·多媒体技术及应用· 文章编号:1000-3428(2019)05-0249-07 文献标志码:A 中图分类号:TP391

基于 charRNN 的复音音乐生成方法

王思源,周建国

(武汉大学 电子信息学院,武汉 430072)

摘 要: 在音乐生成过程中,charRNN 方法只能对单音音乐进行训练,而不适用于多个乐器合奏的复音音乐。为使 charRNN 能适用于复音音乐,提出一种将 MIDI 音乐转换为一种基于一定语法规则的音乐描述语言的方法。利用 charRNN 完成文本训练,得到音乐生成模型,基于十二平均律方法获得音乐的统计特性,从而比较不同音乐片段间的差异。实验结果表明,该方法生成的音乐与真实音乐在结构和听感上比较相似,可用于多轨道复音音乐的自动生成。

关键词: 长短期记忆;复音音乐;自动创作;深度学习;计算机音乐

中文引用格式: 王思源,周建国. 基于 charRNN 的复音音乐生成方法[J]. 计算机工程,2019,45(5):249-255,260.

英文引用格式: WANG Siyuan,ZHOU Jianguo. Polyphonic music generation method based on charRNN[J]. Computer Engineering,2019,45(5):249-255,260.

Polyphonic Music Generation Method Based on charRNN

WANG Siyuan,ZHOU Jianguo

(School of Electronic Information,Wuhan University,Wuhan 430072,China)

【Abstract】 In the music generation process,the charRNN method can only train monophonic music,and is not suitable for polyphonic music of multiple instrumental ensembles. In order to make charRNN suitable for polyphonic music,a method of converting MIDI music into a music description language based on certain grammatical rules is proposed. The text training is completed by using charRNN,thus obtaining a music generation model. The statistical properties of the music are obtained based on the theory of twelve-tone temperament method to compare the differences between the different pieces of music. Experimental results show that the music generated by this method is similar to the real music in structure and hearing,and can be used for automatic generation of multi-track polyphonic music.

【Key words】 Long Short-Term Memory(LSTM);polyphonic music;automatic composition;deep learning;computer music

DOI: 10.19678/j.issn.1000-3428.0050596

0 概述

计算机音乐是指运用计算机生成音符序列,可辅助作曲者进行音乐写作。自 21 世纪以来,影视娱乐节目、动画、游戏等需要大量的原创音乐支持[1],但专业的音乐制作人数量有限,音乐制作成本较高,盗版音乐现象猖獗。因此,使用计算机辅助作曲者、编曲者创作音乐受到国内外研究者的关注。

近年来,深度学习技术逐渐成为科技发展的热点,如通过递归神经网络(Recurrent Neural Network,RNN)处理自然语言。其中,charRNN[2]是 RNN 在自然语言处理中的一种典型应用,被用来学习一些特定序列的内部规律,并生成类似序列,如自动作诗、歌词、小说等。而在数字音乐处理中,音符可以看成是特殊的字符对其进行处理。

对于单音音乐,将音符序列转化成字符序列,按照传统方法进行训练。然而在日常生活中人们所听到的大部分音乐都是由多种乐器混合出来的复音音乐,如钢琴、吉他、手风琴等复音乐器,每次可以同时发出多个声音,很难直接将乐谱按时间顺序转化成对应的文本序列。charRNN 可以学习到音乐内在的逻辑关系。

为充分发挥 charRNN 计算成本低、结构简单、训练速度快的优点,本文提出一种复音音乐生成方法。将乐器数字接口(Musical Instrument Digital Interface,MIDI)文件转化为一种音乐描述文本,利用该描述语言的向量表示训练 LSTM 网络模型,根据得到的模型参数进行音乐序列的模拟生成。

基金项目: 国家重点研发计划(2017YFB0504103)。

作者简介: 王思源(1994—),男,硕士研究生,主研方向为多媒体处理、智能信息;周建国,副教授、博士。

收稿日期: 2018-03-05 **修回日期:** 2018-04-27 **E-mail:** wsy7906@qq.com

1 相关工作

目前,计算机音乐生成方法主要分为 3 类:基于概率模型的方法,基于随机选择与组合的方法和基于深度学习的方法。

基于概率模型的方法是通过大量真实音乐数据,得到真实音乐的统计规律,再利用该规律生成新的音乐。如隐马尔科夫模型(Hidden Markov Model,HMM)[3],对于复音音乐来说,可以用隐含状态 S 表示和声连接,观测状态 O 表示旋律,能方便地为歌曲配上和声或者给和声配上旋律。文献[4]利用马尔科夫模型实现自动作曲,但仅生成一些字符表示的乐谱文件。文献[5]使用赞歌训练隐马尔科夫模型,并生成新的音乐,首次将其应用于复音音乐中。文献[6]根据已有旋律,使用隐马尔科夫模型生成对应的伴奏。但马尔科夫状态的演化方式是线性的,较难学习到长期的依赖关系,且只适用于数据较少的情况。

除基于马尔科夫模型的方法外,国内外学者提出基于随机选择与组合的计算机音乐生成方法。文献[7]使用创造力模型来模拟作曲家作曲过程,但只给出相关概念。文献[8]使用元胞自动机的方法来模拟音乐的生成,由于音乐曲谱和元胞自动机的演化有一定的相似性,因此使用简单的规则即可在一定程度上模拟音乐的生成过程[9]。文献[10]利用语法进化和遗传算法来生成旋律。上述方法多数采用随机选择和组合的思想,将音乐划分成一些固定的片段组合,然后使用各种随机方法对其进行重新排列,并根据一定的标准对其进行评价,逐渐筛选出最优结果。上述方法实现手段不同,但均是随机生成片段后进行排列和重组,需大量的实验进行筛选,稳定性较差。

近年来,基于深度学习的方法有不少研究,其通过神经网络来学习音乐特征,利用音乐特征生成新的音乐片段。文献[11]使用 RNN 网络进行自动作曲,同时采用改进式创作方法,但较难学习到音乐片段间的长期依赖性,生成的音乐容易陷入到个别音符的循环当中。文献[12]使用长短期记忆(Long Short-Term Memory,LSTM)网络来学习生成 12 小节布鲁斯片段,避免 RNN 中的梯度消失问题,可学习到音乐序列中音符之间的长期相关性,使生成音乐的动机明确,段落感更强。文献[13]利用 LSTM 生成打击乐轨道。文献[14]使用基于词组的 LSTM 来学习生成爵士和弦和打击乐轨道。然而使用基于词组的 LSTM 来处理复音音乐问题会导致 LSTM 网络中状态过多、运算量较大,在轨道数或声部较多时尤为明显。为简化训练过程和缩短训练时间,本文将多轨 MIDI 文件通过一定的编码方式转化成文本,并利用该文本训练基于字符的 LSTM 网络。

2 MIDI 文件与复音音乐

2.1 MIDI 文件

MIDI 是一种电子通信协议,将音符的时长、音高、时钟等信息以指令形式表示,音乐也可以用音频的形式储存下来。在音频文件中,很难在乐谱的层面上对音乐进行直接编辑。相比于音频文件,MIDI 文件是抽象化的音乐序列信息,有体积小、易编辑等特点,因此非常适合使用机器学习方法进行训练。MIDI 和音频文件的关系,类似于语音和对应文本的关系。

2.2 复音音乐结构

一首音乐由连续多个小节组成,其小节数是 4 或 8 的倍数。节拍是小节的构成单位,将小节按时间平均地划分成几等份。同一首音乐中每个小节的节拍数固定,且决定了音乐的节奏特性。比如最常见的 4/4 拍,指将一小节平均分成四份,其中每一份的长度被定义为一个四分音符。如图 1 所示,若这一小节以四分音符为一拍,前面的二分音符占二分之一小节,为两拍,后面的 2 个四分音符各占一拍,共计 4 拍,符合 4/4 拍的定义。

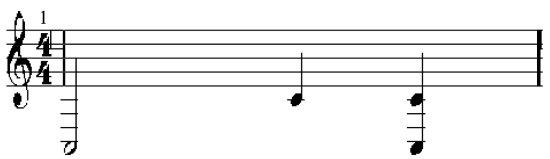

图 1 单一轨道复音音乐

同一个乐器演奏的所有音符同属一个轨道。同一轨道的复音是由同一个乐器同时演奏多个音符。从图 1 可以看出,该乐谱所示的乐器需要在该小节第 4 拍同时演奏 C2 和 C3 这 2 个音。复音音乐中的复音现象还包括了由不同乐器在某一时刻同时演奏的情况。图 2 所示为 2 个轨道的复音音节,在第二小节的第一拍,即箭头所指示的位置,2 个轨道同时演奏一个二分音符。

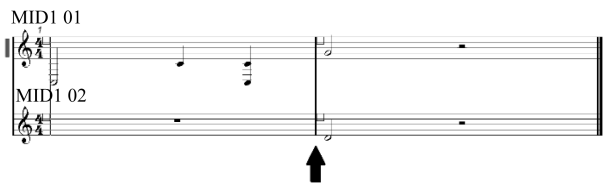

图 2 2 个轨道的复音音乐

音乐往往具有较强的结构特征和循环性,其和弦有着特定的模式[15],如著名的 ii-V-I 进行、流行乐中的卡农进行和 IV-V-iii-vi 进行等,表明音乐存在纵向相关性。另外,由于复音音乐在同一时刻存在多个音符,3 个以上的音符可以构成一个和弦,同时和弦的构成存在横向相关性。

3　基于字符的 LSTM 网络

3.1　递归神经网络与 LSTM

神经网络是根据生物的神经结构设计的计算模型,应用于模式识别、函数逼近、数据压缩等领域。由于神经网络固有的梯度消失[16]问题,RNN 较难学习到时间序列的长期依赖关系,因此 LSTM 引入一个具有复杂结构的 LSTM 单元[17],通过一个单独的内部状态 $C(t)$ 保持长期记忆,可学习到序列内的长期依赖关系,其结构如图 3 所示。

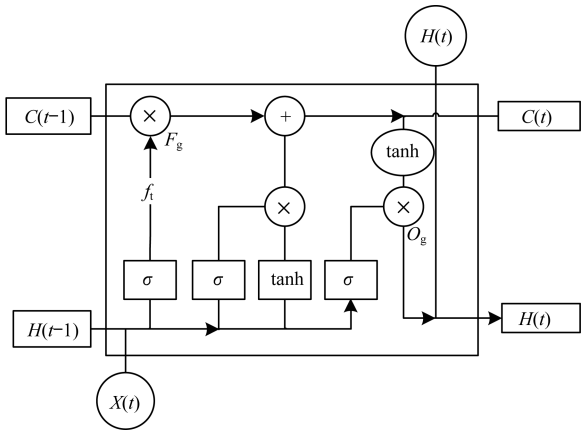

图 3　LSTM 单元结构

3.2　文本描述

本文获得大量真实音乐的 MIDI 文件后,由于不同创作者的习惯不同,曲风不同,在配器和编排风格上也不同,因此从节奏、配器和调性方面对样本进行预处理。

节奏对音乐的微观结构影响较大,为使训练样本中各个片段的节奏保持一致,音乐中 4/4 拍占绝大多数,本文只选择 4/4 拍的音乐作为样本,剔除掉 6/8 拍或 3/4 拍的音乐。不同乐器的编写方式不同,乐器的选择也对音乐的整体结构影响较大。真实音乐往往有许多不同的配器方法,而人声、钢琴和贝司分别决定了旋律层、声层和低音层,对人的听感影响最大。因此,本文使用音乐制作软件将获取到的原始 MIDI 文件剪切到只保留主旋律、钢琴和贝司 3 个轨道,删除掉打击乐、弦乐等多余的轨道。

多数成年人在童年时期没有经过特殊的训练,无法在没有参考音的情况下直接听出声音的频率,因此也无法听出被奏响音的绝对音高。但成年人在一定的训练下,可以获得辨识相对音高的能力,即在已知参考音音高时,可以根据相对音程推算得到它的绝对音高。由于多数人会认为位于不同调上的同一首音乐听感相同,使得把所有的音乐移调到同一个调上再作处理成为可能,即将乐谱归一化。因此,本文将所有音乐统一调整到了 C 调。

由于 charRNN 的输入为字符序列,因此需要将 MIDI 文件转化成文本描述形式。音乐具有节奏性,如果一小节时间长度为 t,音符的起始点和终止点会落在 t/n 位置上,其中,n 一般是 2 次幂或 3 倍的2 次幂。基于上述分析,可以将 t/n 作为编码的单位时间,本文将其定义为一个“word”,用来表示在一个长度为 t/n 的一段时间内所有轨道中,所有被奏响的音符。音乐中所有的音符的时长,都可以用整数倍数个 word 表示。例如,当 $n=32$ 时,一个四分音符可以用 8 个 word 表示,一个八分音符可以用 4 个 word 表示;当 $n=16$ 时,一个四分音符可以用 4 个 word 表示,而一个八分音符可以用 2 个 word 表示。

word 的构造方法如下:

$$m\}p_1p_2\cdots p_m\}b \tag{1}$$

其中,m 表示在单位时间内主旋律轨道所对应的编码,一般对应人声,b 表示贝司对应的编码,p 表示钢琴的编码,}为分隔符。由于钢琴单位时间内可能同时有多个音出现,因此 $p_1 \sim p_m$ 表示从低到高排列的钢琴所有同时奏响的音符。如果音符的结束点和 word 所对应的时间段结束点重合的话,在该音符的后面加上“|”字符表示音符的结束。图 4 所示为 word 构成。在五线谱的第二小节,$n=16$ 可以看作由 16 个 word 构成。其中,第一个 word 的组成音分别为旋律 D4,钢琴为 G3、B3、D4,贝司 G1,对应的 ASCII 码分别为旋律“J”、钢琴“CGJ”、“+”,因此该 word 的编码结果为:

$$\text{J}\}\text{CGJ}\}+ \tag{2}$$

其中,J 对应旋律轨,CGJ 对应钢琴轨,+ 对应贝司轨,}为分隔符。

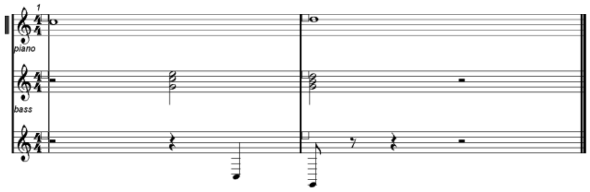

图 4　word 在五线谱上的表示

一首音乐一般由 m 个小节构成,而小节的长度固定,如果每个小节包含 n 个 word,那么 $n \times m$ 个 word 构成一首音乐,且每个 word 之间用空格字符隔开。当一个 word 的时长内没有音符奏响时,该 word 用“}}”表示。

3.3　输入输出向量表示

由于神经网络输入是向量形式,文本格式的训练数据无法直接输入到网络中,本文将训练数据用向量来表示,输入音符 x_n 转化为输入向量 $\boldsymbol{X}_n$。各个音符之间没有明确的数量关系,对其采用 One-hot 编码。MIDI 音符的编号共有 128 个,如果经过编码后的输入向量 $\boldsymbol{X}_n$ 共有 128 维,那么步数为 m 的输入数

据的一个 batch 可表示为$[x_1,x_2,\cdots,x_m]$。

为使 LSTM 网络可以学习到音符、小节乃至乐段之间的相关性,用$\boldsymbol{X}_n$所对应音符在训练数据中相邻的下一个音符的向量$\boldsymbol{X}_{n+1}$作为输出向量$\boldsymbol{Y}$,一个步数为 m 的输出 batch 可表示为$[x_2,x_3,\cdots,x_{m+1}]$,其具体示例如图 5 所示。其中,所有构成的输入输出样本对已经使用实斜线连接,当 m 等于 10 时,输出 batch 截取为输入 batch 向后偏移一步的等长片段。

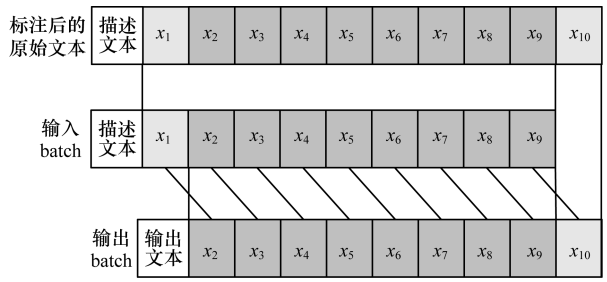

图 5 输入输出向量表示方法示例

经过向量化表示后,标注后的描述文本可以表示为若干组 128 维输入向量 $\boldsymbol{X}$ 和 128 维输出向量 $\boldsymbol{Y}$ 构成的样本对。

3.4 网络模型训练

本文网络结构由 3 个主要部分构成,分别是输入层、隐藏层(LSTM 层)、输出层。其中,隐藏层的层数随着实验变量的设置而改变。从 128 个向量中选取一个作为输出,因此使用 softmax 层作为输出层,同时将输出层与 LSTM 层进行全连接。将上述步骤中获得输入输出向量作为训练数据输入到 LSTM 网络中,反复迭代并计算误差,将误差反向传播更新 LSTM 网络的权值参数,其过程如图 6 所示。LSTM 网络每次读入一个音符作为输入,且该音符的下一个音符作为输出。

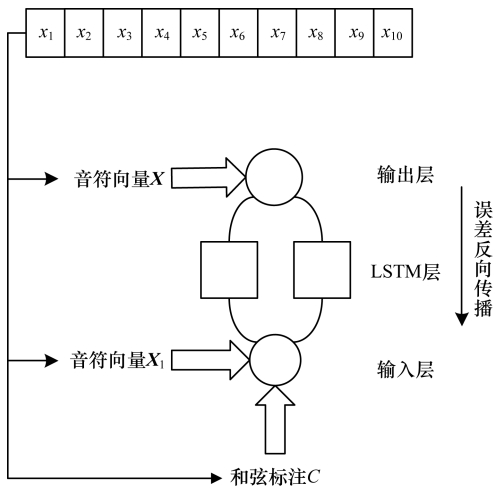

图 6 网络模型训练

网络模型训练具体算法描述如下:

算法 1 网络模型建立

输入 由 ASCII 字符构成的训练样本

输出 音乐序列模型参数(LSTM 的权值参数)

1. 随机初始化 LSTM 模型参数;

2. 将训练数据进行 One-Hot 编码,并送入输出层,进行前向传播;

3. 根据输出层的结果与目标输出进行比较,对误差进行反向传播;

4. 完成规定的迭代次数,最终将网络的模型参数输出。

在前向传播过程中,输出门的计算公式为:

$$\boldsymbol{w}^t=\sigma\left(\sum \boldsymbol{w}_{iw}\boldsymbol{x}_i^t+\sum \boldsymbol{w}_{hw}\boldsymbol{b}_h^{t-1}+\sum \boldsymbol{w}_{cw}\boldsymbol{s}_c^t\right) \tag{3}$$

最终输出为:

$$\boldsymbol{c}^t=\boldsymbol{w}^t\times\boldsymbol{\sigma}(\boldsymbol{s}_c^t) \tag{4}$$

其中,$\boldsymbol{w}$ 表示连接各个部分的权值矩阵,下标表示连接的输入输出,$\boldsymbol{x}_i^t$ 表示当前时刻的输入,$\boldsymbol{s}_c^t$ 表示当前时刻的单元状态。

在后向传播的过程中,状态量为:

$$\boldsymbol{\varepsilon}_s^t=\boldsymbol{b}_w^t\boldsymbol{h}'(\boldsymbol{s}_c^t)\boldsymbol{\varepsilon}_c^t+\boldsymbol{b}_\varphi^{t+1}\boldsymbol{\varepsilon}_s^{t+1}+\boldsymbol{w}_{cl}\delta_l^{t+1}+\boldsymbol{w}_{c\varphi}\delta_\varphi^{t+1}+\boldsymbol{w}_{cw}\delta_w^t \tag{5}$$

其中,$\boldsymbol{\varepsilon}_c^t\overset{def}{=}\dfrac{\partial L}{\partial \boldsymbol{b}_c^t}$,$\boldsymbol{\varepsilon}_s^t\overset{def}{=}\dfrac{\partial L}{\partial \boldsymbol{s}_c^t}$,$L$ 表示损失函数。

为提高学习速度,本文使用交叉熵作为损失函数,其表示预测结果和真实结果的偏离程度,表达式为:

$$H(p,q)=\sum_i p(i)\times\lg\frac{1}{q(i)},i=1,2,\cdots,m \tag{6}$$

其中,$p(i)$表示样本中的真实概率分布,$q(i)$表示实际输出的概率分布,m 表示所有音符编码后的种类数,同时将最终迭代次数设置为 1 000,并记录损失函数的值。另外,需要记录学习到的权值参数,以便使用该模型生成新的音乐序列。在训练 LSTM 网络时,使用 dropout 策略[18]防止过拟合,且 dropout 率设置为 50% 。

3.5 音乐序列生成

训练完后,使用算法 1 中生成的模型来自动生成新的音乐序列。LSTM 网络每次读入一个音符,根据该音符和输入的音符推测下一个可能出现的音符,如图 7 所示。

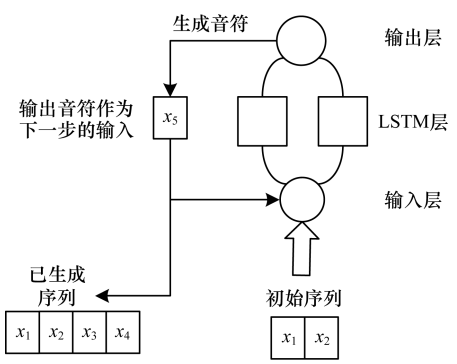

图 7 音乐序列生成过程

音乐序列生成过程如算法 2 所示。

算法 2　音乐序列生成

输入　音乐序列模型参数,初始序列

输出　已生成音乐序列 *S*

1. 随机选择一个长度为 m 预定的初始序列作为网络的初始输入;

2. 按顺序取出初始序列 I 中的一个音符,通过 One-Hot 编码后,作为输入向量 X 输入到 LSTM 网络中;

3. 在初始序列 I 取完后,将 I 作为生成序列 S 的开头部分,否则返回第 2 步;

4. 通过正向传播,从输出层得到输出向量 Y,将 Y 作为下一次正向传播的输入向量 X;

5. 将 Y 解码为音符 y,并直接添加在已生成序列 S 的末尾,若没有到指定的循环次数,则返回第 4 步;

6. 输出所获得的生成音乐序列 S,并通过脚本将 S 转化成 MIDI 序列作为最终结果。

算法 2 中所提到的初始序列既可手动配置,也可随机生成。和弦序列为作曲者指定,生成的文本文件通过脚本转化为 MIDI 文件后,即可进行播放。

4　实验结果与分析

本文基于 charRNN 的复音音乐生成方法流程如图 8 所示。

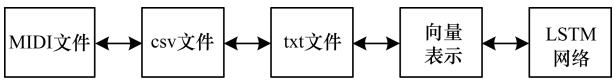

图 8　基于 charRNN 的复音音乐生成方法流程

4.1　实验设置

本文实验运行在处理器为 Intel i5-6500 3.20 GHz,内存 8 GB 的计算机上,预处理时的音乐编辑软件为 Cubase 5.1.2,实验所使用编程语言为 Python 3.6.1 和 C#,并使用基于 Python 的深度学习库 tensorflow 来完成程序设计。

训练数据选自 MIDI 论坛,所有 MIDI 文件均为 3 个轨道,4/4 拍,且没有转调的流行音乐。在 MIDI 文件转化为对应的描述文本后,其长度大约 6 MB。本文使用 John Walker 的 Miditocsv 开源工具把 MIDI 文件转化为 csv 文件,使用 Python 脚本将 csv 文件转化成 txt 文件进行处理。

4.2　LSTM 单元数

LSTM 单元数是每一层 LSTM 层中单元个数,即网络的宽度。LSTM 单元数越多,学习能力一般会越强。本文采用宽度分别为 512、1 024、2 048 这 3 种规格的 LSTM 层进行实验,层数选择为 2 层,样本采用旋律、钢琴、贝司 3 个轨道,时间划分为 32 words/bar,LSTM 单元数与训练速度的关系如图 9所示。

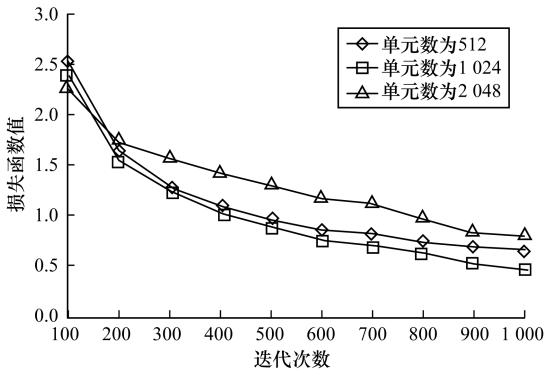

图 9　不同 LSTM 单元数损失函数值对比

从图 9 可以看出,当迭代次数为 1 000 时,单元数为 1 024 时可以获得最好的训练效果。迭代次数越多,网络可能会获得较好的训练效果,但其提升效果并不明显,且训练时间较长。因此,本文设置单元数为 1 024。

4.3　LSTM 层数

LSTM 层数是 LSTM 网络中隐层层数,在前馈神经网络中,层数越多,收敛的速度越慢,学习的效果就越好。本文在每层 LSTM 单元数为 1024 的情况下,选取 1、2、3 层分别进行实验,结果如图 10 所示。

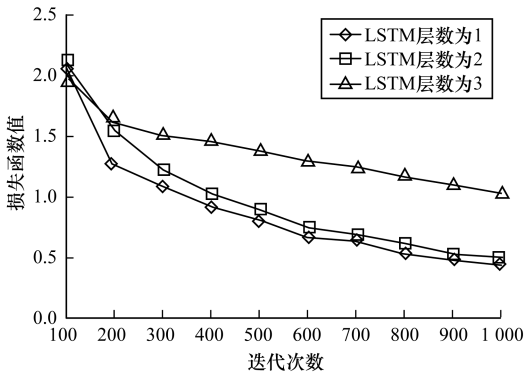

图 10　不同 LSTM 层数损失函数值对比

从图 10 可以看出,LSTM 层数对实验结果的影响较大。在每层单元数为 1 024 的情况下,当层数为 1 或者 2 时,2 层实验结果要略好于 1 层。但当层数为 3 时,损失函数下降速度明显变慢,效果比其他 2 种层数差。因此,在使用 RNN 进行音乐生成的时候,最好选择双层 LSTM 网络。

4.4　word 尺度

本文分别将 MIDI 文件转化为尺度为32 words/bar 和 16 words/bar,测试 2 种情况下的文本文件对训练结果的影响。一个小节分配的 word 越多,编码后获得的文本文件会越长,对原来的 MIDI 文件的描述也更精确,结果如图 11 所示。

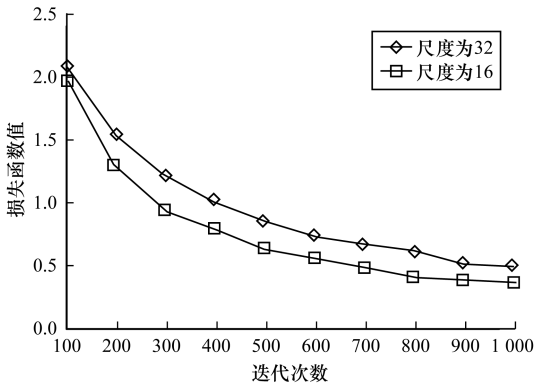

图 11 不同 word 尺度损失函数值对比

从图 11 可以看出,16 words/bar 结果优于 32 words/bar。在编码时,乐词使用更大的时间尺度,会获得更好的实验结果,但会失去一些细节信息,如时值较短的音符等,使得生成的音乐缺少灵活性。

4.5 轨道数量

由于音乐体裁和类型的多样性,每首音乐所使用的乐器也各有不同。不同的乐器有着不同的编配方式,因此乐器的选择对训练样本影响较大。本文选择 2 个轨道进行对比实验,如图 12 所示。

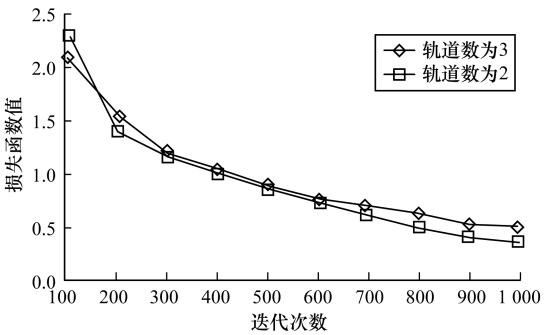

图 12 不同轨道数量损失函数值对比

从图 12 可以看出,训练样本中所包含的轨道数越多,训练效果越差,主要是因为轨道数越多,和声织体就会越复杂,生成的文本语法更复杂,结果与样本之间会产生较大的偏差。

4.6 统计特性分析

十二平均律[19]是音乐中的一种定调方法,将物理频率呈 2∶1 的 2 个单音之间的音程划分为 12 个半音,相隔 12 个半音的 2 个音称为一个八度,这2 个音互为八度音,八度音在人类的耳中听感十分相似。十二平均律可表示为:

$$P_n = P_a(\sqrt[12]{2})(n-a) \tag{7}$$

其中,P_n 表示待计算的音符的绝对音高,P_a 表示参考音高,n 和 a 分别表示待计算音符和参考音符从左数起的位次。

为衡量不同音乐片段的相似性,本文将音乐中出现的所有音符映射到同一个八度上,从而可以得到不同音乐的音高统计特性。根据上述理论,本文对音乐的音高统计特性分析如算法 3 所示。

算法 3 音高统计算法

输入 音乐描述文本

输出 音高统计特性分布

1. 取第一个音符 N;
2. 获得 N 中的音符长度 l,和 N 对应的 MIDI 序号 k;
3. 获得 N 分类序号 Xi,方法为 Xi = k/12;
4. 音符类别 Xi 下的总时长 T(Xi)在原有基础上增加 l;
5. 取下一个音符 N,执行算法第 2 步,直到取完音乐中所有的音符;
6. 其中 X_i 的取值 0 ~ 11 分别映射到 C,C#,…,B 等12 种音符类型。通过 $T(X_i)/\sum T(X_i)$ 可以求出各个音符类型所占的比例。

在十二个音符类别中,出现频率最高的是 C(唱名 do)、D(唱名 re)、E(唱名 mi)、G(唱名 so)、A(唱名 la) 5 种音,分别对应民族调式的五声音阶,这主要由于选取训练样本时使用中文流行歌曲,而中文音乐大部分都有一定的民族特色,这 5 个音占主导地位。本文使用 2 层 LSTM,且每层设置为 1 024 个 LSTM 单元,迭代过程中的音乐统计特性变化如图 13所示。其中,a、b、c、d 分别是迭代 200 次、400 次、600 次和 1 000 次后的结果。当迭代次数为 200 时,真实音乐音高统计如图 14 所示,C、E、G、A 音占优势,但几个非自然音阶(有“#”号)的音符还没有出现。当迭代次数为 400 时,出现部分非自然音阶音符,同时各种音符之间的比例符合真实分布。当迭代次数为 1 000 时,所有的音高均已经出现,且最接近于真实样本中的分布。本文使用 dropout 策略防止过拟合,最终生成的音乐统计特性和真实样本中不完全相同,但较接近真实分布。

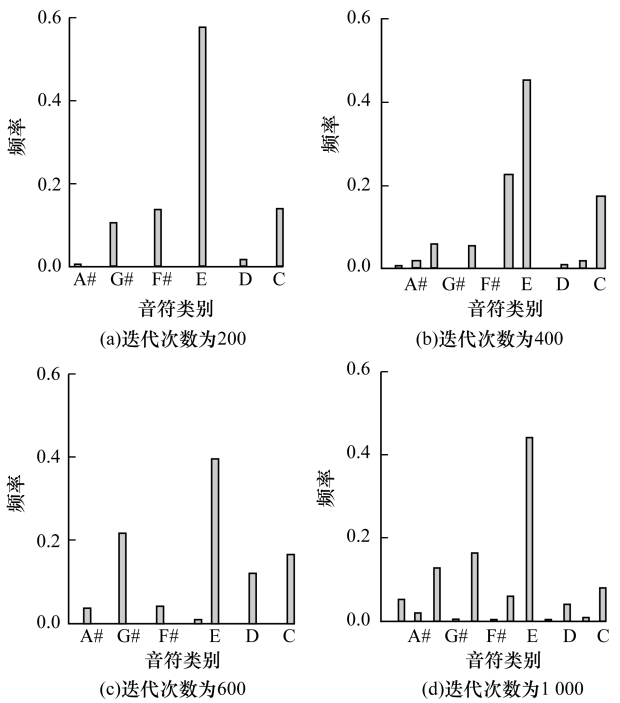

图 13 不同迭代次数下的统计特性

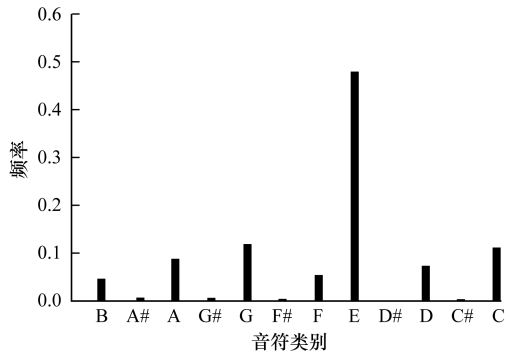

图 14　真实音乐音高统计特性

4.7　听感评测

本文使用准确度(P)、精确度(A)、召回率(R)评估实验结果,计算公式如下所示:

$$P = \frac{TP + TN}{TP + FN + FP + TN}$$

$$A = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

其中,TP 表示判断为真、实际为真,TN 表示判断为假、实际为假,FP 表示判断为真、实际为假,FN 表示判断为假、实际为真。

生成音乐主要分为旋律、低音、和声 3 个声部,且具有明确的旋律性。上述方法生成 5 个音乐片段,与 5 个从真实音乐中截取出的音乐片段混合,让 4 个非音乐专业人士作为测试人员进行辨别,且没有事先告知其真实音乐和合成音乐的比例。

统计结果可得 $TP = 17$、$TN = 6$、$FP = 14$、$FN = 3$,本文方法的精确度为 54.8%,准确度为 57.5%,召回率为 85%,转移率为 30%。可以看出,精确度和准确度均在 50% 左右,表明有判断错误的音乐,而召回率在 80% 以上,表明大部分正例判断正确,负例大部分都被误判为正例。因此,多数人将真实音乐判断正例,同时把大部分合成音乐判断成正例,验证了本文方法的有效性。

5　结束语

本文提出一种复音音乐的生成方法。该方法将 MIDI 片段转化为文本文件,利用 LSTM 网络对其进行训练,得到音乐生成模型。基于十二平均律的音高统计方法,设计人耳听感测试,结果表明,本文方法生成的音乐与真实音乐相似度较高,且音乐片段具有旋律性,符合和声规则。下一步将研究音乐生成模型的连贯性及节奏性。

参考文献

[1] 余立功,卜佳俊,陈纯. 基于内外概率算法的音乐节奏自动生成[J]. 浙江大学学报(工学版),2005,39(12):1969-1972,1983.

[2] GRAVES A. Generating sequences with recurrent neural networks[EB/OL]. [2018-02-01]. https://arxiv.org/pdf/1308.0850.pdf.

[3] BAUM L E,PETRIE T,SOULES G,et al. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains[J]. The Annals of Mathematical Statistics,1970,41(1):164-171.

[4] HILLER L, ISAACSON L M. Experimental music composition with an electronic computer[M]. Westport, USA:Greenwood Publishing Group Inc.,1979.

[5] ALLAN M,WILLIAMS C K I. Harmonising chorales by probabilistic inference [C]//Proceedings of the 17th International Conference on Neural Information Processing Systems. Cambridge,USA:MIT Press,2005:25-32.

[6] SIMON I, MORRIS D, BASU S. MySong: automatic accompaniment generation for vocal melodies [C]// Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York, USA: ACM Press,2008:725-734.

[7] JACOB B L. Algorithmic composition as a model of creativity[J]. Organised Sound,1996,1(3):157-165.

[8] BILOTTA E, PANTANO P, TALARICO V. Synthetic harmonies :an approach to musical semiosis by means of cellular automata [C]//Proceedings of the 7th International Conference on Artificial Life. Cambridge, USA:MIT Press,2002:153-159.

[9] 王存睿,段晓东,刘向东,等. 基于 Hilbert 映射的元胞自动机音乐生成算法[J]. 微电子学与计算机,2010,27(1):5-8.

[10] DE LA PUENTE A O,ALFONSO R S,MORENO M A. Automatic composition of music by means of grammatical evolution[C]//Proceedings of Conference on APL. New York,USA:ACM Press,2002:148-155.

[11] LEWIS J. Algorithms for music composition by neural nets Improved CBR paradigms [EB/OL]. [2018-02-04]. https://quod.lib.umich.edu/cgi/p/pod/dod-idx/algorithms-for-music-composition.pdf?c=icmc;idno=bbp2372.1989.044;format=pdf.

[12] ECK D, SCHMIDHUBER J A first look at music composition using lstm recurrent neural networks[EB/OL]. [2018-02-04]. http://people.idsia.ch/~juergen/blues/IDSIA-07-02.pdf.

[13] LAMBERT A J, WEYDE T, ARMSTRONG N. Perceiving and predicting expressive rhythm with recurrent neural networks [EB/OL]. [2018-02-01]. http://openaccess.city.ac.uk/16489/.

[14] CHOI K,FAZEKAS G,SANDLER M. Text-based LSTM networks for automatic music composition [EB/OL]. [2018-02-01]. https://arxiv.org/pdf/1604.05358.pdf.

(下转第 260 页)

3 结束语

本文提出一种基于微调策略的人脸活体检测方法。通过融合运动信息和图像质量信息,提高人脸活体检测的泛化力,利用卷积神经网络进行训练,并采用网络微调技术判断真假活体。实验结果表明,该方法相对其他方法具有较高的人脸活体检测准确度和泛化力。下一步可将动态纹理特征作为网络输入,并将其应用到并行计算上。

参考文献

[1] PAN Gang, SUN Lin, WU Zhaohui, et al. Eyeblink-based anti-spoofing in face recognition from a generic webcamera[C]//Proceedings of the 11th International Conference on Computer Vision. Washington D. C., USA: IEEE Press, 2007: 1-8.

[2] KOLLREIDER K, FRONTHALER H, FARAJ M I, et al. Real-time face detection and motion analysis with application in "liveness" assessment[J]. IEEE Transactions on Information Forensics and Security, 2007, 2(3): 548-558

[3] BAO Wei, LI Hong, LI Nan, et al. A liveness detection method for face recognition based on optical flow field[C]// Proceedings of International Conference on Image Analysis and Signal Processing. Washington D. C., USA: IEEE Press, 2009: 233-236.

[4] 任安虎,刘贝. 基于 Adaboost 的人脸识别眨眼检测[J]. 计算机与数字工程,2016,44(3):521-524.

[5] SINGH A K, JOSHI P, NANDI G C. Face recognition with liveness detection using eye and mouth movement[C]//Proceedings of International Conference on Signal Propagation and Computer Technology. Washington D. C., USA: IEEE Press, 2014: 592-597.

[6] ANJOS A, CHAKKA M M, MARCEL S. Motion-based counter-measures to photo attacks in face recognition[J]. IET Biometrics, 2014, 3(3): 147-158

[7] MAATTA J, HADID A, PIETIKAINEN M. Face spoofing detection from single images using micro-texture analysis[C]//Proceedings of International Joint Conference on Biometrics. Washington D. C., USA: IEEE Press, 2011: 1-7.

[8] 方天红,陈庆虎,廖海斌,等. 融合纹理与形状的人脸加权特征[J]. 武汉大学学报(信息科学版),2015,40(3):321-326,340.

[9] KIM Y, NA J, YOON S, et al. Masked fake face detection using radiance measurements[J]. Journal of the Optical Society of America A-Optics Image Science and Vision, 2009, 26(4): 760-766.

[10] ZHANG Zhiwei, YI Dong, LEI Zhen, et al. Face liveness detection by learning multispectral reflectance distributions[C]//Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition and Workshops. Washington D. C., USA: IEEE Press, 2011: 436-441.

[11] EASLEY G, LABATE D, LIM W Q. Sparse directional image representations using the discrete shearlet transform[J]. Applied and Computational Harmonic Analysis, 2008, 25(1): 25-46.

[12] 许晓. 基于深度学习的活体人脸检测算法研究[D]. 北京:北京工业大学,2016.

[13] KUTYNIOKG, LABATE D. Shearlets: multiscale analysis for multivariate data[M]. [S.l.]: Birkhäuser Basel, 2012.

[14] CHINGOVSKA I, ANJOS A, MARCEL S. On the effectiveness of local binary patterns in face anti-spoofing[C]//Proceedings of International Conference of Biometrics Special Interest Group. Washington D. C., USA: IEEE Press, 2012: 183-194.

[15] ZHANG Zhiwei, YAN Junjie, LIU Sifei, et al. A face antispoofing database with diverse attacks[C]// Proceedings of the 5th IAPR International Conference on Biometrics. Washington D. C., USA: IEEE Press, 2012: 26-31.

编辑 赵 辉

(上接第 255 页)

[15] 李雄飞,冯婷婷,骆实,等. 基于递归神经网络的自动作曲算法[J]. 吉林大学学报(工学版),2018,48(3):866-873.

[16] HOCHREITER S, BENGIO Y, FRASCONI P, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies[EB/OL]. [2018-02-01]. https://www.bioinf.jku.at/publications/older/ch7.pdf.

[17] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[18] LI Rongjian, ZHANG Wenlu, SUK H I, et al. Deep learning based imaging data completion for improved brain disease diagnosis[C]//Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention. Berlin, Germany: Springer, 2014: 305-312.

[19] KLERK D D. Equal temperament[J]. Acta Musicologica, 1979, 51(1): 140-150.

编辑 赵 辉