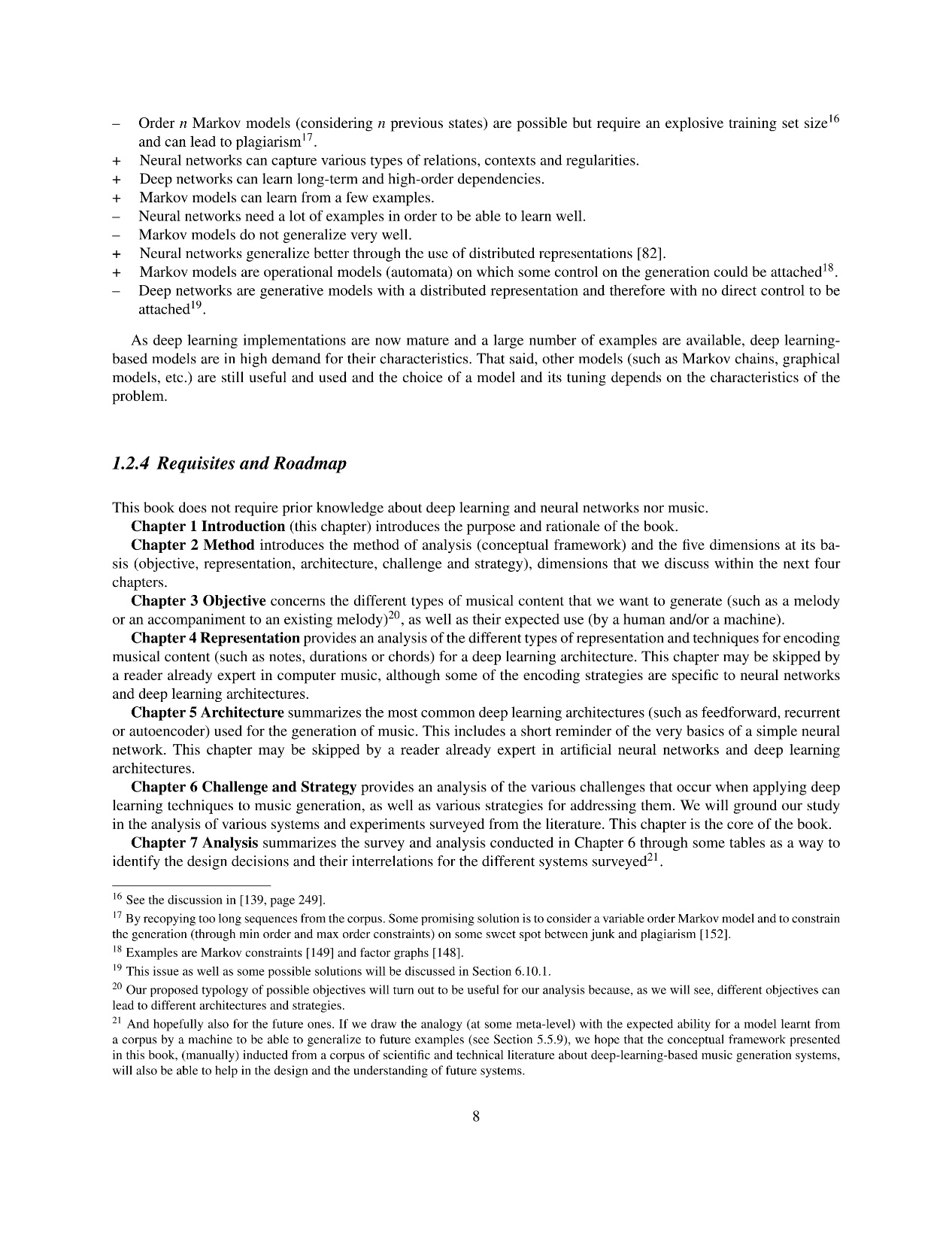
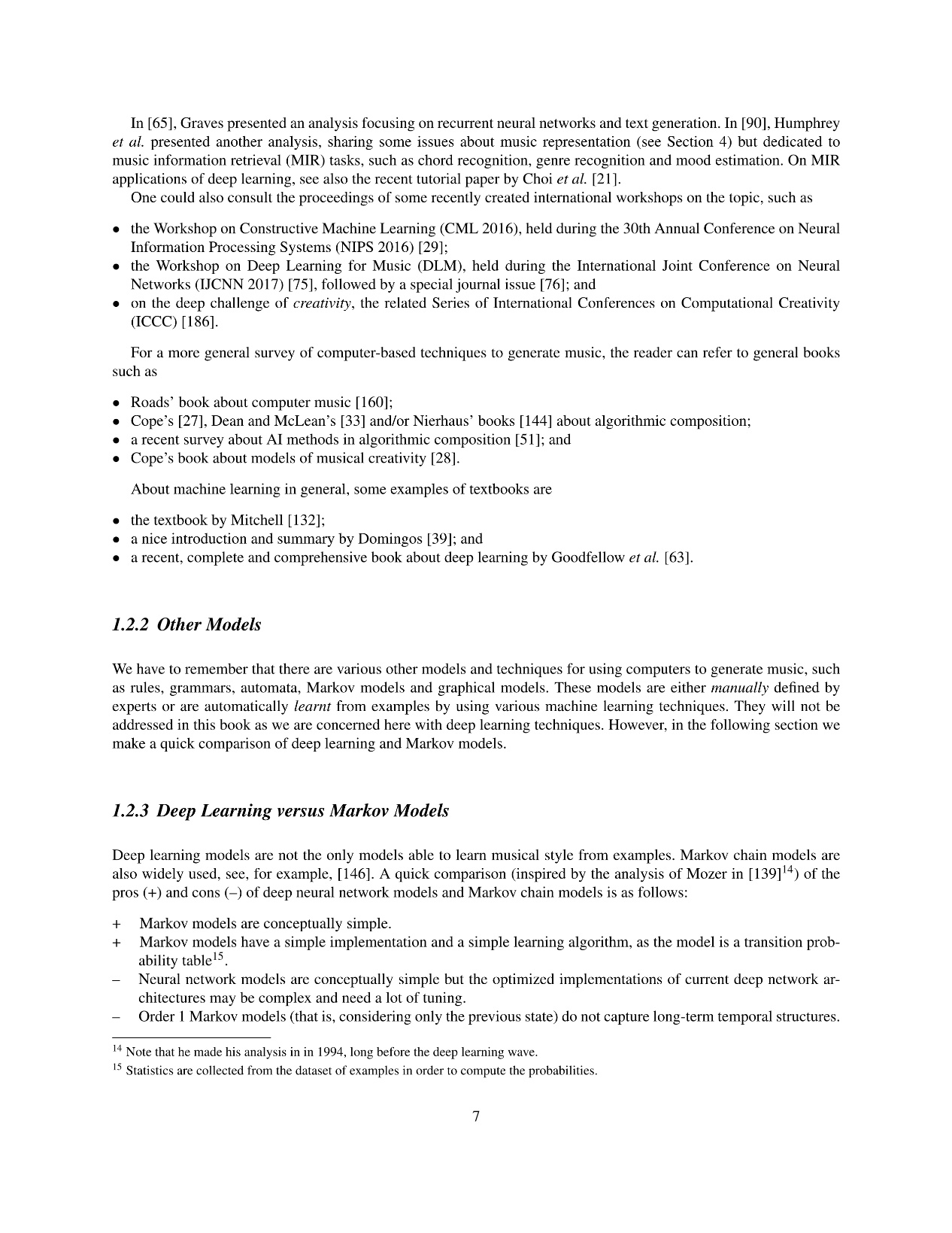
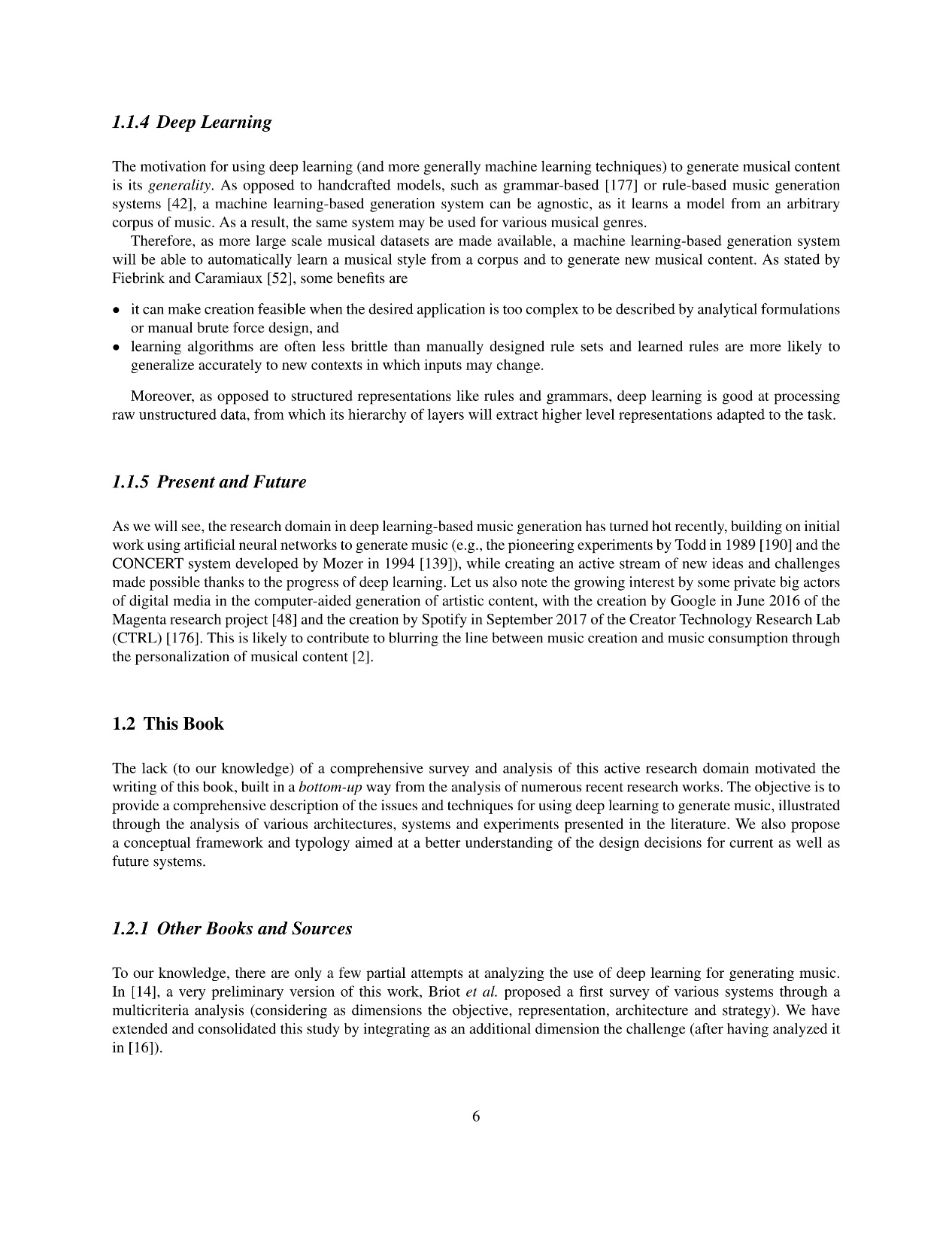
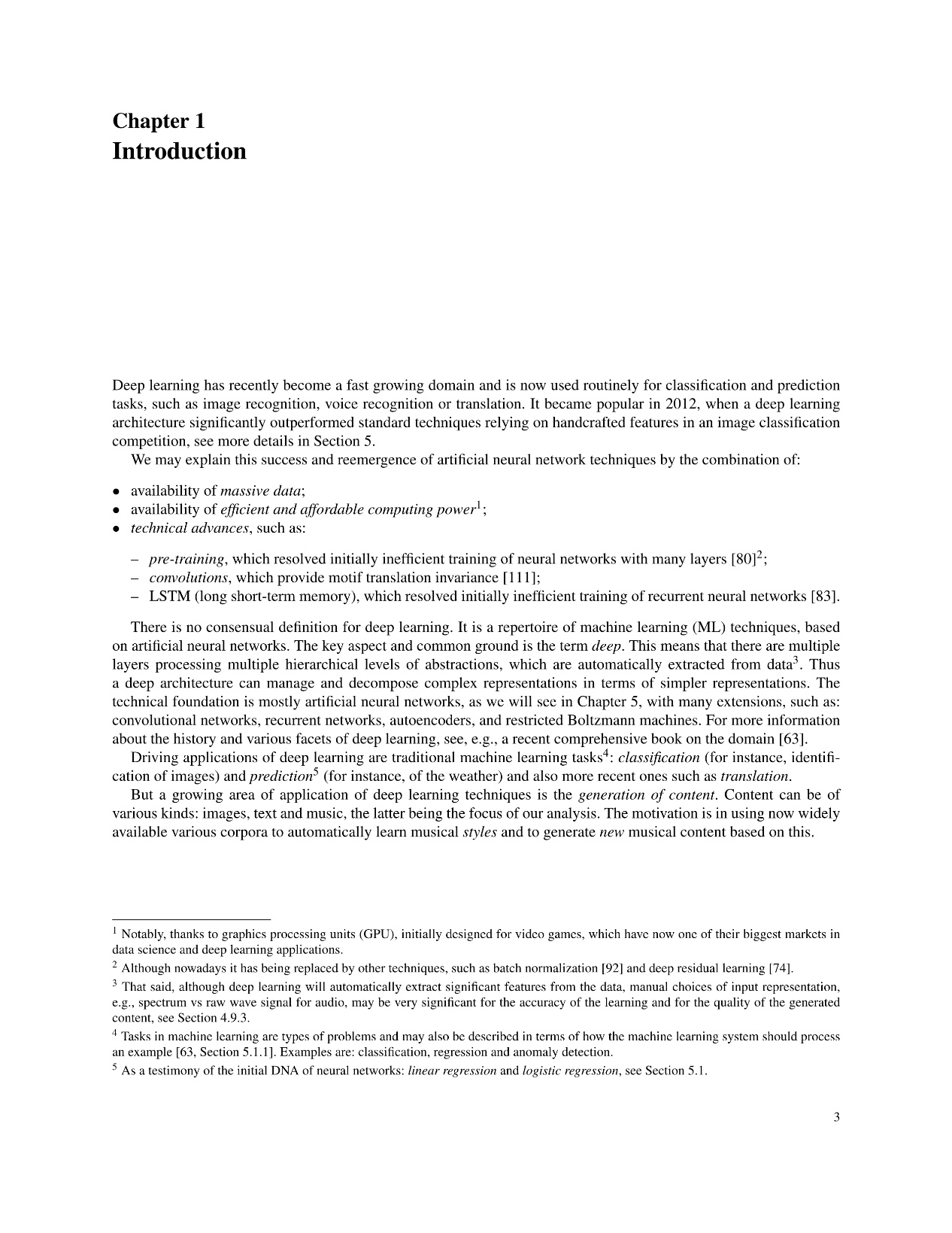
[10]. 刘康华, 和声运用中和弦结构的风格化处理(下)——论纯五声和声材料的构成、运用思维与处理技法. 中央音乐学院学报, 2018(03): 第35-46页.

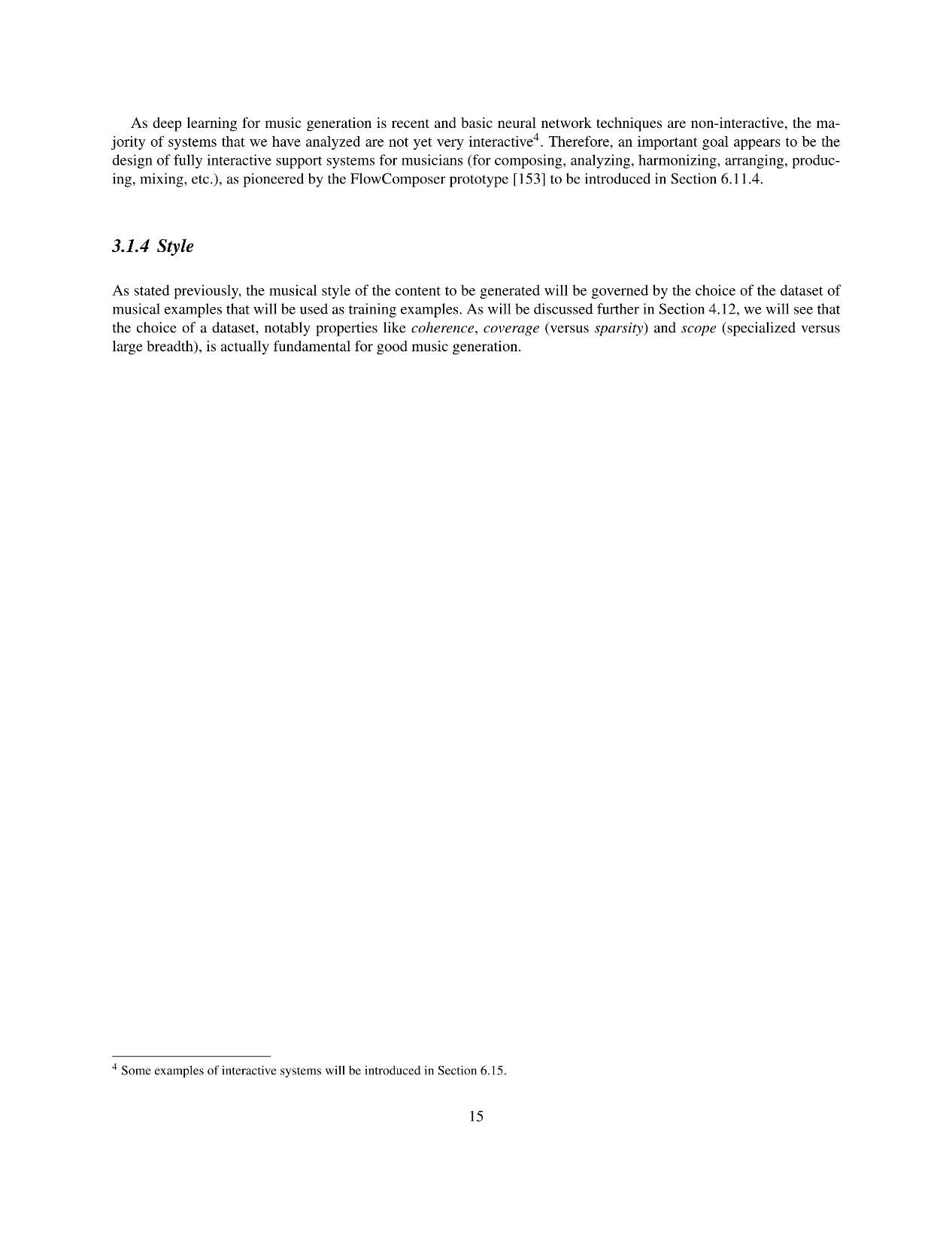
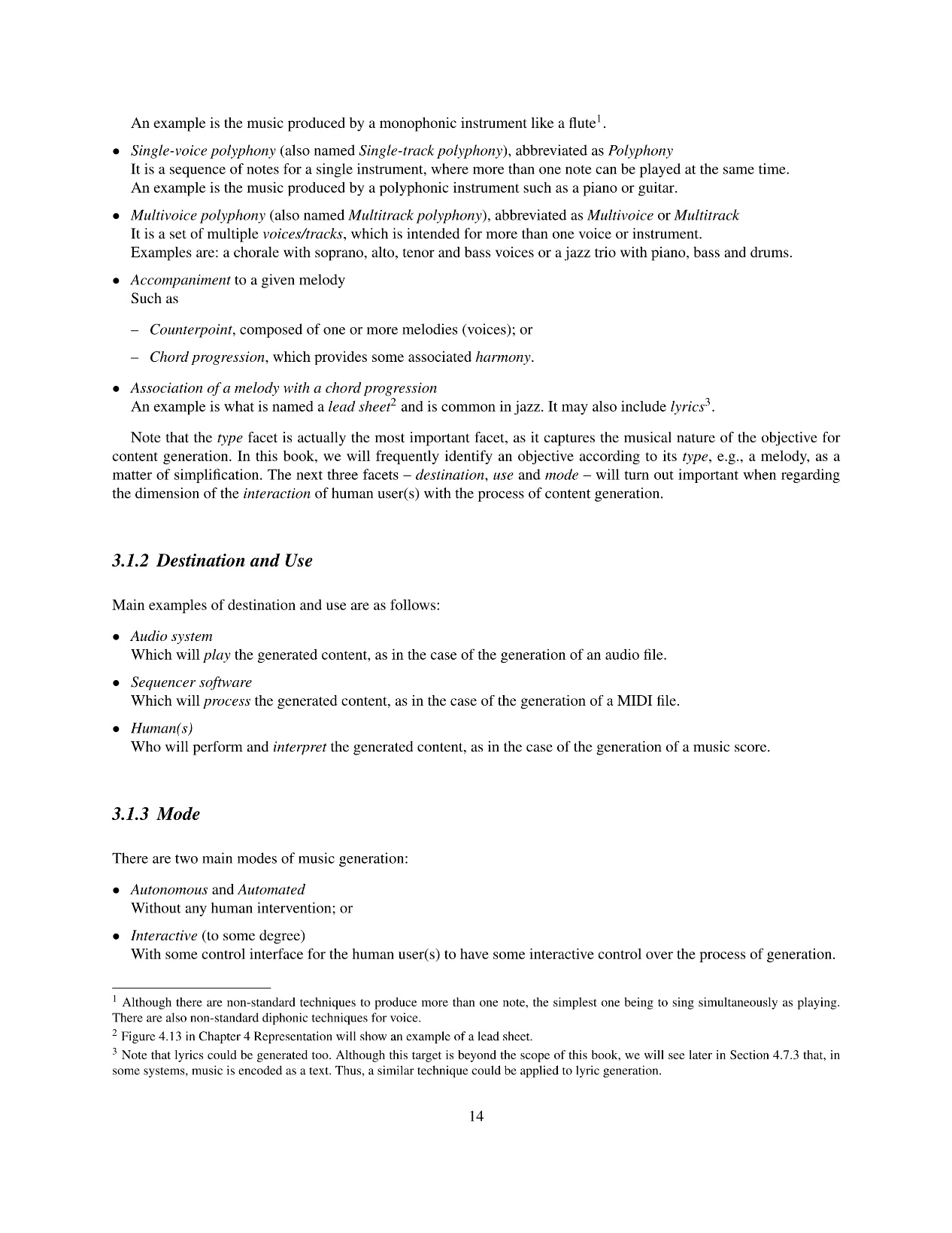
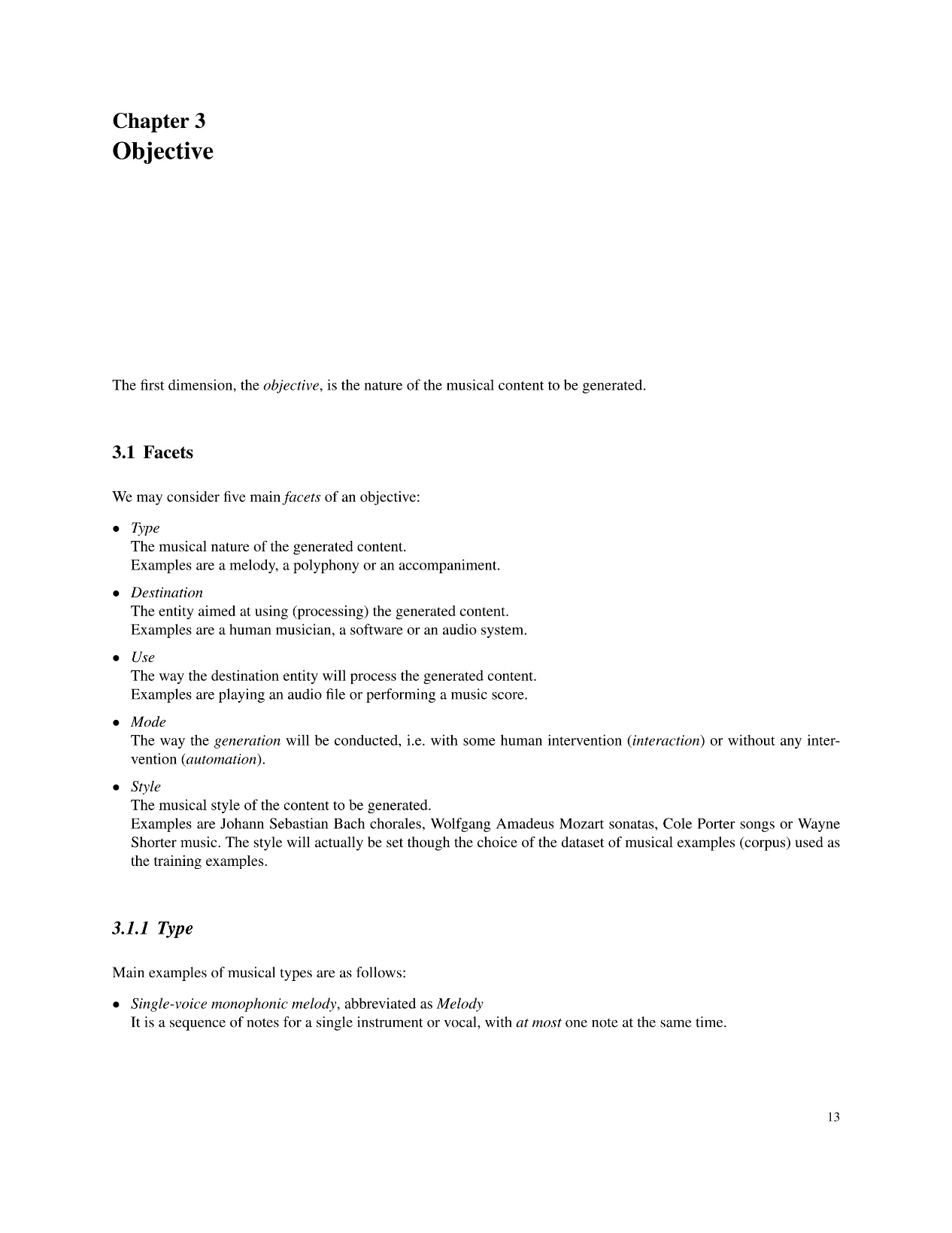
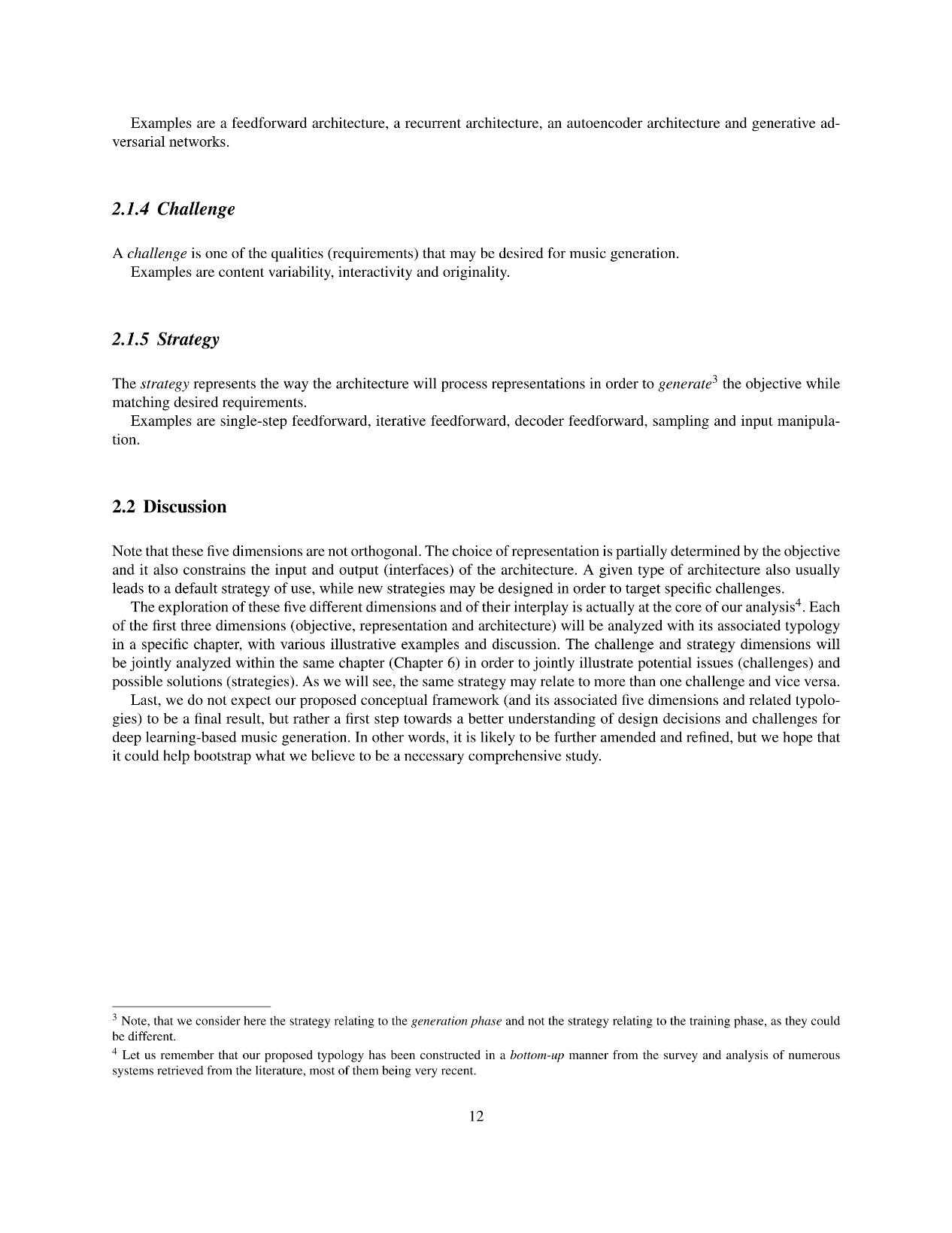
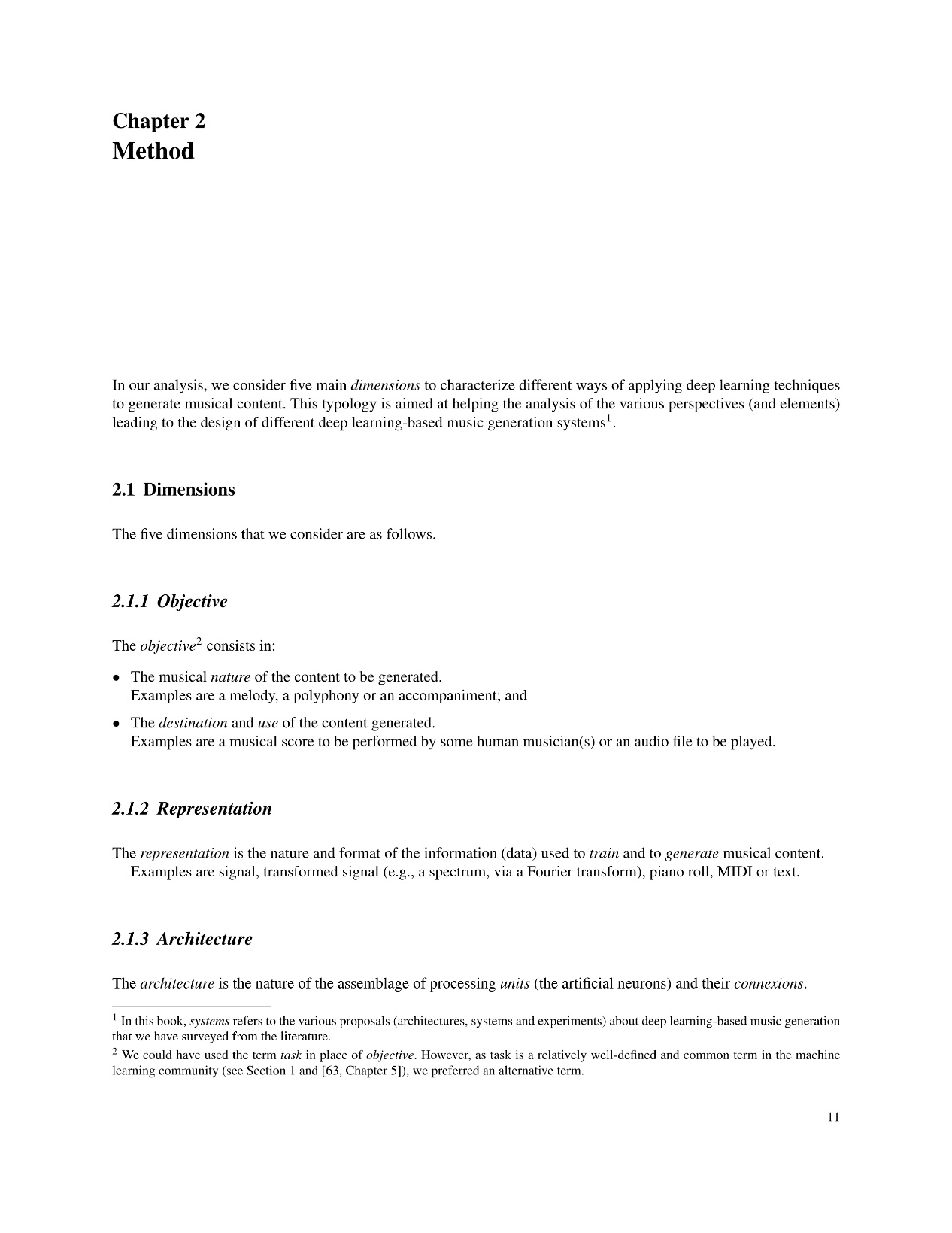
# 

# 致 谢

衷心感谢北京邮电大学软件学院崔毅东教授对本人的悉心教导，他的言传身教会使我终身受益。

外 文 资 料 原 文





**外 文 译 文**

本文对利用深度学习(deep artificial neural network，简称deep learning)生成音乐内容的不同方法进行了调查和分析。我们提出了一个基于五个维度的分析方法:

•目标

-要制作什么音乐内容?

例如:旋律、复调、伴奏或对位。

-为了什么目的，为了什么用途?

由人(在乐谱的情况下)或由机器(在音频文件的情况下)演奏。

•表示

-要处理的概念是什么?

例如:波形，谱图，音符，和弦，节拍和节拍。

-使用什么格式?

例如:MIDI, piano roll或text。

-将如何编码?

例如:标量、单热或多热。

•构建

-使用何种类型的深度神经网络?

例如:前馈网络，递归网络，自动编码器或生成对抗网络。

•挑战

-有哪些限制和挑战?

例如:可变性、互动性和创造性。

•战略

-我们如何建模和控制生成过程?

例如:单步前馈，迭代前馈，采样或输入操作。

对于每个维度，我们对各种模型和技术进行了比较分析，并提出了一些尝试性的多维类型。这种类型是自下而上的，是在分析现有的许多基于深度学习的音乐生成系统的基础上，从相关文献中选取的。这些类型被描述和用来举例说明各种选择的目标，表现，架构，挑战和战略。最后一部分包括一些讨论和展望。

深度学习最近已经成为一个快速发展的领域，现在经常用于分类和预测任务，如图像识别、语音识别或翻译。它在2012年变得流行起来，当时在一场图像分类竞赛中，依赖于深度学习的办法明显优于传统办法。

我们可以通过以下方法来解释人工神经网络技术的成功:

•海量数据的可用性;

•高效和廉价的算力;

•技术进步，如:

-训练前，解决了多层神经网络训练初期效率低下的问题；

-卷积，提供了平移不变性；

- LSTM (long short-term memory)，解决了递归神经网络训练初期效率低下的问题。

深度学习没有一致的定义。它是一套基于人工神经网络的机器学习(ML)技术。关键的方面和共同点是“深度”一词。这意味着有多个层处理多个层次抽象级别，这些抽象级别是自动从数据中提取的。因此，深度架构可以根据更简单的表示来管理和分解复杂的表示。技术基础主要是人工神经网络，有许多扩展，如:卷积网络，递归网络，自动编码器，限制玻尔兹曼机。

深度学习的驱动应用包括传统的机器学习任务:分类(例如，识别图像)和回归(例如，预测天气)。

但应用深度学习技术的一个日益增长的领域是内容的生成。内容可以是各种各样的:图像、文本和音乐，后者是我们分析的重点。其动机是利用现在广泛使用的各种语料库来自动学习音乐风格，并在此基础上生成新的音乐内容。

电脑产生的第一首音乐出现在1957年。这是一段17秒长的旋律，由作者Newman Guttman命名为“银色音阶”，由贝尔实验室的Mathews开发的一款名为Music I的声音合成软件生成。同年，《伊利亚克组曲》是第一首由计算机作曲的乐曲，它是以美国伊利诺伊大学厄巴纳-香槟分校(UIUC)的ILLIAC I计算机命名的。无论是音乐家还是科学家。它是算法组合的一个早期例子，利用随机模型(马尔可夫链)进行生成，并根据期望的属性使用规则过滤生成的材料。

在声音合成领域，一个里程碑是1983年由雅马哈发布的DX 7合成器，它建立在基于调频的合成模型的基础上。同年，MIDI6接口发布，作为一种互操作各种软件和仪器的方式(包括雅马哈DX 7合成器)。另一个里程碑是Puckette在IRCAM上开发的Max/MSP实时交互处理环境，用于实时合成和交互性能。

关于算法组成，在20世纪60年代早期，Iannis Xenakis在他1962年的论文《Atr’ees》中探讨了随机组成的概念。这个想法包括使用计算机快速计算，从作曲家设计的一组概率中计算各种可能性，以生成要选择的音乐作品的样本。在另一种遵循“Illiac套件”最初方向的方法中，语法和规则被用来指定给定语料库或更普遍的调性音乐理论的风格。一个例子是1980年代的一代Ebcio˘glu的组合软件命名的四声部合唱合唱团可以实现约翰·塞巴斯蒂安·巴赫的风格。20世纪80年代末，大卫·科普(David Cope)的名为“音乐智能实验”(简称EMI)的系统扩展了这一方法，使其能够从作曲家的乐谱库中学习，创建自己的语法和规则数据库。

从那以后，计算机音乐继续为普通大众发展，如果我们考虑一下，例如GarageBand音乐作曲和苹果平台(电脑、平板电脑和手机)的生产应用程序，它是Steinberg在1989年发布的最初的Cubase音序器软件的后代。

人工智能(AI)通常分为两大主流:

符号人工智能——处理高级符号表示(例如和弦，和声)和过程

亚符号人工智能——处理低层次的表现(例如声音、音色)和过程(音高识别)

用于音乐的符号模型的例子是基于规则的系统或表示和谐的语法。用于音乐的子符号模型的例子是机器学习算法，用于从音乐作品的语料库中自动学习音乐风格。这些模型可以以生成和交互的方式来帮助音乐家创造新音乐，利用这些新增的“智能”记忆(联想、归纳和生成)来提出建议、草图、推断、映射等。这现在是可行的，因为各种形式的音乐越来越多。声音、分数和MIDI文件，可由电脑自动处理。

最近一个集成音乐创作环境的例子是FlowComposer[153]。它提供了各种象征性和次象征性的技术，例如。其中，马尔科夫链用于建模风格，约束求解模块用于表达约束，基于规则的模块用于产生谐波分析;和一个音频映射模块来生成渲染。另一个集成音乐创作环境的例子是OpenMusic。

sub-symbolic技术的更深层次的整合,如深度学习,通过象征性的技术,如约束和推理,仍然是一个开放的课题,尽管部分集成在受限制的环境中已经存在。

使用深度学习(以及更普遍的机器学习技术)来生成音乐内容的动机是它的通用性。与手工制作的模型(如基于语法的或基于规则的音乐生成系统)不同，基于机器学习的生成系统可能是不可知的，因为它从任意音乐语料库学习模型。因此，相同的系统可以用于不同的音乐类型。

因此，随着越来越多的大规模音乐数据集的出现，一个基于机器学习的生成系统将能够从语料库中自动学习音乐风格并生成新的音乐内容。正如Fiebrink和Caramiaux所述，一些好处是:

•当所需的应用程序过于复杂，无法通过分析公式或手动蛮力设计进行描述时，它可以使创建变得可行。

•学习算法通常不像手工设计的规则集那么脆弱，学习规则更有可能精确地泛化到输入可能改变的新上下文。

此外，与规则和语法等结构化表示不同，深度学习擅长处理原始的非结构化数据，它的层次结构将从中提取适合于任务的更高级别的表示。

正如我们将看到的，基于深度学习的音乐生成的研究领域最近已经成为热点，它建立在使用人工神经网络生成音乐的初步工作(例如，， Todd在1989年的开创性实验和Mozer在1994年开发的CONCERT系统)，同时创造了一系列积极的新思想和挑战，这一切都得益于深度学习的进步。也让我们注意到一些私人编曲家越来越感兴趣的计算机辅助生成的数字媒体艺术的内容,谷歌在2016年6月创建的洋红色的研究项目和Spotify在2017年9月创建的创造者技术研究实验室(CTRL)。这可能有助于通过个性化的音乐内容模糊音乐创作和音乐消费之间的界限。

音乐中的维度大多不是正交的。表现形式的选择部分由目标决定，同时也限制了体系结构的输入和输出接口。给定的体系结构类型通常还会导致默认的使用策略，而新的策略可能是为了针对特定的挑战而设计的。

对这五个不同维度及其相互作用的探索实际上是我们分析的核心。前三个维度(目标、表现和构建)中的每一个都将在特定的章节中通过相关的类型学进行分析，并提供各种说明性的例子和讨论。挑战和战略维度将在同一章中共同分析，以共同说明潜在的问题(挑战)和可能的解决方案(战略)。我们将看到，相同的策略可能涉及到多个挑战，反之亦然。

最后，我们不期望我们提出的概念框架(及其相关的五个维度和相关类型)是最终的结果，而是朝着更好地理解设计决策和基于深度学习的音乐生成的挑战迈出的第一步。换句话说，它可能会进一步修正和完善，但我们希望它能够帮助引导我们认为是必要的全面研究。

转移学习是深度学习和机器学习的一个重要问题。由于培训可能是一个冗长乏味的过程，问题在于能够重用(至少是部分重用)在一个上下文中所学的知识，并将其用于其他上下文中。可以考虑不同的情况，例如:、相似的源和目标域、相似的任务等。这个新的研究子领域，称为转移学习，转移学习也会是深度学习下一大重要方向。

我们没有在我们的分析中处理这个重要的问题，因为它还没有专门针对音乐生成，尽管我们认为它将成为一个研究领域。与此同时，一个例子，虽然仍然简单，是DeepHear的架构和它所学到的是如何从产生旋律的目标转移到产生对位的目标。另一个例子是DeepBach架构允许将目标从前无伴奏合唱生成调整为多声部伴奏。

最后，让我们记住，就目标和技术而言，风格迁移是一个非常具体的迁移学习案例

一个普遍的问题是为特定目标或特定类型的语料库设计的系统的高度专门化。这可以从所调查的架构和方法的多样性中得到证明。请注意，这是人工智能(AI)研究中一个众所周知的问题。有一种倾向是，过于专业化的系统解决特定的问题，特别是在由会议或其他机构组织的竞赛中，可能会与一般问题解决框架的最初目标不尽相同。

同时，产生有趣的音乐内容的总体目标是复杂的，仍然是一个开放的问题。因此，我们既要研究解决一般问题的一般方法，也要研究解决具体次问题的具体方法，还要研究自顶向下和自底向上的方法，同时又要对如何解释、概括和重用所取得的进步和吸取的教训保持兴趣。

这本书不需要关于深度学习和神经网络或音乐的先验知识。

第一章导论(本章)介绍了本书的目的和基本原理。

第二章方法介绍了分析方法(概念框架)及其基础的五个维度(目标、表现、架构、挑战和战略)，这些维度我们将在接下来的四章中讨论。

第3章目标涉及到我们想要产生的不同类型的音乐内容(如旋律或现有旋律的伴奏)，以及它们的预期用途(由人和/或机器)。

第4章为深度学习架构提供了对不同类型的音乐内容(如音符、持续时间或和弦)编码的表现形式和技术的分析。本章可能会被已经是计算机音乐专家的读者跳过，尽管一些编码策略是特定于神经网络和深度学习架构的。

第5章架构概述了最常见的深度学习架构(如前馈、递归或自动编码器)用于产生音乐。这包括一个简单的神经网络基础的简短提示。本章可能会被一个已经是人工神经网络和深度学习架构专家的读者跳过。

第六章挑战和策略分析了在音乐生成中应用深度学习技术时遇到的各种挑战，以及解决这些挑战的各种策略。我们将把我们的研究建立在对各种系统的分析和文献调查的实验上。这一章是这本书的核心。

第七章分析总结了第六章的调查和分析，通过一些表格来识别不同系统的设计决策及其相互关系。

第8章讨论和结论在总结本书之前，回顾了在分析第6章中提出的挑战和战略时触及的一些开放问题。

在我们的分析中，我们考虑了五个主要的维度来描述应用深度学习技术生成音乐内容的不同方式。这种类型旨在帮助分析不同的视角(和元素)设计了不同的基于深度学习的音乐生成系统。

自动音乐制作系统的探索可能是探索作曲过程的一个有趣的视角，同时也是一种评价方法。音乐图灵测试是由巴赫或由深度学习系统产生并由人类音乐家演奏，正如我们将在下面看到的，深度学习技术在这些测试中被证明是非常有效的，因为它们能够从给定的语料库中学习音乐风格，并生成符合这种风格的新音乐。也就是说，我们认为这样的试验与其说是一种目的，不如说是一种手段。

更广阔的视角是帮助人类音乐家在音乐创作的各个步骤:作曲，安排，管弦乐编曲，制作等。的确，为了作曲或即兴创作，音乐家很少从头开始创作新音乐。他/她有意识或无意识地重复和适应他/她已经知道或听过的各种音乐的特点，同时遵循一些原则和指导方针，如和声和音阶理论。基于计算机的音乐家助手可以在作曲的不同阶段发挥作用，激发、暗示、激发和/或补充人类作曲家的灵感。

也就是说，正如我们将看到的，尽管越来越多的系统解决了人的控制和交互的问题，但目前大多数基于深度学习的音乐生成系统仍然专注于自主生成。

对于很多实验来说，评估仅仅是初步的，在很多情况下，评估是由设计者自己进行的。当然也有一些例外，需要更系统的外部评估。

当语料库非常精确时，如巴赫为巴赫机器人或迪普巴赫系统所作的合唱，可以进行图灵测试:将一段音乐呈现给公众，让他们猜测它是计算机生成的原始音乐还是音乐。但是，当目标不是生成高度符合相对狭窄的风格(和语料库)的音乐时，这种方法就更加有限了，比如巴赫合唱，但产生更多的创造性的音乐。

此外,如果我们考虑作为一个系统的总目标的能力协助作曲家和音乐家,而不是自动生成音乐(参见1.1.2节),我们也许应该考虑作为评估标准的满意度的作曲家(值得注意的是,如果计算机的帮助让他撰写并创造音乐,否则他可能会考虑不可能),而不是准确度的满意度(他们经常被一些符合当前音乐趋势的东西所引导)。

一些基本的限制是，没有明确的目指标去衡量创造力和艺术质量。因此，选择一个音乐语料库作为一组训练示例是第一个基本步骤和决策。但是为了能够学到一些有趣的东西，需要选择一个相对连贯/同质的语料库。不幸的是，这将有利于学习风格的音乐内容的质量(实际上，一致性)，而不是它的内在质量(兴趣)。

当前的实验和促进创新的方向主要依赖于避免剽窃的约束和/或激励“舒适区”之外的一代人的启发式方法，深层架构已经从语料库中学习了这些方法，同时平衡惊喜元素和可预测性/可理解性。这种创造性控制可以在培训阶段应用。

为了更好地模拟这种惊喜元素，可以采用另一种(和互补的)方向，即包括一个带有某种期望模型的人工监听器模型。，它可以评估新生成的内容在引用现有内存方面“解释”得有多好。在[41]。

最后，一些额外的基本限制是，目前学习和产生音乐的深度学习技术是基于人工制品，实际的音乐数据，独立于产生它们的过程和文化。如果我们想要设想更深刻的系统，很可能我们将不得不结合一些背景和过程的建模，从而导致产生的是音乐个体，而不是作品本身。的确，当我们考虑艺术史时，创作是在历史文化背景下进行的，既有可能破裂，也有可能改进。因此，一个可能的方向是，不仅要从一个冻结的艺术语料库中模仿内容的生成，而且要尝试模仿一个更动态的创作过程，包括历史文化背景。

**北 京 邮 电 大 学**

**本科毕业设计（论文）开题报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 软件学院 | 专业 | | 软件工程 | | 班级 | 2016211504 |
| 学生姓名 | 孙一帅 | 学号 | | 2016212026 | | 班内序号 | 20 |
| 指导教师姓名 | 崔毅东 | 所在单位 | | 软件学院 | | 职称 | 副教授 |
| 设计（论文）题目 | （中文）音乐旋律的和弦编配算法的研究与实现 | | | | | | |
| （英文）The Research and Implementation of the Chord Assembling Algorithms for Melodies | | | | | | |
| 毕业设计（论文）开题报告内容：   1. 选题的背景和意义：   和弦进行是和声学的范畴,和声学对于音乐的定义是，音乐是符合人情感的时间上的艺术，是由多个稳定到不稳定再到解决的进行构成。常见的进行有自然大调的T-S-D-T（包括其变体，如柴可夫斯基《杜鹃》中的T-TSⅥ-S-SⅡ-D-DTⅢ-T）、自然小调的t-d-s-t、和声小调的t-s-D-t等等。理论上和弦数量有限，即解空间有界，可以用枚举+评估的方法得到答案，虽然这种做法并不符合人类作曲的思路，但它为和弦的自动编配提供了理论可能。  现在很多编曲软件都提供类似于“和弦助手”的功能，在给定旋律的情况下能够自动推算出下一小节合法且悦耳的和弦名称；那么在给定整首曲子旋律的情况下，可以通过机器学习加乐理限制自动提供整首曲子的和弦名称和自动编配，能够为音乐创作人提供灵感甚至代替一部分工作，对于不懂乐理的人也可以在只想到一阵灵感旋律的情况下，得到悦耳合法的一段音乐。   1. 研究内容和拟解决的主要问题：   在进行基于人工智能的音乐创作研究过程中，需要为给定的旋律自动编配和弦。目前已有一些和弦编配算法，但其编配结果平淡、缺少变化。研究基于机器学习的和弦编配算法并与已有算法进行对比和测试。  我们的最终目的是训练出一个机器学习模型，能够根据已给出旋律自动添加和弦，该模型能够根据旋律走向选择最和谐的和弦进行编排，符合基本乐理原理。同时，该模型可以通过训练不同种类的音乐文件来生成不同风格的和弦。最重要的一点是，该模型生成的和弦拥有多种和弦变式，具有类似人的音乐创作习惯。   1. 研究方法及措施：   首先了解python相关代码知识，学习midi文件的处理方法。根据需求，对训练集进行有效的划分与清洗。  其次阅读机器学习相关书籍，选择适合的模型；结合相应的乐理知识，对算法需要选择的和弦不断给予乐理支持，增加准确度和悦耳程度。  最后根据模型代码，调整不同风格的节奏型，并且写入配置文件方便使用和修改。   1. 研究工作的步骤与进度：   第1-2周：结合自身兴趣，与导师讨论选题方向，确定选题内容。针对目标选题，确定主攻方向和主要难点，编写任务书和开题报告。  第3-5周： 查阅相关论文，明确项目实现的具体流程以及相关细节。  第6-10周：学习并完成对各类音乐文件的字符化处理及转换，学习音乐文件有关的python第三方库；了解有关机器学习的算法，并确定模型的方向。  第11-15周：初步得出模型，并尝试调优。复习相关乐理知识，确定并实现下一步调优方向；进行多种测试，提高程序鲁棒性。  第16-19周：编写前端程序，使程序更加方便使用和测试。  第20-24周：撰写毕业论文，准备答辩。   1. 主要参考文献 2. <https://github.com/musescore/MuseScore> 3. <https://musescore.org/zh-hans> 4. 尤静波，基础乐理[J]，2018 5. 李航，统计学习方法，2012 6. 周志华，机器学习，2016   [6] Yifeng Teng，Anny Zhao，generating nontrivial melodies for music as a service[D]  [7] 王卉，浅谈流行歌曲的钢琴编配，2014  [8] 钢琴伴奏多元混合编配实例解析，2019 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 2019年 11月 15日 | | |

**北 京 邮 电 大 学**

**本科毕业设计（论文）中期进展情况检查表**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | | | 软件学院 | 专业 | | 软件工程 | | 班级 | 2016211504 |
| 学生姓名 | | | 孙一帅 | 学号 | | 2016212026 | | 班内序号 | 20 |
| 指导教师姓名 | | | 崔毅东 | 所在单位 | | 软件学院 | | 职称 | 副教授 |
| 设计（论文）题目 | | | 音乐旋律的和弦编配算法的研究与实现 | | | | | | |
| 目前已完成任务 | 主要内容:  选题的背景和意义：  和弦进行是和声学的范畴,和声学对于音乐的定义是，音乐是符合人情感的时间上的艺术，是由多个稳定到不稳定再到解决的进行构成。常见的进行有自然大调的T-S-D-T（包括其变体，如柴可夫斯基《杜鹃》中的T-TSⅥ-S-SⅡ-D-DTⅢ-T）、自然小调的t-d-s-t、和声小调的t-s-D-t等等。理论上和弦数量有限，即解空间有界，可以用枚举+评估的方法得到答案，虽然这种做法并不符合人类作曲的思路，但它为和弦的自动编配提供了理论可能。  现在很多编曲软件都提供类似于“和弦助手”的功能，在给定旋律的情况下能够自动推算出下一小节合法且悦耳的和弦名称；那么在给定整首曲子旋律的情况下，可以通过机器学习加乐理限制自动提供整首曲子的和弦名称和自动编配，能够为音乐创作人提供灵感甚至代替一部分工作，对于不懂乐理的人也可以在只想到一阵灵感旋律的情况下，得到悦耳合法的一段音乐。  主要任务及目标：  在进行基于人工智能的音乐创作研究过程中，需要为给定的旋律自动编配和弦。目前已有一些和弦编配算法，但其编配结果平淡、缺少变化。研究基于机器学习的和弦编配算法并与已有算法进行对比和测试。  我们的最终目的是训练出一个机器学习模型，能够根据已给出旋律自动添加和弦，该模型能够根据旋律走向选择最和谐的和弦进行编排，符合基本乐理原理。同时，该模型可以通过训练不同种类的音乐文件来生成不同风格的和弦。最重要的一点是，该模型生成的和弦拥有多种和弦变式，具有类似人的音乐创作习惯。  已完成任务：  本课题设计和实现的音乐旋律的和弦编配系统，本系统主要分为两大部分：一为算法的模型建立及训练预测，二为实现简易的易于模型使用的前端网页。目前已完成本课题主要内容如下：   1. 模型部分：    1. 选择合适的机器学习模型 2. 选择并训练了适合于处理和预测时间序列中间隔和延迟非常长的LSTM模型。    1. 训练集的选择以及预处理 3. 选择了便于训练的音乐样本，并且实现了训练集的one-hot编码。 4. 实现了音乐文件与字符文件之间的相互转化。    1. 完成了以下多种和弦进行法则的约束 5. 相邻小节和弦一般不同 6. 一些常见的和弦代替 7. 和弦级数走向 8. 套用了一些古典、流行套路，如：“1645”、Romanessca模型。 9. 前端网页部分    1. 预测结果自动生成音乐文件 10. 将生成的和弦字符串转为midi文件，并与旋律文件合并生成完整音乐文件。     1. 前端网页的实现 11. 能够向服务器上传midi旋律文件 12. 可以直接在网页选择模型等参数 13. 可以在线播放下载生成的midi音乐 14. 美化界面 | | | | | | | | |
| 是否符合任务书要求进度 符合 | | | | | | | | |
| 尚需完成的任务 | 1. 模型部分： 2. 旋律文件的鲁棒性。 3. 增加分解和弦的种类。 4. 前端部分： 5. 优化选择选项数目，使之更加易用。 | | | | | | | | |
| 能否按期完成设计（论文）能 | | | | | | | | |
| 存在问题和解决办法 | 存  在  问  题 | * + 1. 特殊型节拍与常见分解和弦不匹配问题。     2. 旋律文件格式的问题 | | | | | | | |
| 拟  采  取  的  办  法 | 1. 采用拉伸或自定义分解和弦来解决。 2. 利用第三方库进行格式转换。 | | | | | | | |
| 指导教师签字 | |  | | | 日期 | | 2020年 3 月 27 日 | | |
| 检查小组意见 | | 进展基本符合预期，后面需抓紧时间。通过中期检查  负责人签字： 2020年 3 月 27 日 | | | | | | | |

**北 京 邮 电 大 学**

**教师指导本科毕业设计（论文）记录表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 软件学院 | 专业 | | 软件工程 | | 班级 | 2016211504 |
| 学生姓名 | 孙一帅 | 学号 | | 2016212026 | | 班内序号 | 20 |
| 指导教师姓名 | 崔毅东 | 职称 | | 副教授 | | | |
| 第1－2周记录：   1. 找到了精确的数据集：我们使用了Wikifonia.org提供的线索表数据库，这是一个公共线索表存储库。不幸的是，该网站在2013年停止了服务，但其中一些数据是由MusicXML格式的西方音乐唱片（包括摇滚，流行音乐，乡村音乐，爵士乐，民间音乐，RnB，儿童歌曲等）组成，并在终止之前和我们仅出于学术目的从数据中提取特征。从获得的数据库中，我们收集了2252个csv文件，它们都是主音，并且所有小节每小节都有一个和弦。这套唱片分为两套-训练套1802首歌曲和测试套450首歌曲。 2. 复习有关声乐只是，如五线谱、和弦、音调转换、五度十二律。 3. 选用合适的LSTM模型以及参数进行训练，生成每一小节的和弦名称。 4. 使用ipad库乐队进行测试，并自己手动编配底奏。 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 2019年 11 月 29 日 | | |
| 第3－4周记录：   1. 学习使用MIT的music21 python第三方库，该库可以处理包括musicXML、MIDI、abc等多种格式的音乐文件，可以从零开始构建音乐文件或对音乐进行分析。更重要的是，除了音符以外，和弦在music21中也有其单独的类型 music21.chord.Chord。 2. 对musicXML格式进行学习，其特有的乐谱保存格式可以更加鲜明地对生成结果进行展示，并且其相较于midi，可以储存更过的音符信息，更像演奏效果。 3. 实现了音乐文件向csv文件的转换，同样利用music21库，读取midi文件中的note，按照时间和小节进行保存。 4. 实现了csv向音乐文件文件的转换，同样利用music21库，根据csv文件中note属性进行自动还原，可以输出musicXML文件并且在musescore上进行演奏。 5. 尝试在已完成输出和弦字符的基础上，实现字符向音符的转换。 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 2019年 12 月 13 日 | | |
| 第5－6周记录：   1. 复习相关乐理知识，和弦进行是和声学的范畴。和声学对于音乐的定义是，音乐是符合人情感的时间上的艺术，由多个稳定到不稳定再到解决的进行构成。常见的进行有自然大调的T-S-D-T（包括其变体，如柴可夫斯基《杜鹃》中的T-TSⅥ-S-SⅡ-D-DTⅢ-T）、自然小调的t-d-s-t、和声小调的t-s-D-t等等。理论上和弦数量有限，即解空间有界，可以用枚举+评估的方法得到答案，虽然这种做法并不符合人类作曲的思路，但它为后者提供了理论可能。 2. 修改模型，将直接生成和弦转位生成旋律音，由乐理知识推出和弦名称。 3. 通过旋律音判断调性：音乐中的调性往往可以由首尾看出，可以编写程序自动判断调性。 4. 尝试利用和弦自然进行法则来添加和弦：在整个进程中，前后自然连接的两个和弦，前一个，可以看作是一个“临时属音和弦”，而后一个则是“临时主和弦”。因为前一和弦的根音，是后一和弦根音的纯五度音。当然，这只是两和弦根音之间“临时属音”与“临时主音”的关系。属音和弦，即构建在属音上的和弦，有时未必有完全的和弦属功能，比如小三和弦，因其三音在主音下方大二度，进入主音的倾向弱化，不具备导音功能。 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 2019年 12 月 27 日 | | |
| 第7－8周记录：   1. 由编曲软件cubase中和弦助手获得灵感：我们可以在工程（Project）菜单下找到和弦垫（Chord Pad），这个功能可以很快的将所有的和弦都分配到各个打击垫上，并且一下就能触发他们来弹奏一些你演奏不了的复杂的和声音序。它甚至可以允许你在钢琴和吉他和弦之间进行切换，因为演奏不同乐器的用户在面对同一个和弦时会有不同的习惯。甚至可以在和弦垫上点击鼠标右键，通过MIDI输入设备弹奏的和弦来为每一个和弦垫规定所要弹奏的和弦，这就意味着你可以重新分配每个音垫的和弦音符来使得这些和弦垫上的和弦不会破坏你整首曲子的结构，我们也可以利用代码来实现这一功能，利用自定义的节奏来进行辅助编曲，使用者可以对同一个曲子选择单和弦或自定义。 2. 利用不同种类的和弦走向法则来对同一首曲子进行匹配，使听感更加悦耳。 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 2020年 1 月 10 日 | | |
| 第9－10周记录：   1. 利用常见的和弦进行法则对和弦走向进行约束，如：1641进行法则、“传奇”进行法则、古典进行法则、Romanessca模型等。利用动态规划算法，推演出最佳的整曲和弦编配。 此外，还增加了一些约束条件，如：相邻小节和弦一般不同、和弦的常见代替、和弦的级数走向；并使用多个简单旋律进行测试。 2. 自行编写了不同拍号的节奏型，程序可以自动按照此节奏型进行和弦编配；取代了此前的一小节一个单音三和弦。解决并测试了不同拍号下、不同调性下各种简单旋律的鲁棒性，并且可以当选择节奏与旋律排号不符时，可以自动转 3. 自学Django框架，对多个单一python文件进行统一调用，方便管理和测试。 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 2020年 2 月 14 日 | | |
| 第11－12周记录：   1. 通过上周的学习，了解Django框架基本知识，实现了一个简单的界面，方便使用者对模型进行使用。 2. 界面可以自动读取后台路径并获得相关参数显示在前端。 3. 使用者可以自行上传想要生成和弦的音乐，并选择模型参数、演奏方式等，一键上传并进行预测。 4. 预测结束后自动跳转页面并拉起musescore，使用者可以再次进行演奏或者二次编辑。 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 2020年 2 月 28 日 | | |
| 第13－14周记录：   1. 学习HTML+CSS，对界面进行美化，将输入选项卡尽量替换为选择选项卡，进一步方便使用者使用。 2. 查阅相关资料，学习JS，最终选择midi.js实现生成midi文件的在线播放、暂停以及下载。 3. 简化页面跳转，使所有逻辑通过部件的隐藏和出现实现在同一页面。 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 2020年 3 月 13 日 | | |
| 第15－16周记录：   1. 优化Django前端逻辑，可以在同一页面重复提交不同旋律，生成不同的乐曲，不必要重新刷新或重启网页。 2. 将整个项目中所包含路径修改为平台无关，打包为Docker项目，可在不同机器部署。 3. 完成了中期所需相关文档和报告。 4. 阅读论文范文，绘制论文所需图。 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 2020年 3 月 27 日 | | |
| 第17－18周记录：   1. 对和弦规则进行了变更，尝试使用贪心法代替动态规划，观察与之前系统的效果差异。 2. 增加测试旋律，比较不同节拍的旋律所产生和弦的效果。 3. 学习更多和声学的相关知识，对系统合理性进行解释。 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 2020年 4 月 10 日 | | |
| 第19－20周记录：   1. 实现了在线读取MusicXML文件并显示乐谱的功能，并实现此功能的web界面及跳转。 2. 对docker镜像进行版本更新。 3. 完成论文的绪论部分。 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 2020年 4 月 24 日 | | |
| 第21－22周记录：   1. 对系统运用的相关技术进行汇总。 2. 对LSTM及Attention机制的原理及优势进行汇总。 3. 对项目鲁棒性进行检查 4. 完成论文的相关技术综述、基于LSTM的和弦编配模型部分。 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 2020年 5 月 8 日 | | |
| 第23－24周记录：   1. 编辑流程图、用例图、运行截图。 2. 对项目错误检查进行测试。 3. 完成论文的和声学规则限制、系统设计、系统实现与测试、结论部分。 4. 修改论文、提交查重。 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 2020年 5 月 22 日 | | |