1.本周翻译并总结了一篇论文

SONG FROM PI: A MUSICALLY PLAUSIBLE NETWORK

FOR POP MUSIC GENERATION

2.自学了python

看完论文之后的个人总结如下

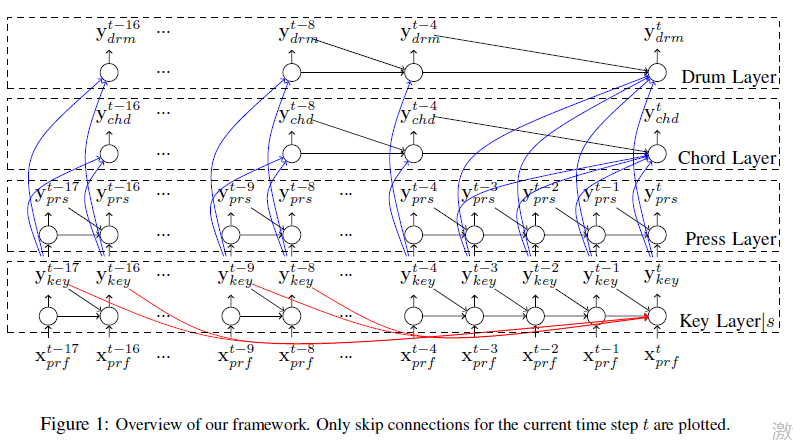
首先，本文是用rnn（循环神经网络）的框架来产生流行音乐的。模型框架是一个层次循环神经网络，层和曾结构编码流行音乐的知识。大致为底层产生训练，而较高层产生鼓和和弦。另外，rnn框架的两个应用，1.神经网络跳舞和卡拉OK 2神经网络讲故事。

早期八十年代采用循环神经网络的方法，通常编码旋律和和弦。因为它们每个时间段只产生一个音符，大多数都是单轨。

此文目标是制作多轨音乐。新颖性主要是将音乐理论的知识结合到构建神经体系结构中的分层模型，并产生多轨流行音乐（旋律，和弦和鼓）。

任何数字序列是π数字序列的子集，所以从完全随机的基本信号也可以创建令人愉快的音乐。 另一方面，作曲家使用特定规则，如将数字序列转换成音乐表。 这些规则在将随机性转化为音乐方面发挥关键作用。

由π的想法，提出分层方法。每个层都是循环神经网络（rnn）， 底层产生旋律，而较高层产生鼓和和弦，使得鼓和和弦层能够补偿旋律，以便产生令人愉快的音乐。 另外对模型在音阶上也有所调整，是旋律发生器学习用特定音阶的音符。



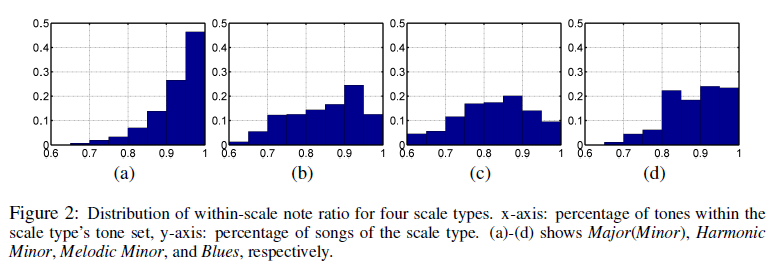
借用音乐理论：当从一个和弦转换到另一个和弦时，移动到临近的和弦通常是好听的，形成了和谐感强烈的和弦的过程。

本文把产生音乐放在了音阶类型上。旋律按音阶产生。把每个时间步长编码带有两个随机变量的旋律，表示正在播放的key（key层）和key被按下的持续时间（press层）。假设鼓和和弦是独立被给予的旋律，在旋律上进行调节，在每个时间步骤中，我们产生和弦（和弦层）以及鼓（鼓层）。 所有层的输出都会产生最后一首歌曲。最后组成一首有旋律，有鼓，有和弦的多轨歌曲。

通过对音乐发生器进行音阶调节，可以更容易地捕获这些规律。 但是并不强制从子集生成的音符，并允许我们的模型在音阶之外生成音符。

由于音阶相对于起始音符的定义，首先尝试分解出其影响力和规范所有歌曲具有相同的起始音符。为了识别歌曲音阶的音阶，计算12个音调的直方图，并与12个不同开始音符的4个比例类型的48个音调子集匹配。 然后，我们通过在所有音符上应用恒定移位来将所有歌曲都以C调为起始音调。

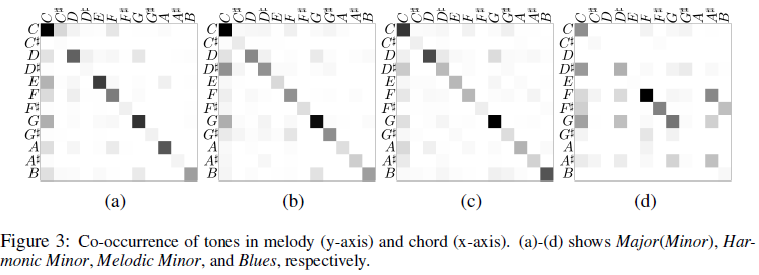
这使得任何歌曲可以分为4种音阶类型。即使这种转变会立即影响所有的音符，它不会影响歌曲的声音（它的和谐）图2，其中x轴表示一首歌的音阶的百分比，y轴表示数据集中有多少歌曲具有该百分比。注意大多数音符遵循音阶规则。此外，不同的音阶类型具有不同的内层分布。 因此，我们用一个单一的随机变量s属于{1,2,3,4}来表示音阶，S对于一首歌来说是固定的，并且在s上对模型进行调整。



用两层rnn产生旋律，用每个时间步长的两个随机变量表示旋律：按下哪个调以及按下的持续时间。 循环神经网络（RNN）是用于产生音阶的关键条件。 然后对音调层（key层）的输出进行调节，第二个RNN在每个时间步长产生按下的持续时间。在本文中，我们利用了LSTM，它们以最基本的形式（单层）通过Wf; Wi; Wo; WC可学习参数的，计算所给出输入xt的隐藏状态ht，。 这里f，i，o，C，e C和h表示忘记门，输入门，输出门，单元状态，输入单元状态和隐藏状态。具体来说，我们使用具有512维隐藏状态的双层LSTM对音调（key）层进行建模，每个时间步长都会输出一个音调（key）。注意，我们在音阶 s 上进行调整，因此我们每个音阶具有不同的参数。 我们只允许C3到C6之间的音符，因为这个范围之外的音符通常太低或太高，不能听起来很好。 我们提醒读者，给定一个音阶，十二个音符（每个八度）中有七个（或六个蓝调）在统计上更可信，但是我们允许该模型从所有12中选择。这导致了37维输出， 因为有36个可能的音符对应于3个八度音阶，每个八度音阶12个音符，加上原音。 让htkey成为时间t的第二个key解码器层的隐藏状态。 我们使用softmax计算每个音调的概率。

使用连接多个功能的向量作为LSTM的输入：先前生成的音符yt􀀀1key，回读功能和旋律曲线的单热编码。 它们包括两个和一个以前的跳过连接（一个bar是8个连续播放的音符），即yt􀀀16key和yt8key。 它们还包含两个附加功能，指示上一个生成的key是从一个或两个前一个bar复制的，即1（yt􀀀1key yt􀀀1􀀀8key）和1（yt􀀀1key yt􀀀1􀀀16key）。 他们还添加了一个5维特征表明当前时间t的二进制编码。 这有助于模型跟踪在4􀀀bar范围内，因此相应的产生音乐。

旋律配置文件的新功能：配置文件代表高层音乐流。 要获得每首歌曲的配置文件，我们计算每个宽度为2bar的每个时间步长的局部音符直方图，并通过k-means将歌曲内的所有局部直方图聚类成10个群集。 我们使用从1到10的从低到高排列的10个聚类，并对簇ID序列应用移动平均值，以使得局部平滑。 这导致每个时间步长的10维单向量向量表示簇ID。 该附加信息允许用户设置歌曲的旋律有所起伏。



单独的key不足以描述如何执行旋律，所以需要知道每个key需要按下的持续时间。用一个512层隐藏状态的双层LSTM生成每个key的持续时间。将按压的持续时间表示为基于产生的旋律的正向计数序列。当按下新的key时，按输出1，按住键顺序输出2,3,4等。当前key被释放时，计数器被重置为1.。我们将press yprs t表示为8维单热矢量。我们的LSTM的输入是yprs t-1，与旋律键ykey t的37维单热编码相连。

旋律和鼓RNN层：和弦可以是多个音符的任意组合，将和弦ychd t表示为具有72个类的单热编码，并使用具有512维隐藏状态的双层LSTM进行预测。 我们每个时间步长都会产生一个和弦。 输入是ychd t-4与ykey t-3连接：t。

学习：使用交叉熵作为我们的损失函数来训练每一层。 我们遵循典型的培训策略，我们在每个层次和时间步骤上进行预测，但是将地面信息提供给下一个。 这有效地分解训练，并允许并行训练所有层。 我们使用Adam优化器，学习率为2e-3，学习率衰减为0.99，每个时期为10个时期。

音乐合成：首先随机选择一个音阶和一个轮廓xprf。 为了生成xprf，我们随机选择一个具有随机持续时间的簇ID，并重复，直到得到所需音乐序列的总长度。 然后，我们在我们的模型中根据所选择的音阶进行推理，并使用xprf作为键层的输入。 在每个时间步，根据P（ytkey）对一个key进行抽样。 我们将其编码为一个单一的热矢量，并传递到press层，和弦和鼓层。 我们以类似的方式在每个时间步取样press，和弦和鼓。

在将输出层叠在一起之前，进一步调整在条形图级别生成的序列。 对于旋律，首先检查每个栏，如果第一步是上一个音符或静音的延续。 如果是后者，就在条形图中找到第一个新按下的音符，并将其移动到条的开头。

实验：

使歌曲有任意的速度：检测每首歌曲的两个相邻音符之间最频繁的间隔，并将该间隔迭代地除以2，直到它落在0.25s和0.5s之间的范围内。 我们使用它作为歌曲节拍持

模型产生新音乐的能力：使用不同的随机初始化生成了50个长度的100个序列。 然后，对于每个序列，我们搜索与训练数据的一部分匹配的最长的子序列，并记录其长度。 我们发现，使用1小时的训练数据，平均匹配子序列长度为3.46s。 有100小时的训练数据，平均长度增加到4.65s，因为有更多的可能的匹配。 序列非常小，因此，我们的模型能够产生新的音乐。

应用：

1.神经跳舞和唱歌

目的：产生音乐，stickman跳舞，卡拉OK文字。

我们使用Newell等人的方法（2016）在视频中跟踪单帧2D人物姿势。我们处理单帧跟踪结果，以确保时间的左右身体一致性，将2D姿态序列转换为3D。我们将姿态相似性定义为所有关节的平均欧几里德距离，并将群体姿势定义为456个群集。通过在key层顶部添加另一个舞蹈层来产生一个stickman跳舞，就像鼓和和弦一样。 我们在每个节拍中产生一个姿势，相当于4次步幅或0.5秒钟在每分钟120个拍的音乐。 特别是，我们预测456姿态之一使用线性投影层，随后是softmax。 在每个时间步长使用交叉熵作为我们的损失功能。 推测时间，我们进一步适用移动平均时间平滑生成3D姿势序列。

所用数据包含50小时无音乐的文字。对于音乐部分，我们通过使用广泛存在的lrc格式将歌词中的每个句子与MIDI音乐暂时对齐，该格式在每个句子的开头记录时间标签。 选择出现至少4次的单词，其中的词汇大小为3390，包括未知和句末。 就像舞蹈一样，我们使用key层顶部的另一个歌词层产生一个单词。

2. 神经讲故事唱歌

目的：唱一首关于照片的歌曲。

利用与时间对齐的歌词和音乐相同的1小时数据集。我们进一步包括我们3390词汇的音素名单，。从神经讲故事者产生的文本开始，我们将它排列成一个时间顺序，每个单词1个节拍和一个短暂的暂停，其中暂停长度被决定使得下一个句子从一个新的栏开始。由于我们的数据集相对较小，因此我们会生成符合文本条件的配置文件，与key相比尺寸较小。这是通过2层LSTM完成的，该LSTM将最后一个时间步长与当前单词的一个热向量连接起来，作为输入，并输出当前配置文件。然后，我们用生成的配置文件生成我们的模型的歌曲。然后，使用所产生的旋律键来确定虚拟歌手的音调频率，假设大型钢琴的琴键对应关系。我们进一步约束，歌手的最终音程总是在E3到G4的范围，我们经验发现是自然音高范围。然后我们用词语Ooh替换词汇外的所有单词，并用生成的音乐播放渲染的歌声。

Python的学习

小结：

1.Python的技术优势：面向对象、免费、可移植、功能强大、可以与其他编程语言混搭、易于学习和使用

2.Python唯一的不足之处是执行速度不如编译型语言（诸如C和C++）那样快

3.Python是一种动态类型的语言，对象和引用分离

4.Python的内存管理包括垃圾回收、分代回收和内存池管理

5.可以使用len、max、min和sum函数来获取列表的长度、以及所有元素的最大值、最小值和总和

6.random模块中的shuffle函数可以将列表中的元素进行随机排序

7.元组是固定的列表，不允许在元组中增加、删除和修改元素

8.由于元组是序列的一种，因此所有适用于序列的操作也可以用于元组

9.集合可以像列表一样用于存放大量的元素，与列表不同的是，集合中不允许有重复元素，且所有元素的位置是无序的

10.使用add方法可以向集合中增加元素，而remove方法则可以从集合中删除指定元素

11.len、max、min和sum函数均可以用于集合

12.需要遍历列表或集合中的所有元素时均可以考虑使用for循环

13.需要判断子集和超集的关系时，可以使用issubset和issuperset方法，此外还可以使用|、&、-和^来求解两个集合的并、交、差和对称差

14.涉及到测试集合或列表中是否包含特定元素，或者从集合或列表中移出指定元素时，集合的效率比列表的效率更高

学到的函数：

len(seq) 返回序列长度

list(seq) 从序列中构建列表

max(args) 返回序列或参数集中的最大值

min(args) 返回序列或参数集中的最小值

reversed(seq) 对序列进行反向迭代

sorted(seq) 返回已经排序的包含seq所有元素的列表

tuple(seq) 从序列中构建元组

set(seq) 从序列中构建集合

.