创新实验课

课题结项报告

课题名称:智能音乐助手

课题名称（英文）：AI-MUSIC

课题组长：冯韵菱

课题组成员：冯韵菱　张靖鸿　许宏涛

报告时间：2020 年　12 月 31 日

目录

一、 项目创意来历及项目意义

1.1 项目创意

目前在乐器的学习中普遍存在着很多问题，初学者在自学乐器（吉他）都会遇到一些困难，如音调不准，按弦位置错误，有错音杂音等等。

而对于这个问题，市场上已经有了一些解决方案，比如现有的智能尤克里里，但我们经过调查使用者的反馈得出的结论是：尽管产品已经很好用了，它们用 APP 搭配物理硬件（LED）来简化学习过程，但并不能让用户像广告上说的“7 天”学会这门乐器，用户总是记不住位置、弹错、手型错误。

图表 1 智能尤克里里

于是我们认为:学习一门乐器本来就不是一件很简单的事,它需要用户专注地投入很多时间，需要不断地练习。

所以我们从另一个角度切入：从辅助和纠错的功能来帮助用户学习，作为一个教练，而不是帮助用户简化这个过程。

而我们发现市场上现有的音乐软件也有一定的局限性: 和弦识别不够强大,缺少对初学者发生错误时指出错误类型的功能

于是就这个思路，我们开展了本项目。

1.2 项目意义

对于乐器自学困难这一普遍问题的原因,我们认为很大可能是初学者没有掌握正确的练习方法。并且，即使注意了练习要点，错误没有得到及时纠正，也会让初学者走很多弯路。因此，在乐器的学习过程中教练显得尤为重要。由于乐器教学需要付出更多的精力和时间，加上教育资源有限，乐器课普遍收费昂贵，而且有指导下的训练仅限于课堂，线下个人的练习纠错困难。

但对于整个行业而言，优质的教师资源稀缺，随着行业竞争的加剧，老师的成本也变得越来

越高，这使得普通家挺再花钱请陪练教师更不现实。而相比于传统乐器陪练，AI+乐器陪练的价格优势十分明显，可以随时随地上课，大大提供了便利性，并且，用户的学习数据可以沉淀到教学系统中，通过自适应学习系统，捕捉和回应学生不同的需求和反馈，一定程度上可以实现个性化的教学。因此，乐器的智能教学更加有机会打破现有教育资源供应的瓶颈，并能以普惠的价格提供给更多家庭。

二、项目研究主要内容

2.1.乐器的音频特性分析与识别

音频特性分析是音高识别以及后续其他研究内容的基础。

它包括以下内容：

1)乐器的单音特性分析

2)乐器的连续弹奏特性分析(即长乐句的特性分析)

3)和弦的音频特性分析。

1)其中单音特性分析与识别即为对乐器的每一个基本音(每个基本音只有单一的频率)的采样分析，根据常识，它应该是一段时间内呈现一定周期的声音信号与一些噪声的叠加。即时域存在一定周期性，频域存在频谱峰。我们拟通过这些特点或者更多观察发现完成乐器的单音识别。

2)乐器的连续弹奏特性分析即为对乐器连续弹奏的声音进行采样分析，由于时域上声音的叠加干扰，在音符相接处信号更加复杂。与此同时，在识别上我们想借助单音识别的基础，这就要求我们学习更多的音频处理知识以完成连续音到单音的切割、从一个音调中滤除其他音的干扰。

3)和弦的音频特性分析与识别则更为复杂，因为它是同一个时间段内多个频率的混合叠加，我们的主要思路是预先进行粒度较细的和弦组成分类，再利用机器学习中的分类模型进行识别预测。

2.2 乐器演奏错误的原因分析

在完成了音频特性分析与识别之后，我们通过对演奏者演奏的乐曲进行采样，由于错因包含有较多的杂音或是刺耳的声音，所以在一定的误差范围内，我们容易判断出演奏者的错音。

于是更进一步，我们想进行演奏错误原因的判断。这是一个错误分类问题，我们从

吉他入手，收集演奏错误对应的样本，比如按弦不准、打品等等问题，我们将进行样本的一些预处理，提取出音频特征点以供机器学习。在此之后，我们可以通过演奏者具体的错误类型给以具体的纠正与指导。

2.3.AI 谱曲创作

在以上三个任务均完成的情况下，我们将人工智能运用到作曲领域，创建一个人工智能音乐作曲模型。它经过大量曲谱库的学习，能够根据所提供的部分弦乐信息完成整个乐章的谱写，为程序的使用者提供灵感与参考，也能够帮助初学者加强对调性分析的理解，培养初学者兴趣，提升用户的使用体验。

三、 项目创新点与项目特色

3.1 自动纠错

在传统的乐器教学项目中，往往是教学视频或是弹唱方式的整合或者按弦的教学。这样的学习模式对于乐器自学者最大的问题在于错误无人指出，仅仅是以机械的方式帮助初学者记忆，如果自学时一开始的错误就无人指出，被不断强化记忆，这对乐器的学习是很不好的。而本项将机器学习与传统方法相结合，不仅能识别弹奏的音符、和弦是否正确，还能将错误进行分类，并给予使用者改善方法，这既是本项目的创新点也是我们项目的出发点。

3.2 对多种乐器适用

在完成一种乐器的算法完成之后，只要更改其他乐器的部分采样信息，就可以不断扩大适用的乐器种类。比如完成吉它的音频分析识别后，我们只需简单地采样钢琴每个音调的信息，便可以完成钢琴演奏的识别功能。

3.3 便捷化

不需要额外的辅助的硬件，随时随地便可使用微信小程序进行辅助练习。

3.4 AI 谱曲

通过 AI 谱曲模型，程序可以基于用户提供的一小段乐曲开头谱成整首曲子，不仅培养初学者兴趣，还能够为程序的使用者提供灵感与参考，提升用户的使用体验。

四、系统方案和技术路线

4.1 关键技术和设计思路

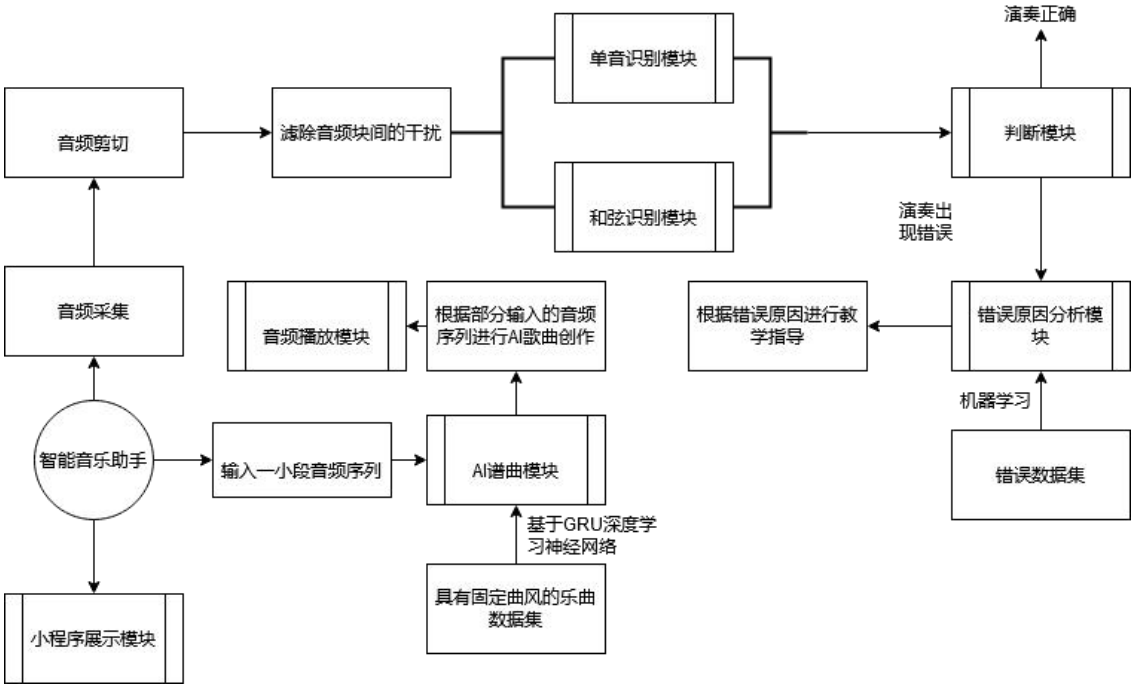
4.1.1 音频处理技术

借鉴传统的数字信号处理方式，利用 python 进行音频文件的分析处理以及特征提取。

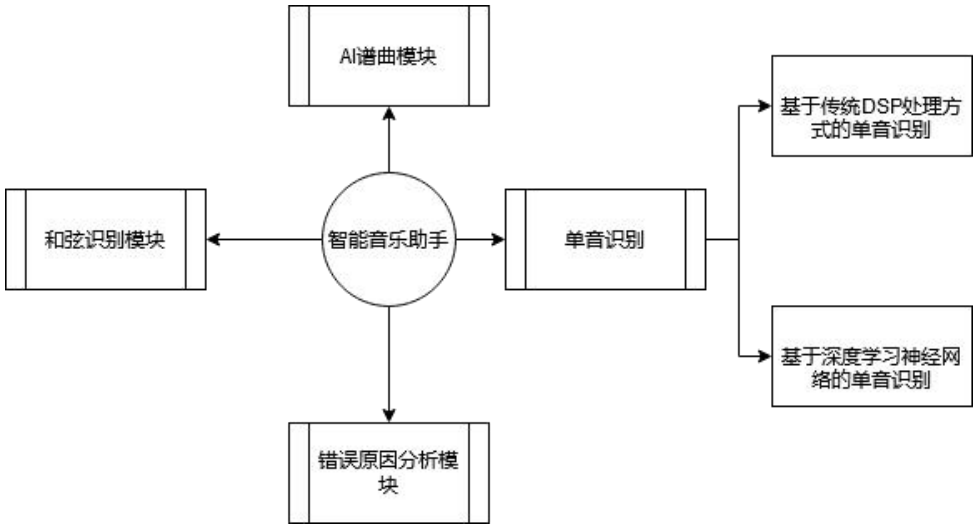
4.1.2 机器学习、深度学习神经网络

不同于传统音频处理的时域频域分析方式，借助提取出的音频样本特征进行学习使得处理结果更加准确

4.2 系统模块图和必要的说明



图表 2 总架构图



图表 3 主要功能示意图

4.3 功能概述（包括子模块的功能）

4.3.1 音频采集模块

将演奏的乐曲音频信号进行采集，保存为无损的.wav 格式。

4.3.2 音频剪切模块

根据音频信号特点，如幅值大小、瞬时能量大小完成一段音乐的分割，得到每个音符的分割时间点以及每个音符的时长节拍。

4.3.3 特征提取模块

对错误样本（各种弹奏错误的音频样本）进行音频特征点的提取(包括其时域序列、傅里叶变换后的频域序列、过零率等特征)，并进行矩阵转换、归一化等处理后作为机器学习模块的学习对象。

4.3.4 单音识别模块

首先会判断输入的音频是否属于单音，若结果为是，便通过训练好的分类模型返回其最有可能的音高(按吉他常用音高共 45 种音高)，若结果为否，则将该音继续放入和弦识别模块进行后续判断。

4.3.5 和弦识别模块

首先判断输入的音频属于和弦还是弹错的音(弹错的音因存在杂音会有较明显的特征)，若为和弦，则根据训练好的分类模型返回其最有可能的和弦组成，若为错音，则继续输入错因分析模块进行错因判断。

4.3.6 错误原因分析模块

将输入的音频提取出特征点，通过已经训练好的分类模型，可以完成对音频错误类型的分类(如按弦过紧、误触大品等等)，进而向用户反馈其错误类型、错误原因以及改善方法。

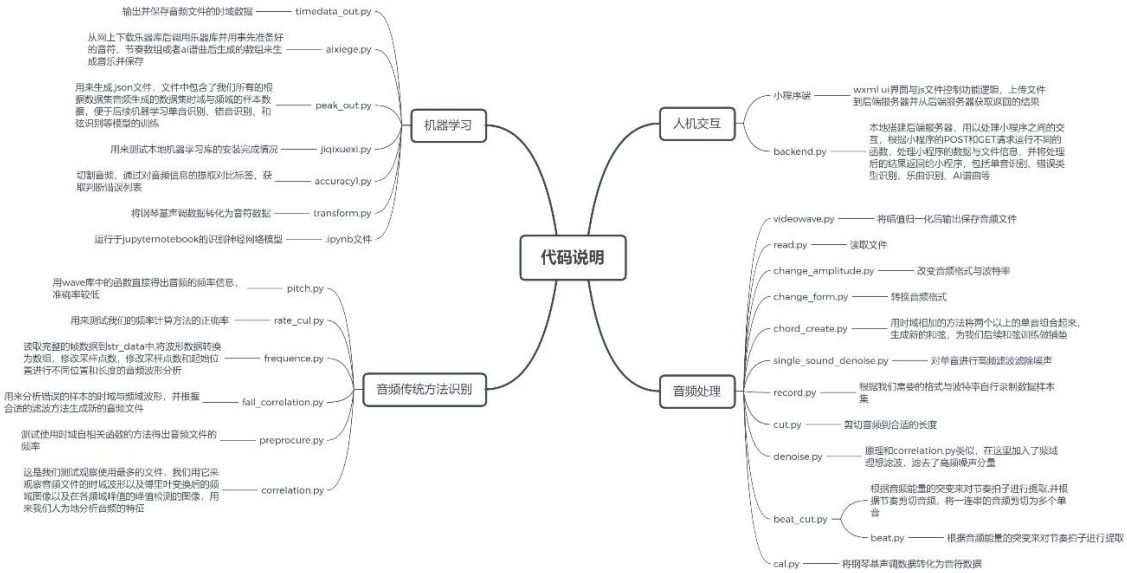
4.3.7AI 谱曲模块

根据用户输入的一小段曲子(即输入 20 个包含时长、音高信息的音符)配合已经训练好的谱曲模型完成整首曲子的创作，并调用乐器库实时演奏创作的曲子。

4.3.8 用户操作模块

以 flask 框架搭建服务器的后端，微信小程序作为前端，用户便可以通过便捷的微信小程序实现所有操作并得到即时反馈。

4.3.9 代码总框图



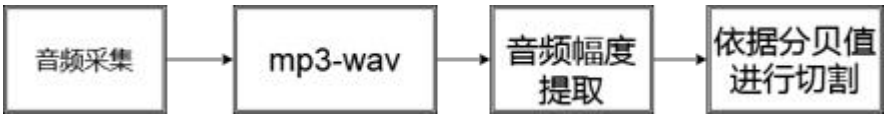
图表 4 代码框架

(在代码文件夹中有上图的图片文件以及 word 版本可以查看)

五、项目的实现

我们将分六个模块来说明项目的实现

5.1 音频切割模块

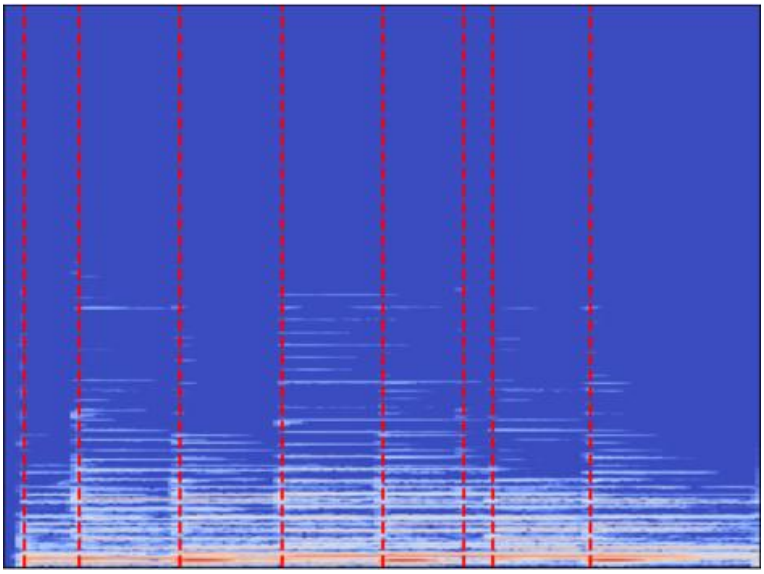


图表 5 音频切割模块示意图

音频切割模块作为输入的入口， 首先将进行音频采集，然后将.mp3 格式转换成.wav 格式，从有损的音乐格式转换为无损的格式，之后我们进行音频幅度(瞬时能量)的提取，再依据如下图所示按照强度分贝值进行切割。

在下图中横轴代表的是时间，纵轴代表的是频率，其中白色横线在某处越密集，表明在这一时间段内的该频率强度越大。

其中对音频分割的质量取决于阈值的设定,而在这一块我们也进行了参数的调节以使切割性能达到最好。而音频在切割之后，我们就可以将多音的长乐句都转换为单音进行识别了。

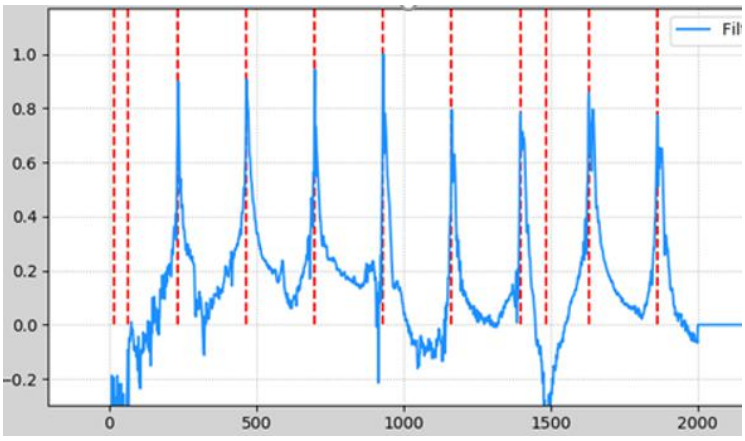


图表 6 音频切割示意图

5.2 单音识别模块

5.2.1 频谱特征峰提取

我们将音频序列经过傅里叶变换后得到它的频谱，如下所示为一个单音的频谱，其含有多个高峰，分别位于基音频率和高次谐波频点(高次谐波为其基音的整数倍)，我们根据频谱在这些点处的跳变性将这些频点都提取出来，如下图所示，红线即为被提取的频点。我们将这种具有一定幅度的跳变性对应的频率称作峰。



图表 7 特征峰提取示意图

5.2.2 基于传统 DSP 方法 在频域判断频率

根据所提取的频点，若提取的频点中准确包含基音频点以及每个高次谐波频点，显然只需要将每个频点间隔做差之后取平均就可以得到基波频率。但实际上这么做正确率仅能达到 80%的正确率，原因如下：

1.频谱不一定能准确囊括每个高次谐波频点，而采取相邻频点作差会因频点缺失造成较

大的误差，如频率若为 80Hz，谐波频点在 160，240，400，480（中间的四次谐波 320Hz 频点缺失） 可见仅一个频点的缺失就造成了 240-420 高达 160 的误差，而在较高频率部分高次谐波频点由于峰的跳变不明显而造成的缺失是比较普遍的，往往会接连缺失多个频点，尽管我们尝试了优化峰的提取,但始终会存在识别的遗漏。

2.频谱虽然包含谐波的高次频点，但是提出的峰值与实际频点存在着偏移误差。

这里需要说明的是，我们没有采用直接判断最小的频率峰，这是因为理论上基音的频点的确是最低的，但是在不同样本中峰值却忽高忽小，音频的基音所对应的峰并不一定被检测出来。再者，如上图所示，在基音的频点前也有可能被检测出峰，直接取最小值误差非常之大。而样本普遍的规律是，乐器演奏出的声音，如果是单音，则会同时包含基音频率和多次谐波频率,因此我们在最开始尝试峰值提取方法时采取了如上的频点提取，作差平均的方法。

5.2.3 基于传统 DSP 的方法 在时域进行判断

考虑到在频域提取的性能欠佳,我们又尝试了在时域进行识别参考。从学习过的通信原理中，统计学上，自相关被定义为两个随机过程中不同时刻的数值之间的皮尔森相关(Pearson correlation).因此，我们可以计算音频的时域自相关函数：

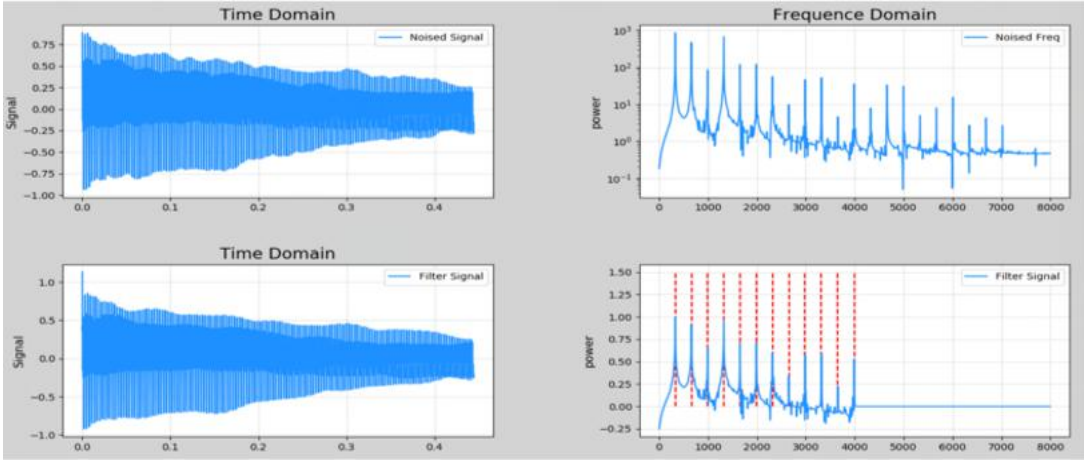
$$R(s,t) = \frac{E[(X_s - \mu_s)(X_t - \mu_t)]}{\sigma_s \sigma_t}$$

我们对其所一些变换，取为如下形式

$$R_f(\tau) = f(\tau) * f^*(-\tau) = \int_{-\infty}^{\infty} f(t+\tau) f^*(t)\, dt = \int_{-\infty}^{\infty} f(t) f^*(t-\tau)\, dt\,,$$

学习了通信原理后不难理解：它表示的含义是信号在经过一段时移后与自己的内积。当信号具有一定周期性的时候，时移接近周期的时候，其表达式近似是信号的平方积分，具有最大值。根据这个性质原理，我们将输入的音频序列时移积分，由于采集的音频信号为离散序列,所以我们只需要相乘求和即可,然后判断其取**最大值**时的**移动时间长度**来判定音频的周期，进而判断单音频率。（$f = \frac{1}{T}$）,而该方法在 2407 个单音样本中的准确率可以达到 95%。

下图为时域与频域识别的波形示意图,其中左上和右上分别为未经处理的时域波形与频域波形,左下和右下分别为经过降噪滤波后的时域波形和频域波形。



图表 8 时域频域识别波形示意图

5.2.4 基于机器学习深度神经网络的方法

5.2.4.1 尝试机器学习原因:

尽管我们使用传统方法已经有了 95%的正确率,但仍然没有达到我们的预想要求,考虑到单音识别只是音频识别的上层入口,如果在一开始便有了百分之五的错误识别率,那么后续的乐句识别,和弦识别,以及错误分类也将会受到很大的影响。并且由于传统方法的计算量较大,在音频的采样率上万 Hz 的情况下,尽管我们只需要截取一个周期长度的音进行识别,其计算时间也在 1s 左右,未能达到我们的预期要求。

5.2.4.2 数据集选择:

于是我们在 keras 平台搭建并测试了以下模型，其原始数据集均来自于 https://magenta.tensorflow.org/datasets/nsynth，(数据集已下发给助教，并附有数据集的详细说明文件)

5.2.4.3 尝试模型一：LSTM 时域 3D 张量模型

训练数据:3D 张量格式,且对时域长度为 0.1s 的音频序列不做任何变换直接输入

其示例格式为:[样本数量，时长，幅度]

训练标签：单音样本对应的频率

深度学习网络结构:LSTM+Dense+回归模型

训练后在测试集上的结果：在 2407 个样本中正确率 **0.149**,并不理想

5.2.4.4 尝试模型二:LSTM 频域 2D 张量模型

训练数据:2D 张量格式,示例格式:[样本数，[特征峰，幅度]],其中特征峰为预先进行特征工程,提取出峰值对应的频率作为输入

训练标签：单音样本对应的频率

深度学习网络结构:LSTM+Dense+回归模型

训练后在测试集上的结果：在 2407 个样本中正确率 **0.574**,未达到预期

5.2.4.5 尝试模型三: LSTM 频域 3D 张量模型

训练数据 3D 张量格式 [样本数，频率，幅度],与模型二的 2D 张量模型不同的是,此处将频域序列不做特征提取,只做归一化,向量化等变换后直接输入网络

训练标签：音高所对应的频率

深度学习网络结构:LSTM+Dense+回归模型

训练后在测试集上的结果：在 2407 个样本中正确率 **0.514**,仍未达到预期

尝试了以上的几个模型后,效果始终欠佳,我们思考:可能对于机器学习的回归模型不断优化的是预测出的频率，而并非我们最终输出的音高，所以可能直接使用分类模型会更为简单。另外，采用 LSTM 网络的时候网络过于复杂，参数过多，训练的效率较低。因此我们尝试了以下模型，不再拘泥于计算频率，而是将特定的音高分类：

5.2.4.6 尝试模型四:CNN 网络多分类模型

训练数据：傅里叶变换后的[频率，幅值]

样本标签：对应的音高类别

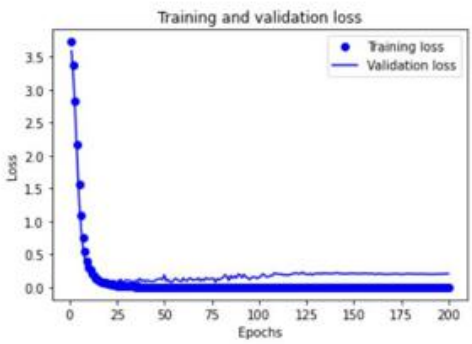
深度学习网络结构：CNN 网络+多分类模型

训练后在测试集上的结果：在 2407 个样本中正确率 **0.94**

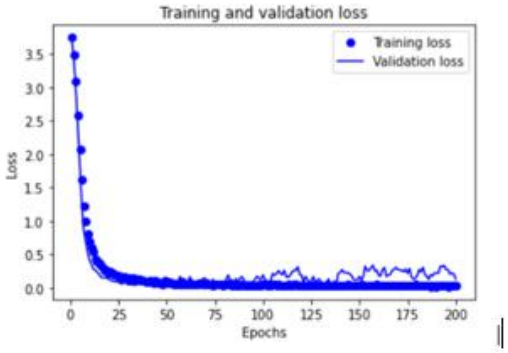
5.2.4.7 神经网络模型的改进

我们对模型四的训练以及验证曲线进行观察，发现神经网络在训练过程中，随着训练轮数的增多，虽然训练集的误差在不断下降，但实际上验证集的误差反而会升高，即模型虽然在训练中贴近训练集，却最终没有更好地贴近未知的数据集，出现了过拟合现象。为了解决过拟合，我们采用了增加 dropout 的方法。其原理是在训练过程中随机断开一些全连接层间的连接，从而使得最终结果更具有普适性。

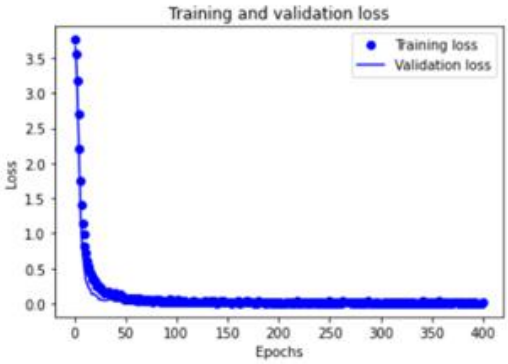
我们通过调整 dropout 的参数不断得到训练曲线进行分析比较，最终确定取 dropout 为 0.13 时能抑制过拟合现象。下图是 dropout 参数分别为 0，0.1，0.13 的曲线，其中 dropout 参数取值越大，全连接层间断开连接的几率就越大，普适性就会越好，但同时造成网络缺失的风险也会越大。



dropout=0 的训练以及验证曲线



dropout=0.1 的训练以及验证曲线



dropout=0.13 的训练以及验证曲线

图表 9 dropout 不同取值时的训练以及验证曲线

从训练误差和验证误差的贴合情况来看，dropout=0.13 时，模型在样本集和训练集上的表现更为贴合，也是我们单音识别最后采用的模型，结果为**在 2407 个样本中：准确率由 94%上升至 97.2%**

5.2.4.7 传统方法与 AI 结合：

在达到了 97.2%的识别率后我们还是有所不甘，在考虑到传统方法也有不低的识别正确率后，我们思考能否将传统的时域与频域计算的方法与之结合。因为我们观察到识别出错的部分往往集中于频域的峰值计算和时域周期自相关计算出的值都不对应音频(大约占总犯错音频的 95%以上)，即两种传统方法的计算值有出入时有很大概率会犯错，于是我们思考能否让机器学习专门处理这些集中了几乎所有错误的样本呢？但在将这些样本作为训练数据单独训练后，我们发现也并未有性能上的大幅提升。

另一方面，我们在去调查了每种方法都会识别出错的样本后发现，这些样本有一部分本身的质量就很差，考虑可能是原数据集也存在一些错误集的原因。最终，我们还是采取了改进的 AI 模型四作为我们的单音识别模块模型。

5.3 和弦识别模块

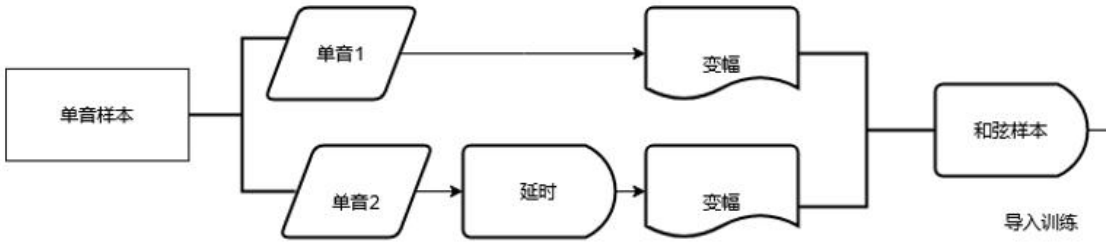
5.3.1 和弦的弹奏原理

以吉他为例，一个和弦音调包含多个基音，它是通过演奏者同时弹奏多根弦得到的。例如，Am-Em 和弦的演奏，Am 和弦使用左手二指、三指分别按四弦、三弦二品，而 Em 和弦正好是是用左手二指、三指分别按五弦、四弦二品。可见，要识别一个和弦，可以通过识别一个和弦的所有单音。但需要说明的是，上述例子中只是最简单的和弦，只需要同时按压两根弦，实际上可能涉及更多弦。

5.3.2 和弦样本的获取

目前网络上并没有大量带和弦标签的样本以供使用，于是根据和弦的弹奏原理，我们采取了自己生成样本的方法：

我们先从单音样本中提取出两个单音，记为单音 1 和单音 2，然后将其中一个进行延时和幅度的成比例变换，另外一个只进行幅度的比例变换。延时模拟的是实际演奏者按弦时候的时间差，幅度变换模拟的是拨弦力度的变化。我们按照这个方法，生成了约 12000 个和弦样本。



图表 10 和弦样本的获取示意图

5.3.3 神经网络的组成

我们通过判定组成和弦的所有单音来确定和弦，因此网络模型与单音识别具有一定相似性。因此在探索了单音识别的识别模型基础上，我们采取 CNN 网络+多分类模型来识别和弦。我们将约 11000 个样本划分出约 1200 个训练样本，将训练样本的傅里叶变换后的[频率，幅值]作为输入，将对应的和弦类别作为输出。最终在 10315 个样本中模型表现出的识别准确率为 90%。

5.4 错误类型识别模块

由于错误类型的如果只从音频进行区分难度很大，只有经验丰富的音乐老师才能仅凭耳朵分辨出学生弹错的原因。于是我们考虑利用机器学习训练出分类器，利用分类器对其错误进行分类。

首先，我们需要获得训练样本，由于在网络上并没有办法找到相关的数据集，所以我们只能通过录制以错误手型弹奏出的音频并加上标签的方法来获取样本，但因为工作量很

大，我们最终只获取了 200 个样本。我们总结了以下几类错误，并给出了大致的音频特点:

1.位置弹错:此类错误有别于其他错误类型，单纯是与曲谱对应的音高无法对应，即音高位置弹错。

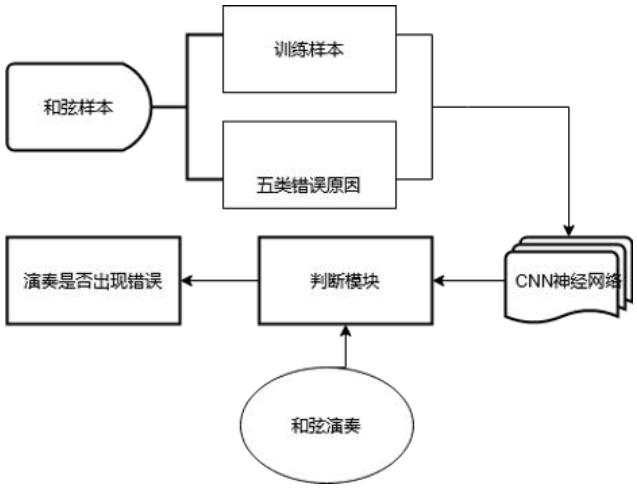
2.没按紧弦

3.琴弦打品

4.右手拨弦后触碰琴弦

5.左手按到了品丝上

将上述获取的 200 个样本输入 CNN 神经网络训练之后得到了我们的模型，最终的分类性能是将测试集分类为 5 类错误，正确率为 **55%**，尽管我们对网络的参数调节进行了很多尝试，但可能受限于样本数量太少，正确率始终不算太高。在模型识别出错误之后，小程序也会及时提醒，并反馈给使用者改善方式。



图表 11 错误类型分类流程图

在得到了分类结果后，我们也尝试从机器学习的结果反推出其分类原因，最后总结出各类型的音频特点如下:

1.位置弹错:其音质清晰。

2.没按紧弦:声音较小，声音沉闷，声音持续时间短。

3.琴弦打品:有突兀的爆炸的噪声，声音刺耳。

4.右手拨弦后触碰琴弦:在弹奏前有摩擦的刺耳的声音，音符纯净度低。

5.左手按到了品丝上:主声音同样较闷，但会有刺耳噪声。

图表 12 小程序反馈示意图

5.5 AI 谱曲模块

5.5.1 灵感来源

目前 AI 在自动写曲的应用还在起点，我们的大部分灵感以及参考都来自于 AI 在文本写作中，参考论文 *2017-11-11 Research on Automatic Writing of Football News based on Deep Learning*

其中讲述了深度学习网络用语自动生成足球新闻的报道。我们还参考了 AI 写小说、编对话的方法。它们的原理都具有相似点，那就是将文本化为一个个基本单位，这个单位一般是一个单词或者一个词条，然后将这些词条映射到一个多维空间，简称词嵌入。之后，将由一定个基本单位组成的句子作为训练样本输入，将下一个词/词条/句作为训练输出标签。

5.5.2 我们需要什么样的样本？

首先我们需要将一首曲子分为基本单元。一首曲子由一个个音符以及其对应的节拍组成，于是我们将曲子转为[音符类别，时长]的二维序列。休止符也同样被我们归为一种值为 0 的音符类别。

5.5.3 样本的获取

由于目前网络上并没有我们所要求的[音符类别，时长]这样的二维序列文件，于是我们通过人工手动转换五线谱的方式录入了周杰伦的 10 首曲子，转换为了所需格式。这项工作不仅需要有一定的识谱能力，而且每一首曲子的转换工作量都非常大，如果通过完全人工转换得到足够的训练样本是几乎不可能也没有意义的。

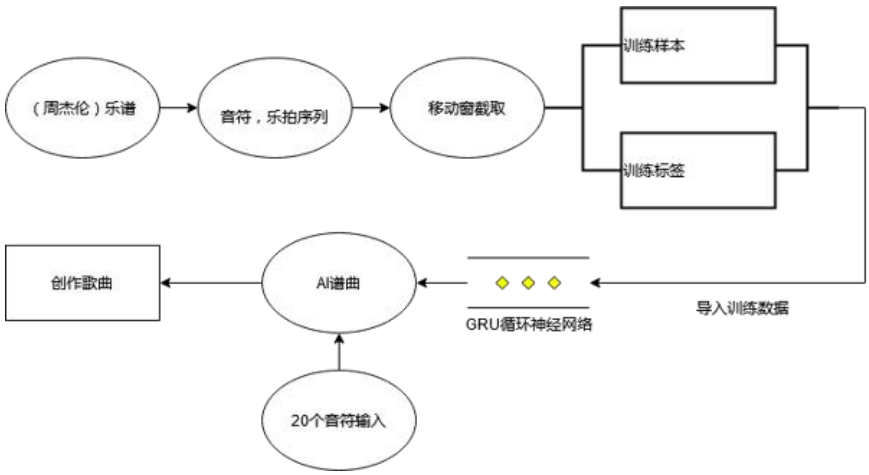
因此我们采取了下列的方式进一步获取样本：假设《青花瓷》乐曲长度为 112，即有 112 个[音符类别，时长]单位。我们将 1-20 个提取出来作为训练输入数据，第 21 个作为输出标签，然后依次往后获取（2-21 作为输入，22 个作为输出标签），就好像用窗来截取这个序列一样。最后一共得到了 1021 个训练数据，这些样本中共包含 41 种音符，每个音符都至少包含一个，但是数量不均，时长包含 12 种，数量同样不均。

我们的训练数据全都是从周杰伦的曲子中提取出来的，因为我们未来进一步是希望可以训练出专门的曲风，比如可以做摇滚、嘻哈、古典等不同曲乐的生成，所以我们先处理的是具有相类似音乐风格的音频。

5.5.4 神经网络组成

预测音符部分：GRU 循环神经网络+CNN+41 分类模型

预测时长(节拍)部分：GRU 循环神经网络+CNN+12 分类模型



图表 13 AI 谱曲流程图

5.5.5 谱曲过程

用户需要输入一段长度为 20 的音频序列(即上述要求的[音高，时长])，我们将分别预测第 21 个单位的音符以及时长，并将其合并至原音频上，再使用范围为 2-21 的音频继续作为输入，来预测第 22 个，以此类推，完成整首曲子的创作。在完成创作后，我们将输出乐谱并进行演奏，其中演奏采用了 Python 中的 mingus,fluidsynth 等音乐库和名为 GeneralUserSoftSynth 乐器库文件以及我们自己编写的格式转换文件。

5.6 人机交互模块

在得到上述各个模块训练出的模型后，我们将各个模型储存，用 Flask 框架搭建服务器的后端，用便捷的微信小程序作为前端，就可以打通端到端的连接。

其中微信小程序的各个功能按钮绑定对应的后端页面网址，而在 Flask 后端代码中也设置对应的路由调用所有已经整理好的代码、模型，并返回给前端。

图表 14 微信小程序示意图

5.7 课题成果

5.7.1 最终成果以及达到的技术指标

1. 单音识别方面，要达到 97.4%以上的准确率(指音高的识别准确率，以半音为单位)
2. 和弦识别达到 90%以上的准确率
3. 整段乐谱识别要达到 95%以上的准确率（指音高的识别准确率，以半音为单位）
4. 经过机器学习后的纠错模块能够判断错误原因，并达到 55%以上的准确率
5. 整个音频分析流程时延不大于 2s（假设音频时长为 5min）

5.7.2 已具备的研究条件，尚缺少的研究条件及解决方法

1. 缺少更精准的分割算法

2.缺少错误类型分类中供机器学习中的错误样本

3.缺少更多供 AI 谱曲模型学习的曲谱样本

5.8 项目未来展望

1. 我们想要实现更复杂的和弦识别，比如包含更多组合、更多个音的和弦识别，让吉他、钢琴这些设计多和弦的乐器也能够准确识别。
2. 我们还想要得到更好的分类性能，由于我们目前只有两百个样本，识别准确率也有待提高，我们期望之后用更多的错误类型样本，来提高错误类型分类的性能。
3. 我们还希望改善 AI 辅助创作这一环节，可以增加更多限制条件比如固定曲风，或者固定调性地生成乐曲，可以帮助用户进行乐理的学习或者调性的分析等等。

六、人员分工和项目推进时间表

6.1 组内人员分工情况

冯韵菱:音频剪切模块，音频的时域频域分析，判断模块，机器学习模块，人机交互模块，贡献率：33.3%

张靖鸿:音频的时域频域分析，音频特征点提取，错误原因分析模块，人机交互模块，贡献率：33.3%

许宏涛:单音识别模块，音频错误样本采集，机器学习模块，AI 谱曲模块，贡献率：33.3%

注:由于项目在进行中我们遇到了很多困难，几乎每部分所有成员都有参与，但可能各自侧重有所不同，所以上述分工可能并非绝对。

6.2 项目推进时间表

图表 15 项目推进时间表

6.3 组间组内协作及分享说明

6.3.1 组间协作

由于我们组的研究课题与其他组差别较大，很难交流成果。但在最后搭建 flask 后端时，我们与 12 组人群密度检测仪的同学请教了后端搭建的经验，很感谢他们在项目最后给我们提供的帮助！

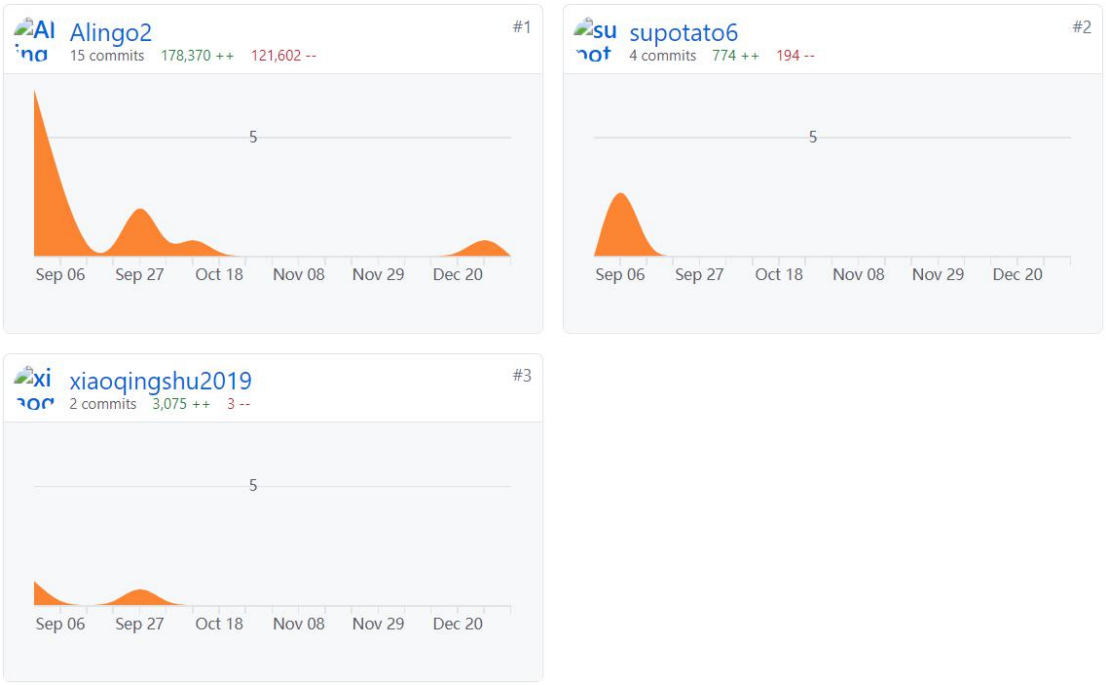
6.3.2 组内协作

虽然我们的组员横跨 28，29 两班，但我们沟通还算方便，以面对面交流为主，而我们之间的文件也通过 Github 实现共享和同步。

我们积极探讨了数字信号处理 DSP 的相关知识该如何应用到项目中来，找到了时域频域的两种识别方法。起初我们都是深度学习小白，一起学习《python 与深度学习》，理清了张量等重要概念，探讨了深度学习网络模型的搭建以及网络调整方式，也为我们的项目也打下

了基础。

代码更新情况如下图，在中后期的时候我们直接通过 qq 传输文件，因此没有在图中显示，github 上更新的代码只是冰山一角，就有 20 版之多。全部的代码已经在课题成果中附上链接。



图表 16 代码更新情况

七. 课题所用器材列表及说明

由于本项目主要以程序为主体，仅需要使用采集音频用的手机电脑以及演奏的乐器(吉他)。

八、课题成果链接

演示视频地址：https://www.bilibili.com/video/BV18i4y1w7oS

项目代码地址：https://github.com/Alingo2/AiMusicCoach

九、参加本课程的收获、体会及对课程的建议

9.1 收获

做完整个项目，我们总结了我们的收获

首先是对 AI 应用场景的思考。AI 不应该沦为噱头，应用 AI 也不是为了炫技，而是由于传统方法存在着缺陷才需要 AI 来解决问题。我们在做这个项目的时候，第一个功能就是要实现单音识别，这本是一个 DSP 数字信号处理问题，但我们在应用传统方法的时候遇到了问题，这才采用 AI。我们的目的是为了解决传统方法不能解决的问题，或者是提高准确率，或者是减小算法的开销。这一思路也始终贯彻着我们的这一项目，

我们在需要解决某一问题的时候，都先尝试传统方法能不能解决，解决的效果好不好，不好在什么地方。然后再进行 AI 机器学习，希望能够得到提升。我们在进行音频切割的时候，我们基于音频分贝值跳变的传统方法表现的极好，已经做到将一首歌准确地分割为一个个音，这时候再采用 AI 就显得没有必要。我们在进行单音识别的时候，传统算法的表现较好，能够达到 94%，但尚未达到理论最优解，所以我们试图用 AI 的方法想要超越其性能。在进行错误分类甚至 AI 谱曲的时候，我们根本没有想出基于数字信号处理的算法，所以想用 AI 方法进行低复杂度实现。从这个角度看，AI 是一种新的方法论。

其次是我们对神经网络调整经验的学习。起初用神经网络做单音识别的时候，缺少经验的我们一上来就采用了最复杂的 LSTM 循环神经网络模型，但其表现不佳对我们打击很大。我们在测试了几个模型之后，开始从问题本身来分析我们需要什么样的神经网络，我们觉得这是一开始设计网络非常重要的一个起点，就像是做图像识别会率先采用 CNN，因为它善于抓取图片的局部特征，比如判断一辆车，最后的结果可能是第一层判断有没有四个轮子，第二层判断车的框架是不是长方形，等等。我们后来也思考可能对于机器学习的回归模型不断优化的是预测出频率，而不是音高，所以可能直接按分类模型会更为简单。另外，采用 LSTM 网络的时候网络过于复杂，参数过多，训练的效率较低。因此我们尝试了不再拘泥于计算频率，而是将特定的音高分类的分类 CNN 模型，最终取得了很好的效果。在这个基础上，我们还用 dropout 克服过拟合优化了网络。总结出在搭建网络中的几点，一是学会了通过测试来设计网络，即不断尝试观察结果，二是学会了分析问题的特点来设计网络，去思考解决的问题是什么样的，它需要什么样的网络能帮助我们少走弯路，三是学会了根据训练曲线调整网络，比如我们观察单音识别的训练集和验证集曲线观察到过拟合现象，所以采用 dropout 方法。同时，我们在神经网络的搭建中也有很多很多的收获，尽管过程非常艰难，我们小组的成员都从抱着《Python 深度学习》这本书，从第一页开始学起，但我们想，也正是因为我们这份认真，才有了扎实的基础能解决后面数不尽的问题，不管是后面的从 0 开始搭建网络、优化模型，甚至是自己尝试编写损失函数，可能很多过程并没有直接的回报，但我们认为整个过程是宝贵的，我们也通过这些经历学到了很多。

最后的收获是来自我们对项目应用本身的思考。我们小组在进程中也发现整个项目都非常具有创新性，和弦分类、错误分类、AI 谱曲在网上都找不到参考，我们几乎完全是凭着兴趣硬把这个项目吃下来。以 AI 谱曲为例，对于很多公司来说它都处于一种

起步阶段。在我们做答辩的那一天，也就是 2020 年 12 月 13 日，网易在《2020 网易未来大会》上公布了第一首完全由人工智能技术生成的歌曲《醒来》，展现了 AI 赋能音乐创作，带来大批量生产商业化音乐的魅力。这也引发了我们对 AI 作曲这一功能的更深入思考，作曲这一功能实际很简单，随机生成音符也可以叫做作曲，但这样的曲子显然毫无意义。我们的 AI 谱曲模型在训练过程中究竟有没有学会乐理，我们采用的 GRU 循环神经网络有没有在记忆之中，跳出单纯地以出现概率来预测下一音符的约束，真正达到智能作曲，是我们未来所应关注的。然而留给我们的时间很少，我们在三周之内搭建了网络，用周杰伦的一些歌曲训练了网络模型，并用其生成了我们的歌曲，而去弄清网络内部的机理，弄清它是否学会了乐理可能需要数几十倍的时间，但我们觉得这是未来很好的一个研究方向。

我们希望在未来可以为 AI 谱曲创造具体化的落地场景，也将不断完善这一功能。比如目前它只是谱出一个乐器一个声道的谱子，这使得我们的结果听起来其实是非常单调的，我们可以不断扩充直接让其奏出架子谷，吉他，钢琴，大提琴等等的乐谱，直接奏出一场音乐会。我们还可以为创作的音乐设定风格，比如传统、摇滚、嘻哈等等，以此来减少不确定性增强应用性。这些都是这个项目非常好的未来延展，也是我们一直以来的动力之一。

9.2 建议

最后是给课程的一些真诚的建议：

非常感谢创新实验课的各位老师和助教对我们组一直以来的指点与帮助，我们组在这门课中感受到了非常好的鼓励创新的氛围，也因此能够将自己的想法加以实践转为小成果。不过对于本次 AI 主题的创新，没有找到弄清网络内部结构的方法，对最终结果做出解释，还是有些遗憾。所以我们也真心希望在以后的课程中能否邀请一些从事 AI 领域的老师来给同学们指导，提高大家的视野与能力，让我们能够在理论知识和方法论的综合指导下进行有效的创新。谢谢！

十. 参考文献

[1] *《Python 深度学习》François Chollet*

[2] *《通信原理》周炯槃 第 4 版*

[3] *2017-11-11 Research on Automatic Writing of Football News based on Deep Learning* （AI 谱曲灵感来源）

[4] *https://www.cnblogs.com/YuanZiming/p/13070766.html#%E4%B8%8B%E8%BD%BD%E5*

%B9%B6%E9%85%8D%E7%BD%AEfluidsynth （Python 演奏音乐）

[5] https://blog.csdn.net/qazwsxza/article/details/102655670 （librosa 库提取音频特征）

[6] https://www.jiqizhixin.com/articles/2019-01-11-25（python 音频信号处理）

[7] https://www.cnblogs.com/meelo/p/7839453.html（python 音频切割）