这一星期自学了python 看了一篇论文

1. Python

1.print语句可以用来打印由逗号隔开的多个值

2.可以使用import … as…语句进行函数的局部重命名

3.通过序列解包和链式赋值功能，多个变量赋值可以一次性完成，通过增量赋值则可以原地改变变量

4.块是通过缩排使语句成组的一种方法，可以在条件以及循环语句中使用

5.条件语句根据条件执行或者不执行一个语句块，几个条件可以串联使用if elif else

6.简单来说，断言就是肯定某事件为真，如果为假断言就

7.循环语句在条件为真时继续执行同一语句块，可以使用continue语句跳过块中某个语句直接进入下一次循环，也可以使用break语句跳出当前循环

8.通过列表推导式，可以从旧的列表中产生新的列表、对元素应用函数、过滤掉不需要的元素等等

9.pass语句什么都不做，可以充当占位符使用；del语句用来删除变量（即对象的引用），但是不能删除值；exec函数执行存储在字符串中的Python代码；eval函数用于计算以字符串形式书写的表达式，并返回计算结果值

10.对象：包括特性和方法，特性作为对象一部分的变量，方法则是存储在对象内部的函数；函数与方法的区别在于方法总是将对象作为自己的第一个参数

11.类：代表对象的集合

12.多态：对不同类型和类的对象进行同样对待，即无需知道对象属于哪个类就能调用方法

13.封装：对象可以将它们内部状态隐藏起来

14.继承：一个类可以是一个或多个类的子类，子类从超类继承所有方法

15.接口：对象的公开的方法和特性

16.递归：函数直接或间接调用自身，一切用递归实现的功能都可以用循环来实现，但是递归函数可读性更强

17.函数式编程：Python有一些支持函数式编程的机制，包括lambda表达式和map、filter及reduce函数

map(func, seq[, seq…])

对序列中每个元素应用函数

filter(func, seq)

返回函数值为真的元素的列表

reduce(func, seq[, initial])

对序列中的元素反复调用函数，等同于func(func(func(seq[0], seq[1]), seq[2],…)

eval(source[, globals[, locals]])

将字符串作为表达式计算并返回值

enumerate(seq)

产生用于迭代的（索引，值）

range([start,] stop[, step])

创建整数列表

reversed(seq)

产生seq中值的反向版本

sorted(seq,[, cmp][,key][,reverse])

返回seq中值排序后的列表

zip(seq1, seq2,…)

创造用于并行迭代的新序列

2.论文

Text-based LSTM networks for Automatic Music Composition

基于文本的自动音乐组合的LSTM网络

Abstract.

In this paper, we introduce new methods and discuss results of text-based LSTM (Long Short-Term Memory) networks for automatic music composition. The proposed network is designed to learn relation-ships within text documents that represent chord progressions and drum tracks in two case studies. In the experiments, word-RNNs (Recurrent Neural Networks) show good results for both cases, while character-based RNNs (char-RNNs) only succeed to learn chord progressions. The pro-posed system can be used for fully automatic composition or as semi-automatic systems that help humans to compose music by controlling a diversity parameter of the model.

Keywords: LSTM, RNN, automatic composition, chord progressions

概要：

在本文中，我们引入了新的方法，并讨论了基于文本的LSTM（长时间内存）网络的自动音乐组合的结果。 提出的网络旨在在两个案例研究中学习代表和弦进行和鼓轨的文本文档内的关系。 在实验中，对于这两种情况，RNN（循环神经网络）都显示出良好的结果，而基于字符的RNN（char-RNN）只能成功地学习和弦进行。 提出的系统可用于全自动组合或半自动组合系统，通过控制模型的多样性参数来帮助人类组成音乐。

关键词：LSTM，RNN，自动组成音乐，和弦进展

1简介

Music can be represented as a sequence of events and thus it can be modelled as conditional probabilities between musical events. For example, in harmonic tracks, some chords are more likely to occur than others given the previous chords, while the whole chord progressions often depend on the global key of the music. In many automatic composition systems, these relationships are simplied by assuming that the probability of the current state p(n) only depends on the probabilities of the states in the past p(n - k)…p(n-1). A sequence of musical events - notes, chords, rhythm patterns - is generated by predicting the following event given a seed sequence.

音乐可以表示为一系列的事件，因此可以将其模拟为音乐事件之间的条件概率。 例如，在谐波轨道中，与以前的和弦相比，一些和弦更可能发生，而整个和弦的进行通常取决于音乐的全局key。 在许多自动组合系统中，通过假设当前状态p（n）的概率仅取决于过去p（n-k）... p（n-1）中的状态的概率来简化这些关系。 一系列音乐事件 - 音符-和弦，节奏模式 - 通过预测给定种子序列的以下事件来产生。

2 Keunwoo Choi, George Fazekas, and Mark Sandler

Hidden-Markov models (HMMs) are one of the most popular methods to model and predict sequences. HMMs are based on the assumption of k = 1 (Markov assumption) given the sequence of the hidden states which determine the visible states. Choral harmonisation is generated after learning chorales by Bach using a HMM in [1], where 229 and 153 chorales are used for training and testing, respectively. In [14], chord progressions are generated to accompany a

melody to help non-musicians to create music using a HMM. The training set of the HMM consists of 298 lead sheets including pop, rock, R&B, jazz, and country music. In the prediction, the system generates chords using a 62\*62 chord transition probability matrix. In practice, HMMs had been the most suitable for time-series modelling given the data, computing power, and feasible optimisa-tion strategies. One of the drawbacks of HMMs, however, is the ine\_ciency of1-of-K scheme of its hidden states. The memory of HMM is limited to log2(N) bits when there is N hidden states, which requires to learn N2 parameters for the transition matrix.

Hidden-Markov（HMM）模型是模拟和预测序列最流行的方法之一。HMM基于k = 1（马尔可夫假设）的假设，给定了隐藏状态的序列确定可见状态的序列。合唱协调是在巴赫利用HMM [1]中学习合唱团之后产生的，其中分别使用229和153首合唱团进行训练和测试。在[14]中，产生的和旋过程去完成一个旋律，帮助非音乐家用HMM来创造音乐。 HMM的训练集包括流行音乐，摇滚，R＆B，爵士乐和国际音乐等298个lead sheets。在预测中，系统使用62\*62和弦转换概率矩阵来产生和弦。在实践中，由于数据，计算能力和可行的优化策略，HMMs最适合时间序列建模。然而，HMM的一个缺点是其隐藏状态的1-K方案的无效性。当存在N个隐藏状态时，HMM的存储器被限制为log2（N）位，这需要学习转换矩阵的N的2次个参数。

Recurrent Neural Networks (RNNs) allow for incorporating long term depen-dency in the model. Jordan net [8], a simple version of RNNs, is used in [12] to generate chord sequences. In [13], melodies were generated by a system named CONCERT, which is trained on sets of 10 Bach pieces to generate melodies by note-wise prediction. One ability CONCERT lacks is to learn the global structure; this may be due to the diculty of training an RNNs. Theoretically, it can remember in\_nitely long sequences, although in practice it is limited by the vanishing gradient problem [7]. During the training of back-propagation through time, the gradient is extremely diminished by multiplications of sigmoid operations.

循环神经网络（RNNs）允许结合模型的长期依赖。 约旦网[8]，一个简单版本的RNNs被用于[12]来产生和弦序列。 在[13]中，旋律是由一个名为CONCERT的系统产生的，该系统由10组巴赫乐曲组成，通过音符预测来产生旋律。 CONCERT缺乏的一个能力是学习全球结构; 这可能是由于训练RNN的困难。 理论上，它可以记住无限长的序列，尽管实际上它受到消失的梯度问题的限制[7]。 在通过时间的反向传播的训练期间，通过S形运算的乘法极大地减小了梯度。

LSTM (Long Short-Term Memory) units solved this vanishing gradient problem [7]. LSTM allows the gradient to be owed by a separate path with not multiplication but addition operations. LSTM is adopted in [4] to learn 12-bar Blues chords progressions and melodies. [11] focuses on the generation of percussive tracks using LSTM network. The network in [11] directly analyses audio content of drum tracks and learns features using LSTM.

LSTM（长短期记忆网络）单元解决了这个消失的梯度问题[7]。 LSTM允许梯度由不带乘法但加法运算的独立路径承担。 在[4]中采用LSTM来学习12个蓝调的蓝调和弦演奏和旋律，重点是使用LSTM网络生成打击乐曲。 该网络直接分析鼓轨道的音频内容，并使用LSTM学习功能。

In this paper, we introduce applications of character- and word-based RNNs with LSTM units for the automatic generation of jazz chord progressions and rock music drum tracks. Our work is di\_erentiated from previous works by two aspects. First, the LSTM networks we use are designed to learn from text data rather than representations of musical symbols or numeric values. Directly using text data minimises the overall design procedures for the encoding-decoding scheme and the network. Second, compared to the previous research [1],[4],[14], the LSTM networks is trained using a large dataset, which enables itself to learn more complex relationship between the chords in a large set.

在本文中，我们介绍了基于字符和字的RNN与LSTM单元的应用，用于自动生成爵士和弦进行和摇滚乐鼓。 我们的工作与以前的作品有两个方面的差异。 首先，我们使用的LSTM网络旨在从文本数据中学习，而不是音符或数值的表示。 直接使用文本数据可最大限度地减少编码解码方案和网络的整体设计过程。 第二，与以前的研究[1]，[4]，[14]相比，LSTM网络使用大型数据集训练，这使得自己能够在大集合中学习更多的和弦之间的关系。

In the Section 2, we introduce character-based RNNs and the proposed architecture. In Sections 3 and 4, two case studies on the applications of RNNs to automatic composition are explained - for jazz chord progressions and rock music drum tracks. We conclude the work in Section 5.

在第2节中，我们介绍基于字符的RNN和建议的架构。 在第3和第4节中，介绍了关于RNN应用于自动组合的两个案例研究 - 用于爵士和弦进行和摇滚乐鼓。 我们在第5节总结工作。

Text-based LSTM networks for Automatic Music Composition

基于文本的自动组合音乐的LSTM网络。

2 The architecture

2.1 Character-based RNNs

2架构

2.1基于字符的RNN

Char-RNNs are RNNs with character-based learning [15], which is different from the conventional approach of word-based learning. When applied to the texts of chords, a char-RNN predict a vector that corresponds to a character (e.g. predict a based on C:m, and predict j based on C:ma), while a word-RNN predicts a vector, which corresponds to a unique chord (e.g. C:maj based on G:maj ). Using char-RNNs in this work has two merits.

Char-RNN是具有字符学习的RNN [15]，与传统的基于字的学习方法不同。 当应用和弦的文本时，char-RNN预测与字符相对应的向量（例如，基于C：m预测，并基于C：ma预测j），而词RNN预测一个向量，其中 对应于独特的和弦（例如基于G：maj的C：maj）。 在这项工作中使用char-RNNs有两个优点。

First, it is based on the minimal assumption - there is no constraint on the form of the text representation of music. It is worth inspecting if RNNs can learn musical information with such a weak assumption.

首先，它是基于最小假设---对音乐的文本表示形式没有限制。 如果RNN可以用这么弱的假设来学习音乐信息是需要检查的。

Second, fewer number of characters means fewer number of states, which results in reducing the computational cost. From a linguistics point of view, sequence learning methods such as HMMs and RNNs used to model each word(e.g.chord) as a state as it is natural to \_nd the relationships between words. One drawback of word-based learning is the large number of states (or the size of vocabulary); in natural language processing tasks, the vocabulary size easily exceeds few thousands to even few millions. In the proposed method the size of the chord vocabulary is 1,259. With character-based prediction, this decreases to 39.

第二，较少的字符数字意味着更少的状态数量，这导致降低计算成本。 从语言学的观点来看，诸如HMM和RNN之类的序列学习方法被用来将每个单词（例如弦）建模为一个状态，因为它自然地减少了单词之间的关系。 基于单词的学习的一个缺点是自然语言处理任务中的大量状态（或词汇大小），词汇大小容易超过数千甚至数百万。 在提出的方法中，和弦词汇的大小是1,259。 以字符为基础的预测，这减少到39。

The price of small vocabulary size is a longer sequence; as we need to learn character by character, the model should remember a longer sequence of states. As mentioned above, the LSTM unit helps the RNNs to learn this long-termdependency better. This trade-off does not necessarily benefit as in Section

小词汇大小的代价是一个更长的序列，因为我们需要逐个学习字符，模型应该记住更长的状态序列。 如上所述，LSTM单位帮助RNN更好地了解这种长期依赖性。 这个权衡不一定会在第一节中得到。

2.2 The Proposed Architecture

We use two LSTM layers, each of which consists of 512 hidden units. Dropout of 0.2 is added after every LSTM layers [16].

2.2提议的架构

我们使用两个LSTM层，每层由512个隐藏单元组成。 在每个LSTM层之后添加0.2的舍弃[16]。

We use the Keras deep learning framework [2]. During the optimisation, categorical cross-entropy is used as a loss function and optimisation is performed by ADAM [9]. This optimiser shows an equivalent \_nal performance to Stochastic Gradient Descent with Nestrov momentum with faster convergence.

我们使用Keras深度学习框架[2]。在优化过程中，分类交叉熵被用作损失函数，ADAM进行优化[9]。 这个优化器显示了具有Nestrov势头的随机梯度下降的等效性，具有更快的收敛性。

The prediction is stochastic. In each prediction for time index n, the network outputs the probabilities of every states. To make the system tunable,We employ a diversity parameter \_ in the prediction stage (see Eqn. ??), which suppresses (a < 1) or encourages (a > 1) the diversity of prediction by re-weighting the probabilities. In detail, the probabilities of i-th state, pi, are re-weighted as^ pi = exp (log(pi)/a). Then, one of the states is selected by sampling a state according to the re-weighted probabilities. As stated in Section 3, we perform experiments

预测是随机的。 在时间索引n的每个预测中，网络输出每个状态的概率。 为了使系统可调，我们在预测阶段采用分集参数\_（参见方程式），其通过重新加权概率来抑制（a <1）或鼓励（a> 1）预测的多样性。 详细地，第i个状态pi的概率被重新加权为^ pi = exp（log（pi）/ a）。 然后，通过根据重新加权的概率对状态进行采样来选择状态之一。 如第3节所述，我们进行实验。

4 Keunwoo Choi, George Fazekas, and Mark Sandler

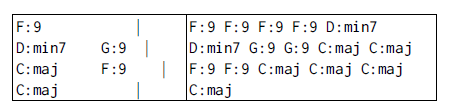


Table 1: An example of the text representations of chord progressions in score (left) and the training data (right). A 4-bar chord progression is generally written in the form on the left, where the positions of the chords loosely indicate the chord change timings. On the right, the text show how the score on the left is represented in the training data. Here, the chords for every quarter notes are explicitly written and bar indicators are removed.

表1：乐谱（左）和训练数据（右）中和弦进行的文本表示的示例。

4bar和弦进程通常以左边的形式写入，其中和弦的位置松动地指示和弦变化定时。 在右侧，文本显示了左侧的得分如何在训练数据中表示。 在这里，每四分之一音符的和弦被明确写入，并且条形指示符被去除。

be predicted to complete a chord in char-RNNs while each state correspond to

a chord in word-RNNs.

The dataset, code and audio \_les are released on web.2

预计在char-RNN中完成和弦，而每个状态对应于词RNN中的和弦。

数据集，代码和音频文件在web.2上发布

3 Case Study 1: Chord progressions

3.1 Representation

The goal of this experiment is to generate chord progressions by training an LSTM network on jazz chord progressions. Here, we do not use any musical interpretation of the chords such as binary vectors to represent pitch and chords (as in [5]) but completely rely on their text representations. Table 1 shows an example of a chord progression and the corresponding texts. The left is an example of a chord notation in The Realbook score, where the positions of chords are loosely related to the timings of chord changes. The score on the left is converted into the text on the right, which speci\_es every chord for each quarter note.

3案例研究1：和弦进行

3.1代表

这个实验的目标是通过在爵士乐和弦进行中训练LSTM网络来产生和弦进行。 在这里，我们不使用诸如二进制向量之类的和弦的任何音乐解释代表音调和和弦（如[5]），而是完全依赖于它们的文本表示。 表1显示了和弦进行和相应文本的示例。 左边是“真实”评分中和弦符号的一个例子，其中和弦的位置与和弦变化的时间松动相关。 左边的得分被转换成右边的文字，其中每个音符的每个和弦都是特定的。

We used 2; 486 scores from The Realbooks and The Fakebooks as training data. Every score file was parsed from band-in-a-box format to .xlab format. Then they were transposed to the key of C while every blank quarter note was filled with its preceding chord as in the Table 1. Finally, we put\_START\_ and \_END\_ flags (any distinctive words can be used as ags) at the beginning and the end of each score.

我们用“真实书”和“假书”的2.486分作为培训资料。 每个分数文件从一个盒子格式解析为.xlab格式。 然后将它们转置到C的关键字，而每个空白四分音符都填满了前面的和弦，如表1所示。最后，我们将\_START\_和\_END\_标志（任何独特的词可以用作标志）在开始和结束 的每个成绩。

Although the key was transposed to C, only 867 (out of 2,846) scores end with C:maj (30%), followed by 489 G:7 (17%), 186 C:maj6 (7%), 52 F:maj (2%), and 1,252 scores end with the others { 237 chords (46%). This is because the The Realbook chord progressions usually end with chords for a turn-around to make the progressions natural to repeat the score.

虽然key被转换到C，但只有867（2,846）得分以C：maj6（30％）结束，其次是489 G：7（17％），186 C：maj6（7％），52 F：maj （2％），1,252分与其他成绩结束（237和弦（46％））。 这是因为“真书”和弦的进行通常以和弦结尾，以便转换，使的自然的重复得分。

Text-based LSTM networks for Automatic Music Composition

基于文本的自动音乐组合的LSTM网络

There were 1.259 unique chords in the training dataset. In other words, the vocabulary size of word-RNN was 1; 259. However there were only 39 characters in total, which signi\_cantly reduced the computation of char-RNN. The total numbers of chords (words) and characters were 539; 609 and 3; 531; 261,respectively.

训练数据集中有1.259个独特的和弦。 换句话说，词RNN的词汇大小为1.259.然而，总共只有39个字符，这显着地减少了char-RNN的计算。 和弦（字）和字符总数为539; 609和3;分别为511. 261。

3.2 Results

We set the system to output a chord progression for every diversity parameter after every iteration. In this paper, we present four results from each networks (char-RNNs and word-RNNs), part of which are reported in the Table 2. For simplicity, we added bar symbols j and removed repeating chords in the same bar, e.g. j C:7 C:7 C:7 C:7 j reduced to j C:7 j and j C:7 C:7 E:min E:min j

reduced to j C:7 E:min j.

3.2结果

我们设置系统在每次迭代之后为每个分集参数输出和弦进行。 在本文中，我们提出了来自每个网络（char-RNN和word-RNN）的四个结果，其中的一部分在表2中报告。为了简单起见，我们添加条形符号j并在同一个条中去除重复的和弦，例如。 7 C：7 C：7 C：7 j缩小到j C：7 j和j C：7 C：7 E：min E：min j 缩小到j C：7 E：min j

First, both char-RNN and word-RNN showed well-structured results. They learned the local structures of chords and bars after su\_cient number of iterations. In the result, the majority of chords continued for multiples of four,implying a single chord for within a bar. They also learned the local relationships

between ags and chord. After one iteration, the ags are not placed properly as in the table 2 (a), where \_END\_ is not followed by \_START\_ but repeats itself.

首先，char-RNN和word-RNN都显示出结构良好的结果。 他们在迭代次数之后学到了和弦和bar的本地结构。 结果，大多数和弦持续四分之一的倍数，意味着一个单一的和弦在一个bar。 他们还了解到了和弦之间的地方关系。 在一次迭代之后，ags没有如表2（a）那样正确放置，其中\_END\_不在\_START\_后面，而是重复。

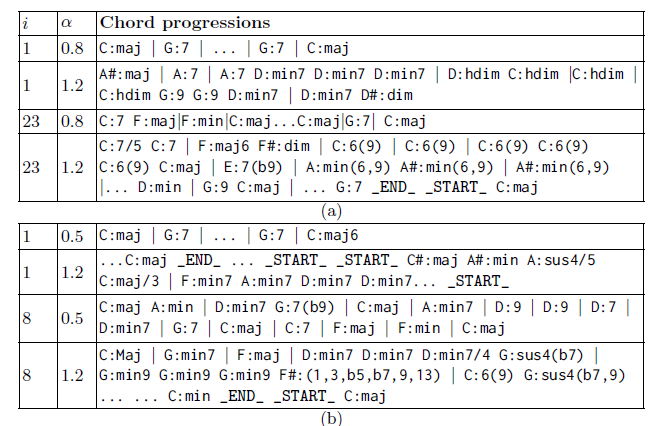


Table 2: Chord progressions generated by char-RNN (a) and word-RNN (b). Bar symbols (j) are inserted for readability and repeated chords in each bar are omitted.

表2：char-RNN（a）和word-RNN（b）产生的和弦进度。 为了可读性插入条形符号（j），并省略每个条中的重复和弦。

Keunwoo Choi, George Fazekas, and Mark Sandler

As training continues, the ags start to appear in a sequence of \_END\_ \_START\_ C:maj as in the training texts. The last chords of the score, i.e., the chord before \_END\_ are not always same as the \_rst chord (C), which is also natural as they vary in the training file.

随着训练的继续，AGS开始出现在\_END\_ \_START\_序列c：作为少校在训练文本。 得分的最后一个和弦，即\_END\_之前的和弦并不总是与第一和弦（C）相同，这也是自然的，因为它们在训练文件中不同。

Second, after sufficient training, both results showed chord progressions that lie in Jazz grammar. Examples are II-V-I progressions (D:min7- G:9 - C:maj), passing chords (A:dim - Ab:dim - G:min7), modal interchange chords (C:min6,Db:maj ) and substitutions (B:7 as a tritone subdominant of F:7) in char-RNN;modal interchanges (G:min7), circle of fifths (Eb:sus - Gb:maj6 - B:maj7), and descending bass (C:maj6,9 - B:dim - A:min7 - Ab:7) in word-RNN. The authors noticed a subtle difference between the results from the two approaches. The results from word-RNN are more conventional progressions than those of char-RNN. However, it cannot be the fundamental difference of the two approaches.Instead, it may be caused by the difference of effective lengths between char and word-RNNs layers - they have the same length of state sequences, but it results in a longer chord sequence in the word-RNN as mentioned in Section 2.2. In other words, the short memory of char-RNN may result in predictions that seem to be less constrained and stereotyped.

第二，经过充分的训练，这两个结果都显示出爵士乐语法中的和弦进行。 示例是II-VI进行（D：min7-G：9-C：maj），通过和弦（A：dim-Ab：dim-G：min7），模态交换弦（C：min6，Db：maj） （G：min7），五分之一圈（Eb：sus-Gb：maj6-B：maj7）和下降的低音（C：maj6）中的（B：7作为F：7的三分音调优势的F：7） ，9-B：dim-A：min7-Ab：7）。 作者注意到两种方法的结果之间有微妙的差异。 来自词RNN的结果比char-RNN的结果更为常规。 然而，它不能是两种方法的根本区别。但是，它可能是由char和word-RNN层之间的有效长度的差异引起的 - 它们具有相同的状态序列长度，但是它导致更长的和弦序列 在第2.2节中提及的词RNN中。换句话说，char-RNN的短暂记忆可能会导致预测似乎不太受约束和定型。

4 Case 2. Drum Tracks

4.1 Representation

4例子2.鼓轨

4.1表示

There are issues when applying LSTM networks to drum tracks including finding a way to create and effective text representation. Both chord progressions and drum tracks are sequences of simultaneous events (pitches and drum components). However, drum tracks do not have a meaningful and compressive representation such as chord and it necessitate an encoding strategy of the track into text. We also need a finer time resolution as generally there are more than four events in a bar.

To encode simultaneous events in a track into texts, we used a binary representation of pitches, i.e., components of drums - kick, snare, hi-hats, cymbals, and tom-toms. For example, 100000000 and 010000000 represent kick and snare, respectively, and a simultaneous playing of kick and snare can be represented by 110000000.

将LSTM网络应用于鼓轨时有问题，包括找到创建和有效的文本表示的方法。 和弦进行和鼓轨是同步事件（音高和鼓组件）的序列。 然而，鼓轨不具有有意义和压缩的表示，例如和弦，并且必须将轨道的编码策略变成文本。 我们还需要更精细的时间分辨率，通常在酒吧中有超过四个事件。为了将轨道中的同步事件编码为文本，我们使用了音调的二进制表示，即鼓的组件 - 踢，圈套，hi-hats, cymbals和 tom-toms。 例如，100000000和010000000分别代表踢球和圈套，同时玩踢球和圈套可以代表110000000。

For efficient representation and learning, only nine components were allowed; kick, snare, open hi-hats, closed hi-hats, three tom-toms, crash cymbal, and ride cymbal.3 We limited the number of events in a bar to 16 by quantising the drum track by 16th-note.

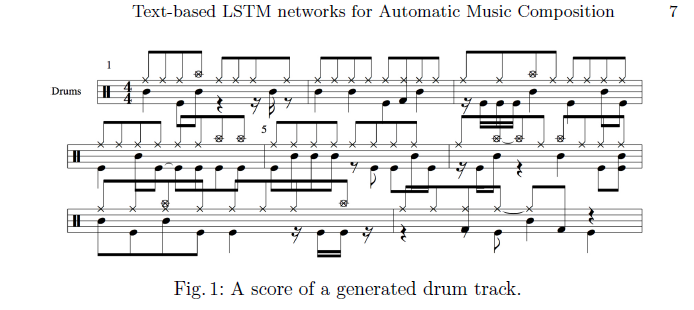
为了有效的表示和学习，只允许9个组成部分; 踢，圈套，打开的帽子，封闭的帽子，三个脚趾，坠落钹和乘坐钹3。我们将一个酒吧的事件数量限制为16，通过量化鼓轨16分。

In the experiment, we first loaded 60 midi files of drum tracks of Metallica and quantised them. Then they were encoded into the above described binary representation. We also added a flag \_BAR\_ as an annotation of the bar segments in order to check if the networks learns the local structure.

There can be theoretically 29 = 512 words, but there are supposedly much fewer words because the combinations of drum components that are played simultaneously are limited. The size of the word vocabulary in the training file is 119 and the \_le consists of 2,141,692 words in total.

在实验中，我们首先加载了Metallica的60个MIDI文件，并量化了它们。 然后将它们编码成上述二进制表示。 我们还添加了一个标志\_BAR\_作为条形段的注释，以便检查网络是否学习本地结构。

理论上可以有29 = 512个单词，但据说可以少得多的单词，因为同时播放的鼓组件的组合是有限的。 训练文件中单词词汇的大小为119，文件总共包含2,141,692个单词。



4.2 Results

Char-RNNs turned out to fail to learn the drum tracks and output arbitrary 0's and 1's without any structures (the results have no spaces or \_BAR\_fl ags).The length of network may be too short to learn the long-term relationship between characters. In char-RNNs, representing a single bar requires 16 events10 characters=160 time steps. Encoding music sequences with only two characters- 0 and 1 (+space to for segmentation) - is an extreme approach for char-RNNs.In this paper, we therefore only report the result of word-RNNs.

4.2结果

Char-RNN证明没有学习鼓轨，并且没有任何结构输出任意0和1（结果没有空格或\_BAR\_fl ags）。网络长度可能太短，无法识别字符之间的长期关系。 在char-RNN中，代表一个单独的条需要16个事件10个字符= 160个时间步长。 编码仅具有两个字符的音乐序列（0和1（+分割空间））是charRNN的极端方法。因此，本文仅报告单词RNN的结果。

Figure 1 shows one example of our results - a part of the generated track

Controlling a provides a way to tune the technical virtuosity of the track. Since large a increases the probabilities of occasional events, large \_ (=1:5)results in tracks with many fill-ins with tom-toms and a crash cymbal. On the other hands, when a< 1, the track almost never contains anything but kick,snare, and hi-hats. As a result, it is possible to use a combination of small and large \_ in a drum track generator that is guided by user, who specifies where to add fill-ins.

控制a提供了一种调整轨道技术精湛度的方法。 由于大的增加了偶发事件的概率，因此大a（= 1：5）导致轨迹与许多填充tom和hihats 另一方面，当一个<1，轨道几乎从来没有包含任何东西，但踢，圈套和帽子。 结果，可以在由用户指导的鼓轨发生器中使用小和大的组合，谁指定在哪里添加填写。

5 Conclusion

We introduced an algorithm of text-based LSTM networks for automatic composition and reported results for generating chord progressions and rock drum tracks. Word-RNNs showed good results in both cases while char-RNNs only successfully learned chord progressions. The experiments show LSTM provides a way to learn the sequence of musical events even when the data is given as text. With the diversity parameter, the proposed algorithm can be used as a tool that helps human composers. In the future, a more complex network with the capability of learning interactions within music (instruments, melody/lyrics) will be examined for a more complete automatic composition algorithm.

我们引入了基于文本的LSTM网络的算法，用于自动构图和报告结果，用于产生和弦进行和摇滚乐轨道。 在这两种情况下，词RNNs都表现出良好的效果，而人类RNN只能成功地学习和弦进行。 实验表明，即使将数据作为文本给出，LSTM也提供了一种学习音乐事件序列的方式。 使用多样性参数，所提出的算法可以用作帮助人类作曲家的工具。 将来，将检查一个具有学习音乐（乐器，旋律/歌词）中的互动能力的更复杂的网络，以获得更完整的自动构图算法。