

基于 Char-RNN 的音乐生成实践与算法作曲尝试

□文 / 池雨峰 指导老师：张志斌、王汝昕



池雨峰

人大附中高二学生。有丰富的机器人赛事经历，作为队长连续参与了5年的美国FRC (FIRST Robotics Competition) 机器人竞赛，擅长电子工程和自动化控制领域。半年前开始接触机器学习，先后学习了keras、tensorflow、tflearn等框架，创造了首个带有人工智能的虚拟人物“奈”。同时，在Kaggle的mnist赛事中取得了全球第1149的成绩。

现任英特尔中国战略合作与创新业务部总监，负责英特尔中国机器人、人工智能以及智慧未来等战略性新兴产业的生态创新和战略合作。1986-1993 年在北京邮电大学学习，并先后获得通信工程学士学位以及电路与系统硕士学位。2000 年加入英特尔中国公司，曾主管过诸多部门，包括通信和嵌入式中国区业务、技术支持业务部以及产品市场业务部等，并在美国总部工作两年，从事战略产品的规划和市场工作。从 2015 年开始推进前沿新兴产业如机器人和人工智能的生态建设，短短的两年时间搭建了中国有影响力的机器人创新生态，快速汇聚了上千支团队和上万活跃的开发 者，配合举办机器人创新生态峰会以及机器人创新挑战赛等活动，有力地推动了机器人有机生态的创新与产业应用。2017 年至今，积极推进中国人工智能产业创新联盟的建设，搭建中国人工智能开放平台，以及举办全球人工智能创新大赛等一系列举措，进一步加强对人工智能战略性新兴产业的影响力和领导力。另外，也关注物联网、智能制造和智能未来等新领域的产业发展趋势和机遇。

张志斌



一、项目背景

英特尔专注于技术产品创新，同时也积极支持广泛开发者的创新能力培养。英特尔创新加速器，机器人创新生态，搭建开放的人工智能平台，举办人工智能以及机器人创新大赛等，这些努力，有助于创新能力的培养以及创新潜力的释放，受益者不仅涵盖产业界创新者和科研机构研究人员，也包括学校学生。

2018年5月23日，在有着103年历史的旧金山艺术宫中，英特尔的新晋科技大会——人工智能开发者大会（简称“AIDC”）拓宽人工智能生态，在“软件+硬件+生态”合作的基础上，也更加关注与AI开发者的互动。在AIDC现场，英特尔展示了一段用神经计算棒来赋能的AI作曲演示。英特尔中国战略合作与创新业务部在人工智能和机器人等前沿科技领域推动创新的支持，不仅提供适合学校和学生研究使用的开发平台，而且深入到学生创新项目的辅导，支持那些对前沿科技有浓厚兴趣的学生开展人工智能和机器人的研究项目，培育学生创新能力和兴趣，加速人工智能和机器人前沿科技在中国教育的普及推广。其中，来自北京人大附中高二年级的学生池雨峰，便是得到英特尔支持的人工智能领域的一位小有名气的“开发者”。他对人工智能有着浓厚的兴趣和爱好，在英特尔创新业务部的支持指导下，开展了颇有趣味和成效的研究项目。下面就是他自主研究完成的科研项目，希望这样的案例影响到更多的学生，深入到更多的学校，从而营造一个中国人工智能时代全面创新的热烈氛围。

二、基础原理

循环神经网络（RNN，Recurrent neural Network）是上世纪80年代末提出的一种神经网络结构模型，这种网络的本质特征是处理单元之间内部的反馈连接和前馈连接。从系统观点看，它是一个反馈动力系统，在计算过程中体现过程动态特性，比前馈神经网络具有更强的动态行为和计算能力。循环神经网络现已成为国际上神经网络专家研究的重要对象之一，其原理如图1所示：

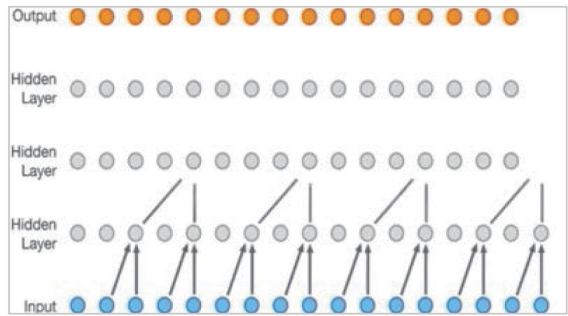


图 1：循环神经网络原理

三、关键词

- RNN 循环神经网络
- LSTM 长短期记忆网络
- Algorithmic Composition 算法作曲

-MIDI Generation 音乐生成

四、AI 作曲背景

1843 年，Ada Lovelace 曾在作品中写道：“引擎可以编写精致而科学的音乐作品，作品很复杂，范围很广。”1952 年，Lejaren Hiller 发现将计算机的控制变量由几何数转换成音符，这些代码就完全可以用于谱曲。他利用计算机进行了一系列作曲技法实验。1957 年，历史上第一支完全由计算机生成的音乐作品诞生，即弦乐四重奏《依利亚克组曲》（Illiad Suite）。1960 年，俄罗斯研究者 R.Kh.Zaripov 发行了全世界第一篇关于用「Ural-1」计算机进行算法音乐作曲的论文。1980 年，执教于美国加州大学圣克鲁兹分校艺术系的大卫·库佩（David Cope）从编写作曲程序出发，着迷于通过基于乐理的技法重组来创造模拟不同过往音乐家风格的音乐。他开发的作曲软件 Experiments in Musical Intelligence（EMI）能通过特殊的匹配过程，专门模仿约翰·塞巴斯蒂安·巴赫的风格。

在卢森堡和伦敦，Pierre Barreau、Denis Shtefan、Arnaud Decker 和 Vincent Barreau 创造了一种名为 Aiva 的 AI，并教它如何创作古典音乐，其作品被用作电影导演、广告公司，甚至是游戏工作室的配乐。Google Brain 研发了一款名为 Magenta 的 AI 作曲软件，利用特定的音乐风格生成新的音乐。索尼公司开发了 AI 谱曲工具 Flow Machines，并用它制作了一首名叫《Daddy’s Car》的歌曲。

从基础理论设计与数学逻辑同构并进行符号化组织的角度来看，音乐虽然是一门艺术，却有很强的可计算性，音乐模式背后蕴含着数学之美。常规的作曲技法，如旋律的重复、模进、转调、模糊、音程或节奏压扩，和声与对位中的音高纵横向排列组合，配器中的音色组合，曲式中的并行、对置、对称、回旋、奏鸣等等，都可以被描述为单一或组合的算法。这些从本质上决定了 AI 技术可以较好地应用到音乐创作上。而巴赫的乐曲具有结构工整的和谐美，其内在严密的逻辑性使其成为算法作曲实践的良好训练材料。

五、创作动机

我喜欢弹琴，在学习钢琴曲的过程中发现，这是一个很适合用 ML 来做的项目——有固定的数据大小（88 个键固定不变）以及直接的数据来源（MIDI 文件）。同时，我平时会自己录制一些弹琴的视频上传。但是视频的 BGM 经常会侵权，所以我就想做一个能够自行创作音乐并且 copyright-free 的项目。

一遍遍练习却总是不小心弹错音，眼前琴谱上的一个个音符就像一个个数据点，我却无法找到其中的规律。苦恼之余我意识到，这种工序不如交给机器去完成。经过一定的学习了解，我在 Youtube 上面发现，有人通过将 MIDI 转化为字符的方式实现了类似的想法。因此，我借鉴了此方法并对其进行了改良。

六、尝试过程

准备

- Windows 10 操作系统
- Python 3.6.4
- Tensorflow-gpu 1.7.0

数据获取

首先，我找到了 <http://www.jsbach.net/midi/index.html> 网站，下载了其中 `The English Suites` 系列中的所有钢琴曲。解压 zip 文件之后，我们获得了 48 首巴赫作品的 MIDI 文件，总共 480KB，98 分钟。之后需要用 `Midicsv.exe` 这个工具将 `.mid` 文件转化为 Python 可以处理的 `.csv` 文件。下载地址为：<http://www.fourmilab.ch/webtools/midicsv/>。

可以通过几行 Python 代码调用这个工具，实现批量转化：

```
```Python
import os

def cvt_midi_2_csv(filepath):

 midi_list = list(os.walk(filepath))[0][2] # 遍历文件夹下的所有文件，输出文件名

 for midi in midi_list:

 csv = midi.replace('.mid', '.csv')

 os.system("tool\Midicsv.exe .\midi\\"+midi+" .\csv\\"+csv) # 调用 midicsv.exe

cvt_midi_2_csv('midi') # 这里是储存 midi 的文件夹
```
```

格式转换

虽然现在我们得到了 `.csv` 格式的文件，但是文件中含有很多的多余信息，如：

```
```csv
...
```

```
2, 2160, Note_on_c, 0, 81, 107
2, 2400, Note_on_c, 0, 76, 107
2, 2520, Note_on_c, 0, 73, 107
2, 2880, Note_on_c, 0, 69, 107
2, 2976, Note_on_c, 0, 81, 0
2, 3000, Note_on_c, 0, 80, 107
2, 3072, Note_on_c, 0, 76, 0
...
...
```

因此需要重新设计一种文件格式。

我们的需求有：

- 尽量减少不必要 / 重复的字符
- 保留原有的时间格式
- 保留多个音的关系（同时弹奏等等）

受到 Youtube 上 [carykh](https://www.youtube.com/channel/UC9z7EZAbkphEMg0SP7rw44A) 的启发，我决定使用一个 ASCII 字符来代表一个音（[ascii-code](https://www.ascii-code.com/) 是一份很好的对照表），并且将整个乐曲分解为一个个的小片段，片段之间用空格隔开。这样就可以把一首乐曲转化为一段字符串。

Python 同样可以很容易地完成这个任务，于是，我们得到了一个转化后的十分精简的 `input.txt` 文件。

逻辑结构

程序调用功能实现：

- `train.py` 用于模型训练
- `sample.py` 用于样例分析
- `char\_rnn\_model.py` 导入 Char\_RNN 模型

输出文件夹逻辑架构：

```
...
dist
├── data
│ ├── input.txt # 训练文本格式重排
│ └── dependencies # 文件夹包含 MIDI-CSV 格式转换文件
├── Csvmidi.exe
```

```
├── Midicsv.exe
├── midi # 文件夹包含训练所需的 MIDI 文件
├── output # 文件夹包含保存的模型
├── scripts # 其他依赖文件
...
```

注意：`train.py` 默认文件使用 UTF-8 代码格式。

模型训练

出于训练难易度和环境配置的角度，我选择使用由 sherjilozair 提供的 [char-rnn-tensorflow](https://github.com/sherjilozair/char-rnn-tensorflow)。这是由 Andrej Karpathy 著名的 Char-RNN 移植到 TF 平台的项目。Char-RNN 是一个使用多层循环神经网络的特征序列预测模型 (RNN, LSTM or GRU)。下载并解压完项目之后，在命令行中输入：

```
...
powershell
python .\train.py --data_dir data/bach --save_dir bach_save --rnn_size 256 --num_
layers 3 --num_epochs 100 --save_every 500
...
```

之后就是一段等待的时间。

最终的结果如图 2 所示：

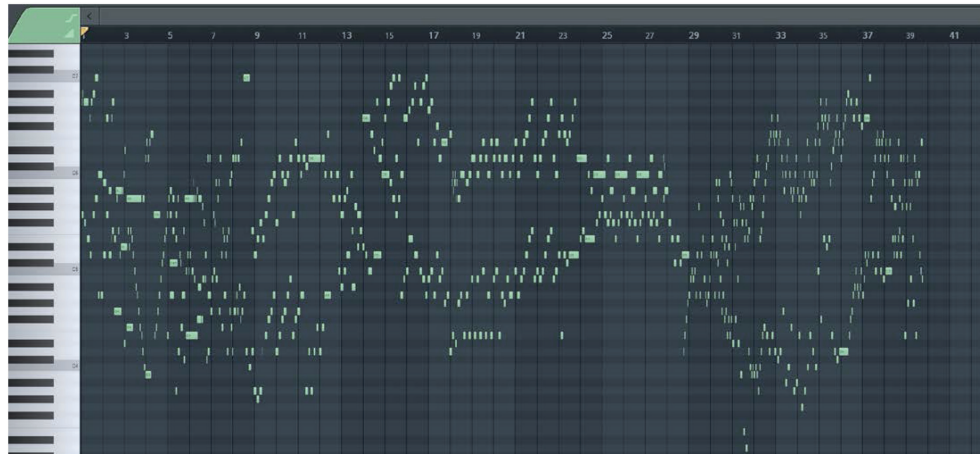


图 2：基于 Char-RNN 的音乐生成实践与算法作曲尝试结果

七、总结

有效的重组必须以大量的音乐分析为基础。获得清晰的风格定义，再小心翼翼地选

取不损伤音乐性的最基本的风格标签元素为单位（一般以多个音高形成的旋律单元或和弦进行为单位）进行重组，依此产生一份“同样风格”下的新作品。目前 AI 作曲领域研究的方向主要在深层特征的提取与应用和混合系统的构造上，其仍面临以下几个难点：

1. 音乐的创作表示。大多数音乐组曲过程较为复杂，现有特征提取机制尚不能够精确掌握一部作品的全部信息，比如，作品中与乐句、调性等相关的音乐信息一般无法体现。如何精准表示音乐的细部特征，提取音乐的深层逻辑，建立表层结构和深层逻辑的关系，是 AI 作曲亟待解决的基础性问题。
2. 作品的质量评估。人类对音乐作品的评判往往比较感性，因此作曲系统中的质量评估机制是一个非常重要的部分，它往往会引导创作的方向，甚至最终决定作品的成败。把人类的审美观用机器能够理解的语言描述出来，并建立有效的评判标准，是研究人员首要面对的问题。
3. 艺术的审美评价。通过 AI 技术实现的作曲终究缺少艺术的“灵魂”，无法进行灵感创作。未来通过 AI 制造出能够和人类协同进行乐曲创作的程序，从数据的视角来诠释人类的音乐艺术真谛，这才是算法作曲与人类智慧有机结合，符合普世价值观与审美体系的发展方向。

