

从左到右顺序是：

采样旋律，调性，和弦，权重特征，权重音，结构和弦，终止和弦

后面是4个特征

结构和弦，权重特征是c++代码

终止和弦，权重音是python代码

原始数据集

[0, 0, 0, 0]|C.MAJOR|[]|[0]|[0, 0, 0, 0]|[31, 33, 38]|[0]

[0, 0, 0, 0]|C.MAJOR|[]|[0]|[0, 0, 0, 0]|[31, 33, 38]|[0]

[0, 0, 0, 67]|D.MINOR|[]|[67, 71, 74]|[0, 0, 0, 0]|[31, 33, 38]|[0]

[67, 67, 67, 79]|C.MAJOR|[]|[67, 71, 74]|[0, 1, 1, 0]|[31, 33, 38]|[0]

[79, 79, 81, 79]|C.MAJOR|[48, 52, 55]|[67, 71, 74]|[0, 0, 0, 0]|[31, 33, 38]|[1]

[79, 79, 77, 76]|C.MAJOR|[48, 52, 55]|[76, 79, 83]|[0, 0, 0, 0]|[31, 33, 38]|[1]

[76, 76, 76, 72]|C.MAJOR|[48, 52, 55]|[72, 76, 79, 81]|[0, 1, 1, 0]|[31, 33, 38]|[0]

[72, 74, 76, 77]|C.MAJOR|[48, 52, 55]|[72, 76, 79, 81]|[0, 0, 0, 0]|[31, 33, 38]|[0]

[77, 77, 79, 77]|C.MAJOR|[50, 53, 57]|[72, 76, 79, 81]|[0, 0, 0, 0]|[31, 33, 38]|[0]

[77, 77, 76, 74]|C.MAJOR|[50, 53, 57]|[74, 77, 81]|[0, 0, 0, 0]|[50, 53, 57]|[0]

[74, 74, 74, 67]|C.MAJOR|[55, 59, 62]|[74, 77, 81]|[0, 1, 1, 0]|[50, 53, 57]|[1]

[67, 67, 67, 76]|C.MAJOR|[55, 59, 62]|[74, 77, 81]|[0, 1, 1, 0]|[50, 53, 57]|[1]

[76, 76, 77, 76]|C.MAJOR|[48, 52, 55]|[67, 71, 74]|[0, 0, 0, 0]|[50, 53, 57]|[1]

[76, 76, 74, 72]|C.MAJOR|[48, 52, 55]|[72, 76, 79]|[0, 0, 0, 0]|[50, 53, 57]|[0]

[72, 72, 72, 76]|C.MAJOR|[48, 52, 55]|[72, 76, 79]|[0, 1, 1, 0]|[50, 53, 57]|[0]

[76, 76, 76, 69]|C.MAJOR|[48, 52, 55]|[69, 72, 76]|[0, 1, 1, 0]|[50, 53, 57]|[0]

版本二

[((0, 0), 256)]|[(C.MAJOR, 32), (D.MINOR, 16), (C.MAJOR, 208)]|[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 67, 67, 67, 67, 79, 79, 79, 81, 79, 79, 79, 77, 76, 76, 76, 76, 72, 72, 74, 76, 77, 77, 77, 79, 77, 77, 77, 76, 74, 74, 74, 74, 67, 67, 67, 67, 76, 76, 76, 77, 76, 76, 76, 74, 72, 72, 72, 72, 76, 76, 76, 76, 69]|[[]，[]，[]，[]，[48, 52, 55]，[48, 52, 55]，后面省略10个和弦]|[96, 208]

每轮计算公式(一轮计算需要计算16条):

第一个中括号内容 ：目前直接写((0, 0), 256) 情感VA标签、持续时长

第二个中括号内容：(调性名1, 这轮计算中连续出现的次数\*16), (调性名2, 这轮计算中连续出现的次数\*16)... 因为每一行时长为16

第三个中括号内容：把这轮计算中出现的采样旋律数组去掉中括号后全部放在这个数组中。采样旋律，每个时长为4

第四个中括号内容：把和弦数组全部放到这个数组中，和弦数组不用去中括号直接放。每个和弦的时长为16。

地五个中括号内容：每轮计算中，连续出现的1结束位置的index值\*6，注意index值为1~16。每个数字代表的是乐句的分割点，也就是和弦终止式的结束点。

版本三

[((0, 0), 256)]|[(C.MAJOR, 32), (D.MINOR, 16), (C.MAJOR, 208)]|[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 67, 67, 67, 67, 79, 79, 79, 81, 79, 79, 79, 77, 76, 76, 76, 76, 72, 72, 74, 76, 77, 77, 77, 79, 77, 77, 77, 76, 74, 74, 74, 74, 67, 67, 67, 67, 76, 76, 76, 77, 76, 76, 76, 74, 72, 72, 72, 72, 76, 76, 76, 76, 69]|[(0, 44)，(67, 16)，(79, 12)，(81, 4)，(79, 12)，(77, 4)，(76, 16)，(72, 8)，(74, 4)，(76, 4)，(77, 12)，(79, 4)，(77, 12)，(76, 4)，(74, 16)，(67, 16)，(76, 12)，(77, 4)，(76, 12)，(74, 4)，(72, 16)，(76, 16)，(69, 4)]|[96, 208]

每轮计算公式(一轮计算需要计算16条):

第一个中括号内容 ：目前直接写((0, 0), 256) 情感VA标签、持续时长

第二个中括号内容：(调性名1, 这轮计算中连续出现的次数\*16), (调性名2, 这轮计算中连续出现的次数\*16)... 因为每一行时长为16

第三个中括号内容：把这轮计算中出现的旋律数组去掉中括号后全部放在这个数组中。采样旋律，每个时长为4

第四个中括号内容：(连续出现的旋律1, 出现的次数\*4), (连续出现的旋律2, 出现的次数\*4)...二维旋律，音高和时长

第五个中括号内容：每轮计算中，连续出现的1结束位置的index值\*6，注意index值为1~16。每个数字代表的是乐句的分割点，也就是和弦终止式的结束点。

————————

测试/生成时，旋律提前能知道，和声需要等生成后才知道。所以终止和弦、结构和弦的代码肯定要保留在项目里，测试的时候用。

训练时，本身让模型去拟合这些数据，肯定要输入一对数据中的左边生成右边，所以旋律、和声都提前知道了。所以4个特征可以都提前处理好，以加快训练速度。

————————

对于接下来的歌曲改编，本质上也是这样的。

旋律->2维旋律，旋律->和声。（旋律+情感->2维旋律，旋律+情感->和声）

测试/生成时，只知道旋律，不知道还没生成的2维旋律 / 和声。所以输入包含是维旋律 / 和声的 情感乐理特征提取代码 需要保留在项目里，测试的时候用。

训练时，本身让模型去拟合这些数据，肯定要输入一对数据中的左边生成右边，所以2维旋律 / 和声等所有音乐内容都提前知道了。所以几个情感乐理特征可以都提前处理好、放在数据集中，以加快训练速度。

【但这些特征是留给下一个timestep用的！是对这个timestep生成出来的音乐内容，的特征提取！提取出来后这个timestep早就结束啦！】这三个文件中是不是这么放的我不确定，有没有往后做一个移位啊，这个需要问一下whq等会，检查检查

情感相当于原来的调性。这种属于输入，不是生成的东西，但是确实不算音乐内容。接下来我们中间就包括调性+情感 两个。

调性肯定会用在几个情感乐理特征的提取代码中；但情感则不会用到！只是保留着

一个纠正：

1. 情感乐理特征，应该是只基于音乐内容的！情感直接作为几维向量 跟我们最终的情感乐理特征拼！【正确】
2. 情感乐理特征的提取，用上这个情感，会不会更有效？但这个是本次timestep生成时输入给模型的情感，我们提出情感乐理特征的原因就是怕生成的效果不好、到达不了输入给模型的情感的要求。然后这个情感乐理特征是要能够直接通过深度学习变成某种情感的，代表着实际生成的音乐的真实的情感，传递给下一个timestep的生成。【所以是不能结合情感的！ 错误】