SONG FROM PI: A MUSICALLY PLAUSIBLE NETWORK

FOR POP MUSIC GENERATION

歌曲从PI：音乐的一个可扩展网络

POP音乐发生

ABSTRACT

We present a novel framework for generating pop music. Our model is a hierarchical

Recurrent Neural Network, where the layers and the structure of the hierarchy encode our prior knowledge about how pop music is composed. In particular, the bottom layers generate the melody, while the higher levels produce the drums and chords. We conduct several human studies that show strong preference of our generated music over that produced by the recent method by Google. We additionally show two applications of our framework: neural dancing and karaoke, as well as neural story singing.

概要

我们提出一个创造流行音乐的新颖框架。 我们的模型是一个层次循环神经网络，层和层结构编码了我们关于流行音乐组合的以前的知识。 特别地，底层产生旋律，而较高的层产生鼓和和弦。 我们进行了几项人类研究，表明我们所产生的音乐对Google最近采用的方法产生的强烈偏爱。 我们还展示了我们的框架的两个应用：神经网络跳舞和卡拉OK，以及神经网络讲故事。

1．INTRODUCTION

Neural networks have revolutionized many fields. They have not only proven to be powerful in performing perception tasks such as image classification and language understanding, but have also shown to be surprisingly good “artists”. In Gatys et al. (2015), photos were turned into paintings by

exploiting particular drawing styles such as Van Gogh’s, Kiros et al. (2015) produced stories about

images biased by writing style (e.g., romance books), Karpathy et al. (2016) wrote Shakespeare inspired novels, and Simo-Serra et al. (2015) gave fashion advice.

一，简介

神经网络使许多领域发生了革命。 他们不仅被证明在执行诸如图像分类和语言理解等感知任务方面是强大的，而且还表现出好的令人惊讶的“艺术家”才能。 在Gatys等人 （2015），把照片通过利用特定的绘画风格变成了绘画，如梵高，Kiros等人 （2015）制作了关于由写作风格（例如浪漫书籍）偏见的图像的故事，Karpathy等人 （2016）写了莎士比亚灵感小说，Simo-Serra等人 （2015）给了时尚咨询。

Music composition is another artistic domain where neural based approaches have been proposed.

Early approaches exploiting Recurrent Neural Networks (Bharucha & Todd (1989); Mozer (1996);

Chen & Miikkulainen (2001); Eck & Schmidhuber (2002)) date back to the 80’s. The main variations

between the different models is the representation of the notes and the outputs they produced,

which typically encode melody and chord. Most of these approaches were single track, in that they

produced only one note per time step. The exception is Boulanger-lewandowski et al. (2012) which

generated polyphonic music, i.e., simultaneous independent melodies.

基于神经的方法提出，音乐作品是另一个艺术领域。

早期采用循环神经网络的方法（Bharucha＆Todd（1989）; Mozer（1996）;

Chen＆Miikkulainen（2001）; Eck＆Schmidhuber（2002））可以追溯到80年代。 主要变化

不同模型之间的代表是他们生产的乐谱和产出，这通常编码旋律和和弦。 这些方法中的大多数都是单轨，因为它们每个时间段只产生一个音符。 例如Boulanger-lewandowski等人 （2012）

产生复音音乐，即同时独立的旋律。

In this paper, we aim to generate pop music, where the melody but also chords and other instruments make up what is typically called a song. We draw inspiration from the Song from \_ by Macdonald 1,a piano video on Youtube, where the pleasing music is created from a sequence of digits of π. This video shows both the randomness and the regularity of music. On one hand, since any possible digit sequence is a subset of the π digit sequence, this implies that pleasing music can be created even from a totally random base signal. On the other hand, the composer uses specific rules such as A Harmonic Minor scale and harmonies to convert the digit sequence into a music sheet. It is these rules that play the key role in converting randomness into music.

在本文中，我们的目标是产生流行音乐，其中旋律，而且和弦和其他乐器都会产生通常被称为歌曲的东西。 我们从麦克唐纳1号的歌曲中吸取灵感，麦克唐纳1号是Youtube上的钢琴影片，其中由π这个数字创建令人愉快的音乐。 这个视频显示了音乐的随机性和规律性。 一方面，由于任何可能的数字序列是π数字序列的子集，这意味着即使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，例如A Harmony Minor scale和Harmony，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。

Following the ideas of Songs from π, we aim to generate both the melody as well as accompanying effects such as chords and drums. Arguably, these turn even a not particularly pleasing melody into a well sounding song. We propose a hierarchical approach, where each level is a Recurrent Neural Network producing a key aspect of the song. The bottom layers generate the melody, while the higher levels produce drums and chords. This enables the drum and chord layers to compensate for the melody in order to produce appleasing music. Adopting the key idea from Songs from π we condition our model on the scale type allowing the melody generator to learn the notes that are typically played in a particular scale.

遵循歌曲从π序列产生的想法，我们的目标是产生旋律以及伴随的和弦和鼓的效果。 可以说，这些旋律甚至不是特别令人愉快的旋律成为一个好听的歌曲。 我们提出一种分级方法，其中每个级别都是循环神经网络，产生了这首歌的关键方面。 底层产生旋律，而较高层产生鼓和和弦。 这使得鼓和和弦层能够补偿旋律，以便产生令人愉快的音乐。 采用来自用π产生歌曲的关键思想，我们对我们模型在音节上进行调整，使旋律发生器学习通常以特定音阶播放的音符。

We train our model on 100 hours of midi music containing user-composed pop songs and video

game music. We conduct human studies with music generated with our approach and compare it

against a recent approach by Google, showing that our songs are strongly preferred over the baseline.

In our human study we also perform an ablation analysis of our model. We additionally show two

new applications: neural dancing and karaoke as well as neural music singing. As part of the first

application we generate a stickman dancing to our music and lyrics that can be sung with, while in

the second application we condition on the output of Kiros et al. (2015) which writes a story about an

image and convert it into a pop song. We refer the reader to http://www.cs.toronto.edu/songfrompi/

for our demos and results.

我们用100小时的midi音乐训练我们的模型，这个midi音乐包含用户组成的流行歌曲和视频游戏音乐。 我们使用我们的方法进行音乐的人体学研究，并将其与Google最近采用的方法相比较，表明我们的歌曲比基线更受欢迎。在我们的人类研究中，我们还对模型进行消融分析。 我们还展示了两个新的应用：神经跳舞和卡拉OK以及神经音乐歌唱。 作为第一个应用程序的一部分，我们生成了一个stickman给我们带有歌词的歌曲伴舞，而在第二个应用中，我们对Kiros等的输出进行了调整。 （2015），它写了一个关于图像的故事，并将其转换成流行歌曲。 我们将读者引用到http://www.cs.toronto.edu/songfrompi/了解我们的演示和结果。

2 RELATED WORK

Generating music has been an active research area for decades. It brings together machines learning researchers that aim to capture the complex structure of music (Eck & Schmidhuber (2002);

Boulanger-lewandowski et al. (2012)), as well as music professionals (Chan et al. (2006)) and enthusiasts(Johnson; Sun) that want to see how far a computer can get to be a real composer. Real-time

music generation is also explored for gaming (Engels et al. (2015)).

2.相关工作

几十年来，音乐的产生一直是活跃的研究领域。 它汇集了机器学习研究，旨在捕捉音乐复杂结构（Eck＆Schmidhuber（2002）;Boulanger-lewandowski等 （2012））还有音乐专业人士（Chan et al。（2006））和爱好者（Johnson; Sun），想看到一台电脑可以成为一名真正的作曲家还有多远。 即时的音乐还为了游戏在探索中（Engels等（2015））。

Early approaches mostly instilled knowledge from music theory into generation, by using rules of

how music segments can be stitched together in a plausible way, e.g., Chan et al. (2006). On the

other hand, neural networks have been used for music generation since the 80’s (Bharucha & Todd

(1989); Mozer (1996); Chen & Miikkulainen (2001); Eck & Schmidhuber (2002)). Mozer (1996)

used a Recurrent Neural Network that produced pitch, duration and chord at each time step. Unlike

most other neural network approaches, this work encodes music knowledge into the representation.

Eck & Schmidhuber (2002) was first to use LSTMs to generate both melody and chord. Compared

to Mozer (1996), the LSTM captured more global music structure across the song.

早期的方法主要是通过使用音乐片段以合理的方式拼接在一起的规则，将音乐理论的知识融入到生成中，例如Chan等人（2006年）。 另一方面，自80年代以来，神经网络被用于音乐的生成（Bharucha＆Todd（1989）; Mozer（1996）; Chen＆Miikkulainen（2001）; Eck＆Schmidhuber（2002））。 Mozer（1996）使用了一个神经网络，在每个时间步长产生音调，音长和和弦。不像大多数其他神经网络方法，这项工作将音乐知识编码到表示中。Eck＆Schmidhuber（2002）首先使用LSTM来产生旋律和和弦。 与Mozer（1996）相比，LSTM在这首歌中获得了更多的全球音乐结构。

Like us, Kang et al. (2012) built upon the randomness of melody by trying to accompany it with

drums. However, in their model the scale type is enforced. No details about the model are presented,

and thus it is virtually impossible to compare to. Boulanger-lewandowski et al. (2012) propose to

learn complex polyphonic musical structure which has multiple notes playing in parallel through

the song. The model is single-track in that it only produces melody, where as in our work we aim

to produce multi-track songs. Just recently, Huang & Wu (2016) proposed a 2-layer LSTM that,

like Boulanger-lewandowski et al. (2012), produces music that is more complex than a single note

sequence, and is able to produce chords. The main novelty of our work over existing approaches is a

hierarchical model that incorporates knowledge from music theory to build the neural architecture,

and produces multi-track pop music (melody, chord and drum). We also present two novel fun

applications.

像我们一样，康等人 （2012）建立在旋律随机性的基础上，试图用鼓伴奏。 然而，在他们的模型中，模型类型被强制执行。 没有提供关于模型的细节，因此实际上是不可能比较的。 Boulanger-lewandowski等 （2012）提出通过歌曲学习复杂的和弦音乐结构，它具有并行播放的多个音符。 该模式是单轨，它只产生旋律，而在我们的工作中，我们的目标是制作多轨音乐。 最近黄和吴（2016）提出了一个2层LSTM，，像Boulanger-lewandowski等人 （2012），生产比单音符序列更复杂的音乐，并且能够产生和弦。 我们对现有方法的工作的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）。 我们还提出了两个有趣的应用程序。

3 CONCEPTS FROM MUSIC THEORY

We start by introducing the basic notation and definitions from music theory. A note defines the

basic unit that music is composed of. Music follows the 12-tone system, i.e., 12 is the cycle length

of all notes. The 12 tones are: C, C]=D[, D, D]=E[, E, F, F]=G[, G, G]=A[, A, A]=B[, B.

A bar is a short segment of time that corresponds to a specific number of beats (notes). The boundaries of the bar are indicated by vertical bar lines. Scale is a subset of notes. There are four types of scales most commonly used: Major (Minor), Harmonic Minor, Melodic Minor and Blues. Each scale type specifies a sequence of relative intervals (or shifts) which act relative to the starting note. For example, the sequence for the scale type Major is 2 —〉 2 —〉 1 —〉 2 —〉 2 —〉 2 —〉 1. Thus, C Major specifies the starting note to be C, and applying the relative sequence of shifts yields: The subset of notes specified by C Major is thus C, D, E, F, G, A, and B (a subset of seven notes). All scales types have a subset of seven notes except for Blues which has six. In total we have 48 unique scales, i.e. 4 scale types and 12 possible starting notes. We treat Major and Minor as one type as for a Major scale there is always a Minor that has exactly the same set of notes. In music theory, this is referred to as Relative Minor.

3音乐理论的概念

我们首先介绍音乐理论的基本音符和定义。音符定义了音乐组成的基本单位。音乐遵循12音系，即12是所有音符的周期长度。 12个音是：

 A调是一段很短的时间段，相当于一定数量的节拍（音符）。调的边界由垂直条线表示.音阶是音符的一个子集。最常用的有四种音阶：主要（次要），调和次要，旋律小调和蓝调。每个比例类型指定相对于起始音符起作用的相对间隔（或移位）的序列。例如，音阶Major的序列是2→2→1→2→2→2→1。因此，C Major将起始注释指定为C，并应用相对移位序列产生：C Major指定的音符子集为C，D，E，F，G，A和B（7个音符的子集）。除了有六个蓝调之外，所有尺度类型都有七个音符的子集。总共有48个独特的量表，即4种音阶类型和12种可能的起始音符。我们把主要和次要作为一个类型，对于一个主要的规模，总是有一个具有完全相同的音符的小调。在音乐理论中，这被引用作为关系小调。

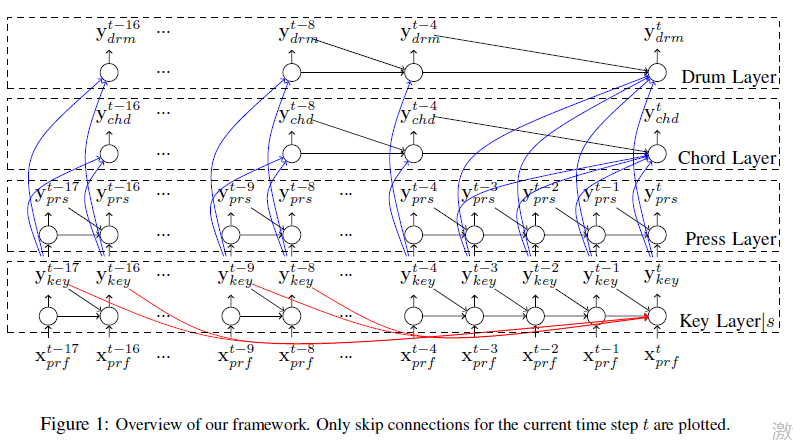


Figure 1: Overview of our framework. Only skip connections for the current time step t are plotted.

图1：我们的框架概述。 仅绘制当前时间步长t的跳过连接。

Chord is a group of notes that sound good together. Similarly to scale, a chord has a start note and

a type defining a set of intervals. There are mainly 6 types in triads chords: Major Chord, Minor Chord, Augmented Chord, Diminished Chord, Suspended 2nd Chord, and Suspended 4th Chord.

The Circle of Fifths is often used to produce a chord progression. It maps 12 chord starting notes

to a circle. When changing from one chord to another chord, moving to a nearby chord on the circle

is often preferred as this forms a strong chord progression that produces the sense of harmony

和弦是一组一起演奏听起来很好的音符。 类似于音阶，和弦具有起始音符和定义一组间隔的类型。 有共六种主要的和弦，主要和弦，小和弦，增强和弦，和弦和弦，暂停的和弦和暂停的。 五号圆圈通常用于产生一个和弦。 它将12个和弦开始音符映射到一个圆圈。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，因为这形成了和谐感强烈的和弦进程。

4 HIERARCHICAL RECURRENT NETWORKS FOR POP MUSIC GENERATION

We follow the high level idea behind the Song from π to define our model. In particular, we generate music with a hierarchical Recurrent Neural Network where the layers and the structure of the hierarchy encode our prior knowledge about how pop music is composed. We first outline the model and describe the details and justifications for our choices in the subsections that follow.

4音乐综合回归网络POP音乐生成

我们跟着宋从π的高层思想来定义我们的模型。 特别是，我们用层次性循环神经网络生成音乐，层次和层次结构编码了我们关于流行音乐组合的先前知识。 我们首先概述模型，并在下面的小节中描述我们的选择的细节和理由。

We condition our generation on the scale type, as this helps the model to pick up the regularities in

pop songs. We encode melody with two random variables at each time step, representing which key

is being played (the key layer) and the duration that the key will be pressed (the press layer). The

melody is generated conditioned on the scale, which does not vary across the song as is typically the

case in pop music. We assume the drums and the chords are independent given the melody. Thus

conditioned on the melody, at each time step we generate the chord (the chord layer) as well as the

drums (the drum layer). The output at all layers yields the final song. We refer the reader to Fig. 1

for an illustration of our hierarchical model.

我们把我们这一产生音乐放在音阶类型上，因为这有助于这个模型来获得流行歌曲的规律。 我们在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）。 旋律是按音阶产生的，而在歌曲中不一样，流行音乐通常是这样。 我们假设鼓和和弦是独立被给予的旋律。 因此在旋律上进行调节，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。 我们将读者引用到图1用于说明我们的层次模型。

4.1 THE ROLE OF SCALE

It is known from music theory that while in principle each song has 12 tones to choose from, most of

the notes are in fact only using the six (for Blues) or seven (for other scales) tone subsets specified

by the scale rule. We found that by conditioning the music generator on scale it captures these

regularities more easily. However, we do not enforce the notes to be generated from the subset and

allow our model to generate notes outside the scale.

4.1音阶的作用

从音乐理论知道，原则上每首歌曲都有12种声调可供选择，但大多数音符其实只使用了比例规则指定的六（蓝调）或七（其他音阶）音调子集。 我们发现，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，我们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。

We confirm the above musical fact by analysing over 100 hours of pop song music from

the midi man dataset. Since scale is defined relative to a starting note, we first try to factor out

its influence and normalize all songs to have identical start note. To identify the scale of a song, we

compute the histogram over the 12 tones and match it with the 48 tone subsets of 4 scale types with

12 different start notes. We then normalize all songs to have start note C by applying a constant shift on all notes.

通过分析来自midi man数据集的超过100小时的流行歌曲，我们确认了上述的音乐事实。 由于音阶相对于起始音符的定义，我们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，我们计算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。

This allows us to categorize any song into 4 scale types. Since this shift affects all notes

at once, it does not affect how the song sounds (its harmony). Our analysis shows that for all notes

in all Major scale songs, 94:66% are within the tone subset. For Harmonic Minor, Melodic Minor,

and Blues the percentage of notes that belong to the main tone set is 87:16%, 85:11%, and 90:93%,

respectively. We refer the reader to Fig. 2, where the x-axis denotes the percentage of within-scale

notes of a song, and the y-axis indicates how many songs in the dataset have that percentage. Note

that the majority of the notes follow the scale rule. Furthermore, different scale types have different

inlier distribution. We thus represent scale with a single random variable s 2 f1; \_ \_ \_ ; 4g which is

fixed for the whole song, and condition the model on it.

这使我们可以将任何歌曲分为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）。我们的分析表明，对于所有主要音阶的歌曲中的所有音符，94.66％在音调子集内。 谐波小调，旋律小调和蓝调的音符属于主音调集合的百分比为87.16%, 85.11%, 和 90.93%。 我们将读者引用到图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。

4.2 TWO-LAYER RNN FOR MELODY GENERATION

We represent the melody with two random variables per time step: which key is pressed, and the

duration of the press. A Recurrent Neural Network (RNN) is used to generate the key condition on

the scale. Then conditioned on the output of the key layer, a second RNN generates the duration of

the press at each time step.

In this paper we take advantage of LSTMs, which in their most basic form (single layer) compute

the hidden state ht given the input xt by

xt by

ft = \_(Wf [xt; ht􀀀1]) (1)

it = \_(Wi[xt; ht􀀀1])

ot = \_(Wo[xt; ht􀀀1]) （1）

fCt = tanh(WC[xt; ht􀀀1])

Ct = ft \_ Ct􀀀1 + it \_ fCt

ht = ot \_ tanh(Ct)

with Wf ;Wi;Wo;WC learnable parameters. Here f, i, o, C, e C, and h denote the forget gate, input

gate, output gate, cell state, input cell state and hidden state.

4.2 两层rnn产生旋律

我们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

In particular, we model the key layer with a two-layer LSTM with 512-dimensional hidden state,

which outputs a note (key) at each time step. Note that we condition on scale s, thus we have

different parameters per scale. We only allow notes between C3 to C6 as notes outside this range

are usually too low or too high to sound good. We remind the reader that given a scale, seven (or six

for blues) out of the twelve notes (per octave) are statistically more plausible, however we allow the

model to choose from all 12. This results in a 37-dimensional output, as there are 36 possible notes

corresponding to 3 octaves with 12 notes per octave, plus silence. Let htkey be the hidden state of

the second key decoder layer at time t. We compute the probability of each key using the softmax:

is the row of V (the output embedding matrix of notes), corresponding to note ytkey.。

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：

（2）

是V行（音符的输出嵌入矩阵），与ytk相对应。

As input to the LSTM we use a vector that concatenates multiple features: a one-hot encoding of the

previous generated note yt􀀀1key , Lookback features, and the melody profile. The Lookback features

were proposed by Google Magenta (Waite et al.) to make it easier for the model to memorize

recently produced notes and potentially repeat them. They include skip connections from two and

one bar ago (a bar is 8 consecutively played notes), i.e., yt􀀀16key and yt􀀀8key . They also contain two

additional features, indicating whether the last generated key has been copied from one or two bars

ago, i.e. 1(yt􀀀1key ; yt􀀀1􀀀8key ) and 1(yt􀀀1key ; yt􀀀1􀀀16key ). They also add a 5-dimensional feature indicating a binary encoding of the current time t. This helps the model keep track where in a 4􀀀bar range it is, and thus produce music accordingly.

作为LSTM的输入，我们使用连接多个功能的向量：先前生成的音符yt􀀀1key，回读功能和旋律曲线的单热编码。 回顾功能由Google Magenta（Waite等人）提出，以便模型更容易记住最近制作的音符，并可能重复。 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。

In addition, we introduce a new feature which we refer to as the melody profile. Intuitively, the

profile represents the high-level music flow. To get the profile for each song, we compute the local

note histogram at each time step with width of two bars, and cluster all local histograms within the song into 10 clusters via k-means. We order the 10 clusters with mean note ordered from low to

high as cluster 1 to 10, and apply moving averages on the cluster id sequence to encourage local

smoothness. This results in a 10-dimensional one-hot vector representation of the cluster id for each

time step. This additional information allows the user to set the melody’s ups and downs of the song

此外，我们介绍一个我们称为旋律配置文件的新功能。 直观地，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。

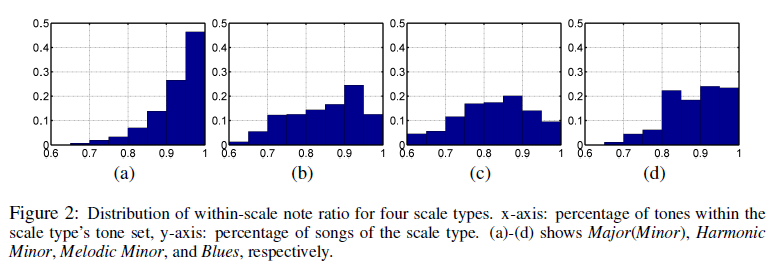


Figure 2: Distribution of within-scale note ratio for four scale types. x-axis: percentage of tones within the

scale type’s tone set, y-axis: percentage of songs of the scale type. (a)-(d) shows Major(Minor), Harmonic

Minor, Melodic Minor, and Blues, respectively

图2：四种尺度类型的尺度内音符比率的分布。 x轴：音调的百分比 音阶类型的音色集，y轴：音阶类型的歌曲百分比。 （a） - （d）显示主要（次要），谐波 次要，旋律小调和蓝调。

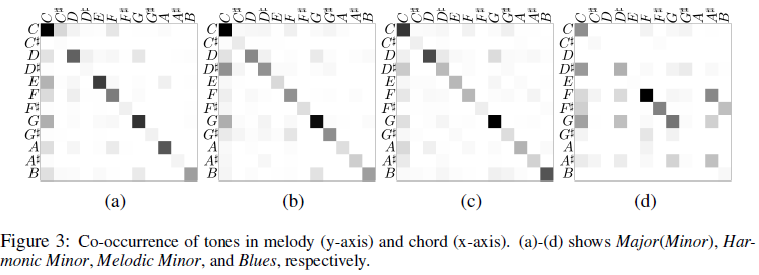


Figure 3: Co-occurrence of tones in melody (y-axis) and chord (x-axis). (a)-(d) shows Major(Minor), Harmonic Minor, Melodic Minor, and Blues, respectively.

图3：旋律（y轴）和弦（x轴）中的音调的同现。 （a） - （d）分别显示Major（Minor），Harmonic Minor，Melodic Minor和Blues。

The keys alone are not sufficient to describe how the melody is performed. Additionally we also need to know the duration that each key needs to be pressed for. Towards this goal, conditioned on the  
melody, we generate the duration of each key with a two-layer LSTM with a 512-dimensional hidden  
state. We represent the duration of pressing as a forward counting sequence that is conditioned on  
the generated melody. The press outputs 1 when a new key is pressed, and sequentially outputs 2,  
3, 4 and so on as the key is held on. When the current key is released, the press counter is reset to  
1. Compared to the event on-off representation of Waite et al., our representation learns the melody  
flow and how to press separately. This is important, as Waite et al. has extremely unbalanced output  
distributions dominated by the repeat-of-holding event. We represent press yprs t as a 8-dimensional  
one-hot vector. The input to our LSTM is yprs t-1, concatenated with the 37-dimensional one-hot  
encoding of the melody key ykey t .

单独的key不足以描述如何执行旋律。此外，我们还需要知道每个key需要按下的持续时间。为了达到这个目标，在旋律的基础上，我们用一个512层隐藏状态的双层LSTM生成每个key的持续时间。我们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.与Waite等人的事件开关表示相比，我们的表示法学习旋律流程以及如何单独按压。这很重要，Waite等由重复举行事件主导的输出分布极不平衡。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。

4.3 CHORD AND DRUM RNN LAYERS  
We studied all existing chords in our 100 hours of pop music. Although in principle a chord can be  
any arbitrary combination of multiple notes, we observed that in the actual music data 99*:*19% of  
the chords belong to one of 72 chord classes (6 types *×* 12 start notes). Fig. 3 shows the correlation  
between the melody’s tone and the starting note of the chord playing at the same time. It can be  
seen that chord is strongly correlated with melody. These two findings inspire our design. We thus  
represent chord **y***chd t* as a one-hot encoding with 72 classes, and predict it using a two-layer LSTM  
with a 512-dimensional hidden state. We generate one chord at each time step. The input is **y***chd t-*4  
concatenated with **y***key t-*3:*t*.

4.3旋律和鼓RNN层

我们在我们的100小时流行音乐中研究了所有现有的和弦。原则上，和弦可以是多个音符的任意组合，我们观察到在实际音乐数据中99.19％的和弦属于72和弦类（6种×12开始音符）之一。 图3显示了旋律的音调和和弦演奏的起始音符的相关性。 可以看出，和弦与旋律强烈相关。 这两个结果激发了我们的设计。 我们就这样将和弦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。

4.4 LEARNING

We use cross-entropy as our loss function to train each layer. We follow the typical training strategy

where we make predictions at each layer and time step but feed in ground-truth information to the

next. This effectively decomposes training, and allows to train all layers in parallel. We use the

Adam optimizer, a learning rate of 2e-3 and a learning rate decay of 0.99 after each epoch for 10

epochs.

4.4学习

我们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。

4.5 MUSIC SYNTHESIS: PUTTING ALL THE OUTPUTS TOGETHER

To synthesize music we first randomly choose a scale and a profile xprf . For generating xprf , we

randomly choose one cluster id with a random duration, and repeat until we get the desired total

length of the music sequence. We then perform inference in our model conditioned on the chosen

scale, and use xprf as input to our key layer. At each time step, we sample a key according to

P(ytkey). We encode it as a one-hot vector and pass to the press, chord and drum layers. We sample

the press, chords and drums at each time step in a similar fashion.

4.5音乐合成：将所有输出一起输出

要合成音乐，我们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，我们根据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。

Before putting the outputs across layers together, we further adjust the generated sequences at the

bar level. For melody, we first check at each bar if the first step is a continuation of a previous note

or silence. If it is the latter, we find the first newly pressed note within the bar and move it to the

beginning of the bar. We do similarly for the windows of two half-bars as well as the four quarterbars.

This makes the melody more likely to be on the beat, and generally sounds better. We verify

this in our experiments.

在将输出层叠在一起之前，我们进一步调整在条形图级别生成的序列。 对于旋律，我们首先检查每个栏，如果第一步是上一个音符或静音的延续。 如果是后者，我们会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。 我们对于两个半柱的窗户以及四个四分之一杆也同样做。 这使得旋律更有可能在节拍中，一般听起来更好。这在我们的实验中我们验证了这个

For chord, we generate one chord at each half bar, which is the majority of all single step chord

generations. Furthermore, we incorporate the rule of chord progression in the Circle of Fifths as

between chords pairwise smooth terms, and compute the final chord using dynamic programming.

For drum, we generate one pattern at each half bar.。

对于和弦，我们在每半个bar上产生一个和弦，这是所有单和弦大多数产生的。 此外，我们将和弦进化规则结合在五弦环中，如和弦成对平滑项之间，并使用动态规划计算最终和弦。 对于鼓，我们在每个半bar上生成一个图案。

Our model generates with scale starting note C, and then applies a constant shift to generate music

with other starting notes. Besides scale, which instrument to use is also customizable. However, we

simply set all instruments as grand piano in all experiments, as the effect and musical meaning of

different instrument combinations is beyond the scope of this paper.

我们的模型产生出音阶从音符C开始，然后应用持续的平移去产生其他音符开始的音乐。 除了音阶，使用哪种仪器也是可定制的。 然而，在所有实验中，我们只需将所有乐器设置为大钢琴，因为不同乐器组合的效果和音乐含义超出了本文的范围。

5 EXPERIMENTS

To train our model, we took 100 hours of pop music from midi man which consists of user-composed

pop songs and video game music. In our generation, we always use 120 beats per minute with 4 time

steps per beat. However, songs in the dataset can have arbitrary speed. To neutralize the effect of

this, we detect the most frequent interval between two adjacent notes for each song, and iteratively

divide or multiply this interval by 2 until it falls in the range between 0:25s and 0:5s. We use this

as a measure of the song’s beat duration. We then adjust the song’s temporal axis so that all songs

have the same beat duration of 0:5s.

5实验

为了训练我们的模特，我们从包括用户组成的流行歌曲和视频游戏音乐的midiman手中拿了100个小时的流行音乐。 在我们这一代，我们总是每分钟使用120次，每次4次。 但是，数据集中的歌曲可以有任意的速度。 为了中和这个效果，我们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续时间的度量。 然后，我们调整歌曲的时间轴，使所有歌曲具有0.5s的相同拍子持续时间。

A MIDI file can be separated into different channels/tracks, where the 9th channel is specifically

preserved for drums. We categorize the rest of non-drum tracks into melody, chord, and else, by

simply setting thresholds on average number of unique notes within a bar and average number of

note changing within a bar, as chords are by definition repetitive. Fig. 4 shows an example of our

music generation.

MIDI文件可以分成不同的通道/轨道，其中第9个通道专门保留给鼓。 我们将其余的非鼓轨分类为旋律，和弦，另外通过简单地设置条形中的独特音符的平均数量的阈值以及条形中平均音符数量的变化，因为和弦的定义是重复的。图4显示了我们的音乐生成的例子。

Figure 4: Example of our music generation. From top to bottom: melody, chord and drum respectively

图4：我们的音乐生成的例子。 从上到下：旋律，和弦和鼓

Table 1: Human evaluation of music generated by different methods: ours and Waite et al.’s Magenta. Ours-

MO and Ours-NA are short for Ours Melody Only and Ours No Alignment. We allowed neutral votes, thus the

sum of the pair is less than 100%.

表1：人们评价通过不同方法生成的音乐：我们和Waite等人的洋红色。 Ours-MO和Ours-NA是Ours Melody Only和Ours No Alignment的缩写。 我们允许中立票，因此，对的总和小于100％。

To evaluate the quality of our music generation, we conduct a human survey with 27 participants.

In the survey, participants are presented with several pairs of 30-second music clips, and are asked

to vote which clip in the pair sounds better. We gave no other information about what they are

listening to. They are also allow to submit a neutral vote in case they cannot decide between the

two choices. In our study, we consider three cases: our full method versus Magenta Waite et al., our

method with melody only versus Google Magenta (Waite et al.), and our method versus our methodwithout the temporal alignment described in Sec.4.5. We randomly generated 10 songs per methodand randomly shuffled each pair.

为了评估我们的产生的音乐的质量，我们对27位参与者进行了调查。 在调查中，参与者都有几对30秒的音乐片段，并被要求投票哪一对声音更好。 我们没有提供关于他们正在听什么的其他信息。如果他们不能在两个选择之间作出决定，他们也允许提交一个中立的投票。 在我们的研究中，我们考虑了三种情况：我们的完整方法与Magenta Waite等人的比较，我们的方法仅与麦考红（Waite等人）进行在旋律方面进行比较，我们的方法与我们的方法相比，没有第4.5节所述的时间对齐。 我们每个方法随机生成10首歌曲，并将每一对都随机洗牌。

As shown in Table 1, most participants prefer songs produced by our method compared to Magenta.

Participants also made comments such as music sounds better with percussion than piano alone,

and multiple instruments with continuous play is much better. This confirms that our multi-layer

generation improves music quality. Few participants also point out that drums sound too different

and do not participate to the melody perfectly, which indicates that further improvements can be still

made. In the second comparison, we study if the quality improvement of our method is only caused

by adding chords and drums, or is also related to our two-layer melody generation with alignment. It

can be seen that without chords and drums, the score drops as expected, but is still much higher than

the Magenta baseline. This is because our method produces less recursion and silence, and faster

and more accurate tempo as mentioned by the participants. In the last comparison, most participants

prefer our full method than the no-alignment version, since beats are more subtle and better timed.

This proves the usefulness of temporal alignment.

如表1所示，与Magenta产生的歌曲相比，大多数参与者喜欢我们的方法产生的歌曲。参与者还发表了一些意见，比如音乐听起来比只有钢琴演奏好多了，并且多种乐器持续的演奏听起来更好。这证实了我们的多层产生音乐提高了音乐的质量。少数参与者还指出，鼓声听起来不一样，不能完美地参与产生旋律，表明可以进一步改进。在第二个比较中，我们研究如果我们的方法的质量改进只是通过添加和弦和鼓，甚至与我们的双重旋律对齐。可以看出，如果没有和弦和鼓，分数将按预期下降，但仍远远高于megenta基线。这是因为我们的方法产生较少的递归和沉默，并且更快，和参与者提到更准确的节奏。在最后的比较中，大多数参与者喜欢我们的完整方法与不对齐的版本相比，因为节拍是更微妙和更好的时间。这证明了时间对齐是有用的。

Finally we study our model’s capabilities to generate new music. Towards this goal, we generated

100 sequences of 50 seconds of length using different random initializations. Then for each sequence,

we search for the longest sub-sequence of keys that matches part of the training data, and

record its length. We find out that with 1 hour of training data, the mean matching sub-sequence

length is 3:46s. With 100 hours of training data, the mean length increases to 4:65s, since there are

more possible matches. The sequences are very small and thus, our model is able to generate new

music.

最后，我们研究我们的模型产生新音乐的能力。 为了实现这一目标，我们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。

6 APPLICATIONS

In this section we demonstrate two novel applications of our pop music generation framework. We

refer the reader to http://www.cs.toronto.edu/songfrompi/ for the music videos.

6应用

在本节中，我们演示了我们流行音乐生成框架的两个新的应用。 我们将读者引用到http://www.cs.toronto.edu/songfrompi/，了解音乐视频。

6.1 NEURAL DANCING AND KARAOKE

神经跳舞和唱歌

In our first application, we attempt to generate both music and a stickman dancing to it, as well as

a sequence of karaoke-like text that people can sing along with. To learn the relationship between

music and dance, we download 1 hour of video from the game Just Dance, as well as the MIDI files

for songs included in the video from different sources. We use the method in Newell et al. (2016)

to track single-frame 2D human pose in the videos. We process the single-frame tracking result to

ensure left-right body consistency through time, and then use the method of Zhou et al. (2016) to

convert the 2D pose sequence into 3D. Example results are shown in Fig. 5. We observe that our

pose processing pipeline is able to extract reasonable human poses most of the time. However, the

quality is not perfect due to tracking failure or video effects. We define pose similarity as average

euclidean distance of all joints, and cluster poses into 456 clusters. We used Frey & Dueck (2007)

as the number of clusters is large.

在我们的第一个应用程序中，我们试图产生音乐和stickman跳舞，以及人们可以一起唱歌的一系列卡拉OK文字。要了解音乐与舞蹈之间的关系，我们从游戏Just Dance下载1小时的视频，以及来自不同来源的视频中包含的歌曲的MIDI文件。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，然后使用Zhou等人的方法（2016）至

将2D姿态序列转换为3D。示例结果如图1所示。我们观察到我们的姿态处理管道能够大部分时间提取合理的人体姿势。然而，由于跟踪故障或视频效果，质量并不完美。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。因为群集数量很大, 我们使用Frey＆Dueck（2007）.

We learn to generate a stickman dancing by adding another dancing layer on top of the key layer,

just like for drum and chord. We generate one pose at each beat, which is equivalent to 4 time

steps or 0.5 seconds in a 120 beat-per-minute music. In particular, we predict one of the 456 pose

clusters using a linear projection layer followed by softmax. We use cross-entropy at each time step

as our loss function. At inference time, we further apply moving average to temporally smooth the

generated 3D pose sequence.

我们学习通过在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。 我们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。

To learn the relationship between music and lyrics, we collect 51 hours of lyrics data from the

internet. This data contains 50 hours of text without music, and the rest 1 hour are songs we collected

from Just Dance. For the music part, we temporally align each sentence in the lyrics with the midi

music by using the widely-existing lrc format, which records the time tag at the beginning of every

sentence. We select words that appear at least 4 times, which yields a vocabulary size of 3390

including unknown and end-of-sentence. Just as for dance, we generate one word per beat using

another lyrics layer on top of the key layer.

我们从互联网收集51小时的歌词数据去学习音乐和歌词之间的关系。 此数据包含50小时无音乐的文字，其余1小时是我们从Just Dance收集的歌曲。 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。 我们选择出现至少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。

6.2 NEURAL STORY SINGING

神经讲故事唱歌

In this application our aim is to sing a song about a photo. We first generate a story about the

photo with the neural storyteller Kiros et al. (2015) and try to accompany the generated text with

music. We utilize the same 1 hour dataset of temporally aligned lyrics and music. We further include

the phoneme list of our 3390 vocabulary as we also want to sing the story. Starting from the text

produced by neural storyteller, we arrange it into a temporal sequence with 1 beat per word and a

short pause for end-of-sentence, where the pause length is decided such that the next sentence starts

from a new bar. As our dataset is relatively small, we generate the profile conditioned on the text,

which has less dimensions compared to the key. This is done by a 2-layer LSTM that takes as input

the generated profile at the last time step concatenated with a one-hot vector of the current word, and

outputs the current profile. We then generate the song with our model given the generated profile.

The generated melody key is then used to decide on the pitch frequency of a virtual singer, assuming

the key-to-pitch correspondence of a grand piano. We further constrain that the singer’s final pitch is

always in the range of E3 to G4, which we empirically found to be the natural pitch range. We then

replace all words outside the vocabulary with the sound Ooh, and play the rendered singing with the

generated music.

在这个应用程序中，我们的目标是唱一首关于照片的歌曲。我们首先用讲故事者kiros等人生成关于照片的故事。 （2015），并尝试伴随音乐生成的文本。我们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，我们也想唱这个故事。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。

7 CONCLUSION AND FUTURE WORK

We have presented a hierarchical approach to pop song generation which exploits music theory in

the model design. In contrast to past work, our approach is able to generate multi-track music. Our

human studies shows the strength of our framework compared to an existing strong baseline. We

additionally proposed two new applications: neural dancing & karaoke, and neural story singing. We

next discuss the limitations and avenues for future work. As most existing approaches our method’s

objective is to learn to produce music at the note level. This can be unsuitable for music, as music

is flexible and intentionally made to be unpredictable when it is composed. This calls for a deeper

study of music theory, as in this paper we are only scratching the surface.

7结论和未来的工作

我们提出了一种流行歌曲生成的分层方法，它在模型设计中利用了音乐理论。 与以前的工作相比，我们的方法能够产生多轨音乐。 我们的人类研究显示了我们的框架与现有强大基线相比的优势。 我们还提出了两个新的应用：神经跳舞和卡拉OK，以及神经故事歌唱。 我们接下来讨论未来工作的局限性和途径。 作为大多数现有方法，我们的方法的目标是学习在音符级别制作音乐。 这可能不适合音乐，因为音乐是灵活的，并且在组成时有意使其变得不可预测。 这就要求对音乐理论进行更深入的研究，就像本文中我们只是在比较浅表。

REFERENCES

Jamshed J. Bharucha and Peter M. Todd. Modeling the perception of tonal structure with neural

nets. Computer Music Journal, 13(4):44–53, 1989.

Nicolas Boulanger-lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies

in high-dimensional sequences: Application to polyphonic music generation and transcription.

In ICML, 2012.

Michael Chan, John Potter, and Emery Schubert. Improving algorithmic music composition with

machine learning. In 9th International Conference on Music Perception and Cognition, 2006.

Chun-Chi J. Chen and Risto Miikkulainen. Creating melodies with evolving recurrent neural networks.

In International Joint Conference on Neural Networks, 2001.

Douglas Eck and Juergen Schmidhuber. A first look at music composition using lstm recurrent

neural networks. 2002.

Steve Engels, Fabian Chan, and Tiffany Tong. Automatic real-time music generation for games. In

AIIDE Workshop, 2015.

Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. volume

315, pp. 972–976, 2007.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. In

arXiv:1508.06576, 2015.

Allen Huang and Raymond Wu. Deep learning for music. arXiv preprint arXiv:1606.04930, 2016.

Daniel Johnson. Composing music with recurrent neural networks. https://goo.gl/YP9QyR.

Semin Kang, Soo-Yol Ok, and Young-Min Kang. Automatic Music Generation and Machine Learning

Based Evaluation, pp. 436–443. Springer Berlin Heidelberg, 2012.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks.

In ICLR 2016 Workshop, 2016.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba,

and Sanja Fidler. Skip-thought vectors. In NIPS, 2015.

David Macdonald. Song from \_. https://youtu.be/OMq9he-5HUU.

Reddit midi man. Midi collection. https://goo.gl/4moEZ3.

Michael C. Mozer. Neural network music composition by prediction: Exploring the benefits of

psychoacoustic constraints and multi-scale processing. Connection Science, 6(2-3), 1996.

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation.

In ECCV, 2016.

Arnold Schoenberg and Dika Newlin. Style and idea. Technical report, Williams and Norgate

London, 1951.

Edgar Simo-Serra, Sanja Fidler, Francesc Moreno-Noguer, and Raquel Urtasun. Neuroaesthetics in

fashion: Modeling the perception of beauty. In CVPR, 2015.

Felix Sun. Deephear - composing and harmonizing music with neural networks. https://goo.

gl/7OTZZL.

Elliot Waite, Douglas Eck, Adam Roberts, and Dan Abolafia. Project magenta. https:

//magenta.tensorflow.org/.

Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Kosta Derpanis, and Kostas Daniilidis. Sparseness

meets deepness: 3d human pose estimation from monocular video. In CVPR, 2016.