

[业界](#) [移动开发](#) [云计算](#) [软件研发](#) [程序员](#) [极客头条](#) [专题](#)[大数据](#) [数据中心](#) [服务器](#) [存储](#) [虚拟化](#) [NoSQL](#) [安全](#) [云先锋](#)[CSDN首页](#) > [云计算](#)[订阅云计算RSS](#)

# 采用深度学习算法为Spotify做基于内容的音乐推荐

发表于 2015-02-20 06:28 | 9872次阅读 | 来源 个人博客 | 8 条评论 | 作者 Sander Dieleman

[Spotify](#) [深度学习](#) [协同过滤算法](#) [机器学习](#) [deep learning](#) [CNN](#)  
[THEANO](#)

**摘要：**本文概述了作者在Spotify的机器学习实践经验，解释了使用卷积神经网络(CNN)做基于音频的音乐推荐的方法，并提出了有关该卷积网络的实际学习效果的心得。采用了GTX 780Ti GPU，Theano软件框架，小批量梯度下降法。

**【编者按】**本文是比利时根特大学（Ghent University）的Reservoir Lab实验室博士研究生Sander Dieleman所撰写的博客文章，他的研究方向是音乐音频信号分类和推荐的层次表征学习，专攻深度学习和特征学习。以下为译文：

2014年夏天，我在 [网络音乐平台Spotify](#)（纽约）实习，致力于使用卷积神经网络（convolutional neural networks）做基于内容的音乐推荐。本文将介绍我使用的方法，并展示一些初步的结果。

## 概述

这篇文章很长，所以先对各节的内容做一个概述：

**CSDN官方微信**

扫描二维码,向CSDN吐槽

微信号：CSDNnews

**程序员移动端订阅下载****每日资讯快速浏览****微博关注**

- **协同过滤 ( Collaborative filtering )** 一个简单的介绍，包括它的优缺点。
- **基于内容的推荐 ( Content-based recommendation )** 在不能得到使用数据时，该怎么办。
- **深度学习预测听众喜好 ( Predicting listening preferences with deep learning )** 基于音频信号的音乐推荐。
- **效率提升 ( Scaling up )** 有关我在Spotify训练卷积神经网络的一些细节。
- **分析 ( Analysis )**：它在学习什么？卷积神经网络音乐学习一瞥，包含多个音频示例。
- **可以应用在哪里 ( What will this be used for )**？我的工作成果的一些应用潜力。
- **今后的工作 ( Future work )**
- **结论 ( Conclusion )**

## 协同过滤

网络音乐平台Spotify传统上 **主要依靠协同过滤** 驱动音乐推荐。**协同过滤** 的原理是根据历史使用数据确定用户的喜好。例如，如果两个用户收听大致相同类型的歌曲，他们的喜好大概就是类似的。反过来，如果两支歌曲由同一组用户所收听，这些歌曲听起来很可能是相似的。这种信息可以被用来做推荐。

纯粹的协同过滤方法，除了相关的消费模式信息以外，不涉及被推荐物品本身的任何信息：也就是它是内容无关的(content-agnostic)。这个特点使得这种方法可以有非常广泛的应用：例如，同样的一个模型就可以用于推荐书籍、电影或者音乐。

不幸的是，这个特点也导致了它最大的不足。由于依据使用数据，流行物品就会比非流行物品更容易得到推荐，因为前者有更多的使用数据。而这通常是刚好与我们所希望的相反。由于同样的原因，这样的推荐常常是很无聊的，而且是可预知的。

特定到音乐还有另一个问题，就是相似使用模式的内容异质性 ( heterogeneity of content with similar usage patterns )。例如，听众可能一次听过了整张专辑，而专辑中可能包含引曲、终曲、插曲、翻唱曲和混音曲等。它们也许不都是该艺术家典型的作品，因此不是一些好的推荐。可是协同过滤算法不能解决这个问题。

[ 微服务应用容器化场景中常见问题总结 ] Java在企业中被广泛应用，当前，选择Spring Cloud作为微服务开发框架成为一个广泛的趋势。微服务架构的复杂性需要容器技术来支撑，应用需要容器化，并使用CaaS平台来支撑微服务系统的运行。本文探讨的主题是来自于企业级Java应用在容器化过程中遇到的基础问题

10月2日 19:10

转发 | 评论

## 相关热门文章

亚马逊AWS启用新logo后又出新招 7天之后将按...  
变现猫获弘道资本2400万Pre-A融资 构建变现产...  
携手合作伙伴推进容器技术发展， Rancher Labs...

## 热门标签

Hadoop	AWS	移动游戏
Java	Android	iOS
Swift	智能硬件	Docker
OpenStack	VPN	Spark
ERP	IE10	Eclipse
CRM	JavaScript	数据库

而它最大的问题，可能还在于不能推荐新的和非流行的歌曲：如果没有可供分析的使用数据，协同过滤方法就会失效。这就是所谓的冷启动问题。我们希望新音乐能够在发行后的第一时间获得推荐，而且我们也希望能向听众介绍那些他们从来没有听过的很棒的乐队。为了实现这些目标，我们就需要有不同的方法。

## 基于内容的推荐

Spotify根据几个月之前从智能音乐平台 [The Echo Nest](#) 得到的反馈信息，最近已经开始考虑在推荐管道中结合其它信息源，以减少这些问题。可以帮助音乐推荐的信息有很多种：比如标签，艺术家和专辑信息，歌词，从互联网上挖掘到的文稿（评论、访谈...），以及音频信号本身。

在这些信息源中，音频信号可能是最难有效利用的。一方面由于音乐音频信号之间的语义差别（semantic gap）很大，另一方面影响听众喜好的因素又多种多样。有的信息可以比较容易地从音频信号中提取出来，比如音乐的类型和演奏的乐器；而其他的就比较有挑战性了，比如音乐的情绪，和发行的年份（或者时期）；还有一些实际上是不可能从音频中得到的：就像艺术家所在的地理位置和抒情的主题。

尽管存在这些挑战，但是很明显地，歌曲的实际声音极大地影响到听众是否愿意收听。于是通过分析音频信号，预测谁可能欣赏这支歌曲，看起来像个不错的主意。

## 深度学习预测听众喜好

去年十二月，我和同事Aäron van den Oord 在NIPS上发表了一篇关于这个主题的论文，题目是‘[Deep content-based music recommendation](#)’（[基于内容的深度音乐推荐](#)）。我们试图这样解决问题：通过训练回归模型（regression model），预测从协同过滤模型输出歌曲的隐藏表征（latent representations），实现依靠音频信号预测收听喜好。这个方法可以使我们在即使没有使用数据的情况下，也能够从协同过滤空间中预测歌曲表征。（正如可以从论文的题目中推测出来的那样，涉及的回归模型是一个深度神经网络）。

这种方法的基本思想是假定许多协同过滤模型都是把听众和歌曲投射到一个共享的低维度隐空间（latent space）中。在这个空间中歌曲的位置包含了影响听众喜好的各种编码信息。假如有两首歌

Ubuntu

NFC

WAP

## 下载专辑

CSDN 资源优选



Docker

download.csdn.net

十五个Docker的优质资源

2017上半年  
软考各科目  
真题及答案  
解析下载2017年上半年软考各科目考试真题  
及答案解析！

CSDN 资源优选

架构

download.csdn.net

[资源优选]第二十五期：15个关于  
架构的精品资源

CSDN 资源优选

git

download.csdn.net

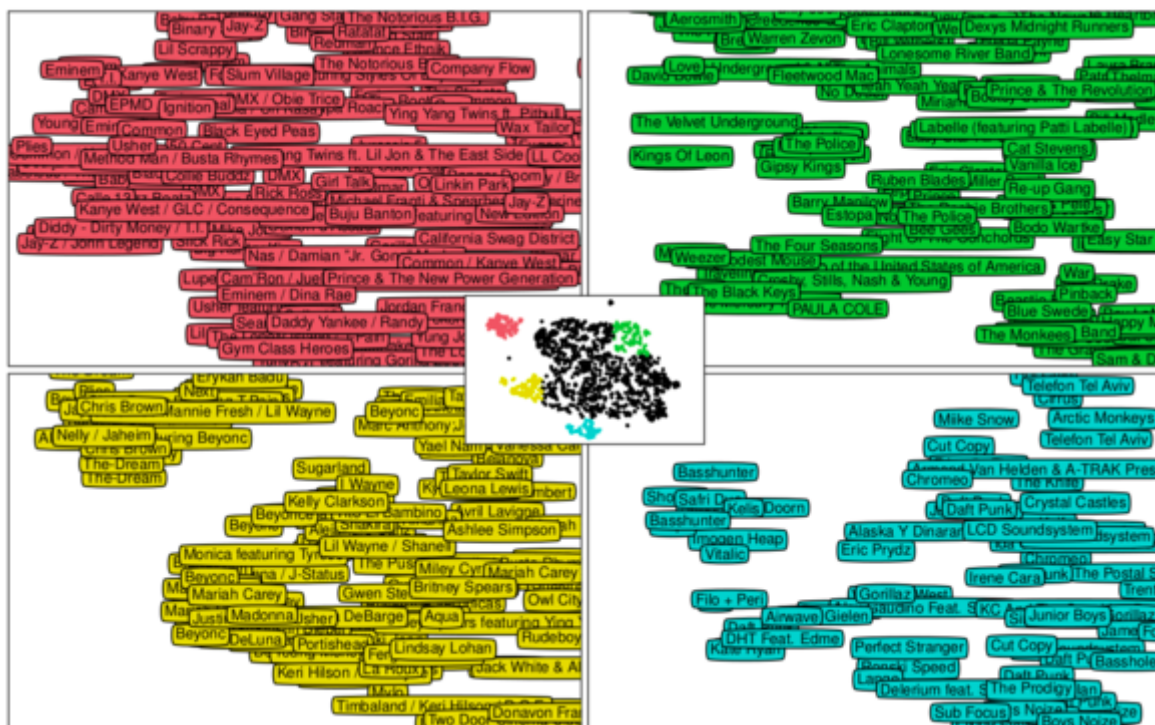
Git各版本汇总专辑



酷播项目

曲在空间上临近，它们很可能是相似的。如果一首歌曲距离一个听众很近，这首歌对他可能就是一个好推荐（如果他还没有听过这首歌）。如果可以通过音频信号预测一首歌曲在这个空间中的位置，那就能够把它推荐给合适的听众，而并不需要历史使用数据。

论文中我们做了可视化效果，即通过投射隐空间中模型的预测结果到使用 [t-SNE 算法](#) 降低的二维空间。从如下的结果图中可以看出，相似的歌曲群集到了一起。说唱乐主要出现在左上角，而电声艺术家聚集在图的底部。



t-SNE算法的隐空间可视化（中部）。几个特写图展现了歌曲投射在特定区域的艺术家的摘自

[Deep content-](#)

[based music recommendation](#), Aäron van den Oord, Sander Dieleman and Benjamin Schrauwen, NIPS

效率提升

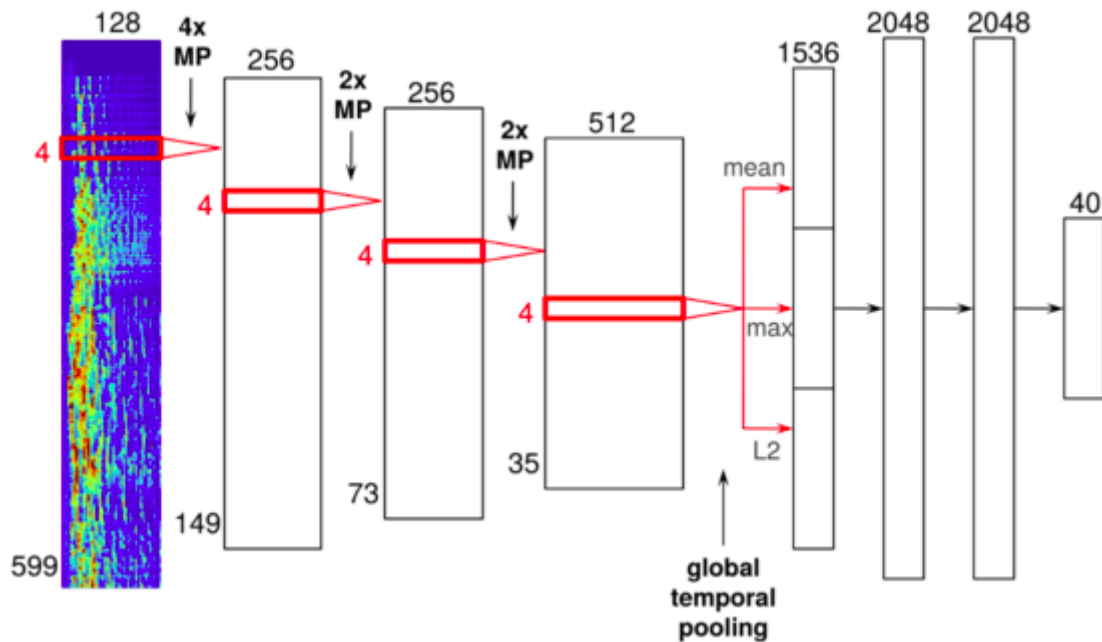
在我们论文中训练的深度神经网络由两个卷积层和两个完全连接层组成。输入是3秒钟音频片断的声谱。对于更长音频片断的预测，只需要把它分成几个3秒钟长的窗口，然后把这些窗口的预测值作平均。

我在Spotify接触了大量歌曲的数据源，以及从不同的协同过滤模型产生的隐藏因素表征（latent factor representations）。我还配备了一台高级的GPU，用于实验运算。它们相当地提升了效率。现在我正在训练总数达7层或8层的卷积神经网络（convnets），使用了大得多的中间表征和更多的参数。

## 架构

下面详细介绍的，是我已经实验过的诸多架构中的一个。它有四个卷积层和三个稠密层（dense layers）。你将看到为了音频信号设计的卷积神经网络，与用于计算机视觉网络任务的传统神经网络，有一些重要的不同。

**警告：**下面有可怕的细节！如果你不太关心诸如ReLU，最大值池化（max-pooling）和小批量梯度下降法（minibatch gradient descent）等细节，尽管直接跳到“分析”段落。



我试验过的用于隐藏因素预测的一种卷积神经网络架构。纵轴是时间轴（在其上卷积）。

网络输入是一系列梅尔声谱（mel-spectrograms），它们有599帧（frames）和128个频点（frequency bins）。梅尔声谱是一种时间-频率表征（time-frequency representation）。是从音频信号的窄重叠窗口傅立叶变换（Fourier transforms）得到的。每一个傅立叶变换构成一帧。然后将这些连续的帧排列成一个矩阵，就形成了这个声谱。最后将频率轴由线性刻度变成梅尔刻度（mel scale）以降低维数，并且采用对数刻度值。

卷积层用红色矩形显示，表现了过滤器滑过输入时的情形。它们使用了线性修正单位（ReLU, 使用的激活函数是 $\max(0, x)$ ）。请注意所有这些卷积都是一维的；卷积仅仅在时间维度出现，而不在频率维度。虽然技术上可以沿着声谱图的两个坐标轴都进行卷积，但现在我并没有这样做。要意识到与图像不一样，声谱图两个轴的意义是不同的（时间和频率），这个非常重要。结果就是，在图像数据中典型的方形过滤器，在这里是没有意义的。



在卷积层之间用最大值池化运算（max-pooling operations）降低时域中间表征采样率，同时增大过程的时不变性。这些操作用“MP”表示。可以看出在每个卷积层中使用了尺寸为4帧的过滤器，在第一与第二卷积层之间是池尺寸为4的最大池化（主要是出于性能方面的考虑），而在其他层之间是池尺寸为2的最大池化。

在最后卷积层的后面，我增加了一个全局时域池化层（global temporal pooling layer）。这一层覆盖整个时间轴，有效地计算时域学习特征的统计值。我引入了三个不同的池化功能：平均值（mean），最大值（maximum）和L2范数（L2-norm）。

我这样做的原因是由于从音频信号中检测到的绝对位置特征，与手头任务的要求不是特别地相关。这里的情况与图像分类不同：在图像分类中，知道一个特征的大概位置就可以了。例如，检测出云朵特征很可能激活图像的上半部分。如果激活在下半部分，可能检测到羊了。在音乐推荐场合，我们通常只对音乐中某些特征整体上是出现还是缺乏感兴趣，所以在时间上做池化是在情理之中的。

另外一种处理方法可以用短音频片段训练网络，通过平均这些窗口的输出得到较长片段的数据，就像我们在NIPS论文中做的那样。不过在模型中引用池化似乎更好一些，因为在学习阶段就可以开始使用这种处理步骤。

2048个线性修正单位的全局池化特征输入到了一串完全连结层（fully-connected layers）。在本网络中这一串只有两个。该网络的最后一层是输出层（output layer），它选用Spotify用过的各种协同过滤算法中的vector\_exp 算法，预测40个隐藏因素。

## 训练

训练网络减少协同过滤模型输出的隐藏因素向量与音频预测之间的均方差(MSE)。这些向量首先要按照单位规范（unit norm）标准作规范化。这样做是为了降低歌曲人气的影响（许多协同过滤模型的隐藏因素向量范数往往与歌曲的人气相关）。在稠密层中采用丢弃法（Dropout）作为正规化方法。

我现在使用的数据集是从Spotify保存的一百万条最流行曲目中截取的30秒长的梅尔声谱。我使用了大约一半曲目用做训练（0.5M），大约5000条做在线验证，其余的用做测试。在训练的时候，通过沿着时间轴做随机的偏移，稍稍调整了声谱，扩展了数据。

实现的网络采用了英伟达（NVIDIA）GeForce GTX 780Ti GPU硬件，[Theano](#)软件框架。使用了小批量梯度下降法，和涅斯捷罗夫冲量因子（Nesterov momentum）。用一个单独的进程进行数据加载和调整，所以当GPU用于大块数据训练时，下一批数据可以并行地加载进来。总共执行了大约750000个梯度更新。我已经记不清训练这个特殊架构的准确时间了，但我记得总的试验时间在18到36小时之间。

## 变化（Variation）

正像我在前面讲到的，这只是我试验过的架构中的一个例子。我已经试过，或将要试验的还有：

- [更多层级](#)！
- 使用最大输出单位（maxout unit）而不是线性修正单位（rectified linear unit）。
- 使用随机池化（stochastic pooling）而不是最大池化（max-pooling）。
- 在网络的输出层引入L2规范化。
- 在时域上拉伸或压缩声谱扩展数据。
- 多重级联各种协同过滤模型输出的隐藏因素向量。

这里是几点工作效果不如预期的地方：

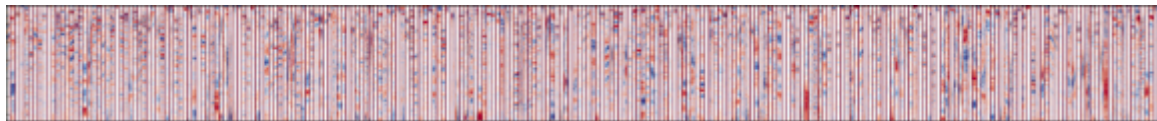
- 用全局时域池化从每个卷积层到网络完全连结层增加的‘旁路’（‘bypass’）连接。基础假设是低级特征的统计结果也对推荐有用，很不幸的是它对训练产生了太多的限制。
- 像[混合密度网络（mixture density networks）](#)一样预测因素的条件方差，得到预测的置信估计，而在隐藏因素预测困难的时候用于识别歌曲。很不幸的是似乎它使训练变得异常困难，而且置信估计也表现得与预期的不同。

## 分析：它在学习什么？

现在到了有点酷的部分：这些网络究竟在学习什么？特征看起来是怎样的？我选择卷积网络解决这个问题的主要原因，是认为根据音频信号的音乐推荐，是一个连接多层次抽象的复杂问题。我希望连续的网络层能像在图像分类问题中那样，渐进地学会更复杂和更多的不变特征。



实际情况看起来确实如此。首先让我们看一看第一个卷积层，它学习直接应用于输入声谱的一组过滤器。这些过滤器是容易可视化的。它们显示在下列图像中。点击就能看到高分辨率版本（5584x562, ~600kB）。负值是红色，正值是蓝色而白色是零值。注意每个过滤器宽度仅仅是四帧。深红色的垂直线将各个过滤器分隔开来。



第一卷积层学习过滤器的可视化。时间轴是横轴，频率轴是竖轴（频率从顶部到底部是增加的）。点击就能看到高分辨率版本（5584x562, ~600kB）。

从这个表示中可以看出，许多过滤器探测出了谐波成分，这体现在不同频率处并行的红蓝条带上。有时候这些条带是向上或向下倾斜的，表示出现了音高的升高或降低。它证明这些过滤器有助于检测人声。

### 低级特征播放表：最大激活

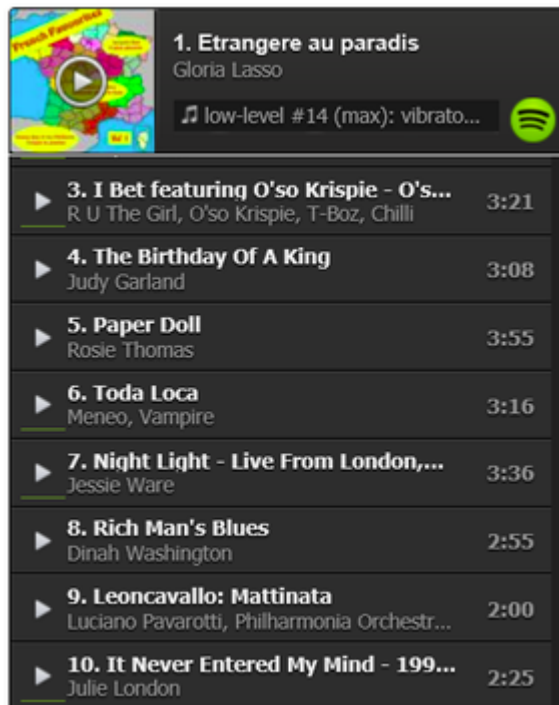
为了对过滤器学习的是什么有更好的理解，我准备了一些最大激活的测试歌曲集播放表。下面是几个例子。网络的第一层有256个过滤器，它们被从0到255编号。注意这个编号是任意的，因为过滤器没有排序。

通过查找在分析的30秒内对给定过滤器最大激活的歌曲，得到了这四个播放列表。我从第一卷积层中选择了几个看起来有趣的过滤器，计算了每个特征表现，然后从整个测试集中查找最大激活。请注意如果要了解过滤器正在接收的内容，应该听取曲目的中段，因为这部分音频信号才是被分析的部分。

下面每个Spotify播放表都有10个曲目。由于版权的问题有些曲目在有的国家收听不到。

过滤器 14：颤音歌唱过滤器

242: 环境气氛 ( ambience )

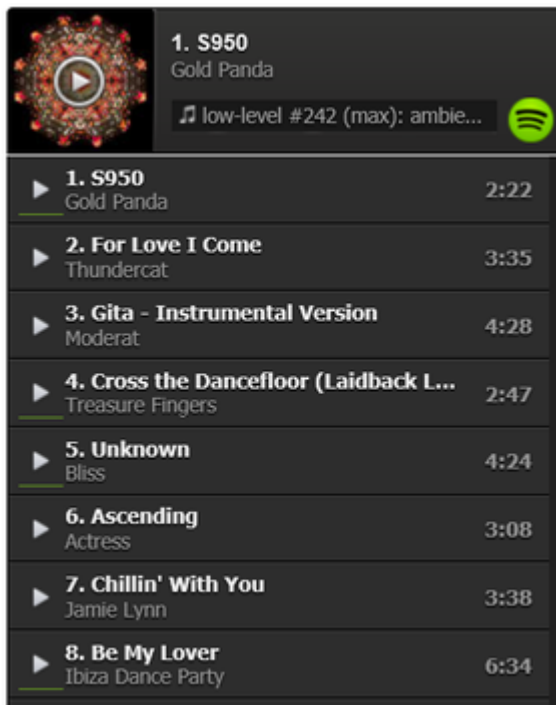


**1. Etrangere au paradis**  
Gloria Lasso

low-level #14 (max): vibrato...

- ▶ **3. I Bet featuring O'so Krispie - O's...** 3:21  
R U The Girl, O'so Krispie, T-Boz, Chilli
- ▶ **4. The Birthday Of A King** 3:08  
Judy Garland
- ▶ **5. Paper Doll** 3:55  
Rosie Thomas
- ▶ **6. Toda Loca** 3:16  
Meneo, Vampire
- ▶ **7. Night Light - Live From London,...** 3:36  
Jessie Ware
- ▶ **8. Rich Man's Blues** 2:55  
Dinah Washington
- ▶ **9. Leoncavallo: Mattinata** 2:00  
Luciano Pavarotti, Philharmonia Orchestr...
- ▶ **10. It Never Entered My Mind - 199...** 2:25  
Julie London

过滤器 250 : 人声大三度 (vocal thirds)

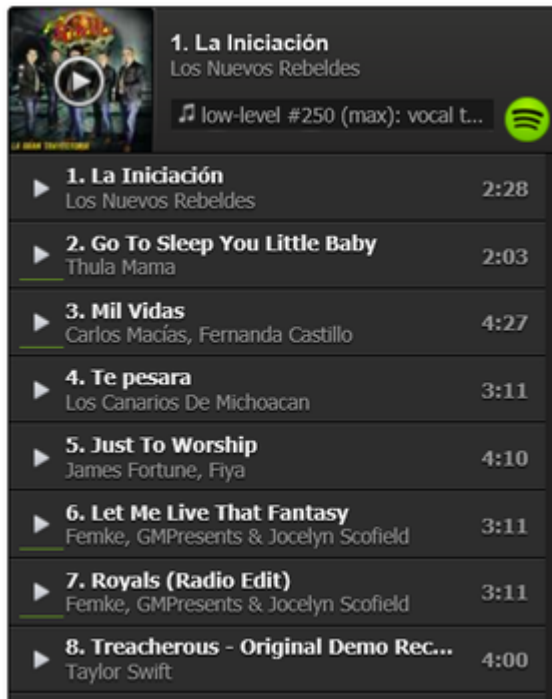


**1. S950**  
Gold Panda

low-level #242 (max): ambie...

- ▶ **1. S950** 2:22  
Gold Panda
- ▶ **2. For Love I Come** 3:35  
Thundercat
- ▶ **3. Gita - Instrumental Version** 4:28  
Moderat
- ▶ **4. Cross the Dancefloor (Laidback L...** 2:47  
Treasure Fingers
- ▶ **5. Unknown** 4:24  
Bliss
- ▶ **6. Ascending** 3:08  
Actress
- ▶ **7. Chillin' With You** 3:38  
Jamie Lynn
- ▶ **8. Be My Lover** 6:34  
Ibiza Dance Party


过滤器 253: 低音鼓



1. La Iniciación  
Los Nuevos Rebeldes

low-level #250 (max): vocal t...

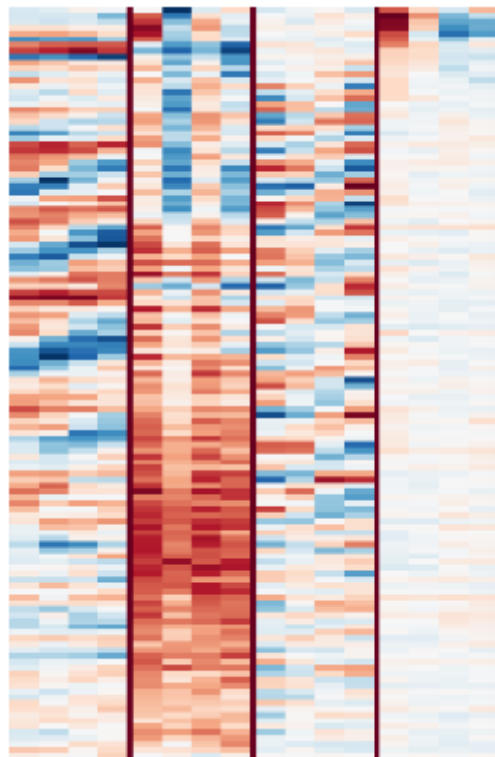
- 1. La Iniciación  
Los Nuevos Rebeldes 2:28
- 2. Go To Sleep You Little Baby  
Thula Mama 2:03
- 3. Mil Vidas  
Carlos Macías, Fernanda Castillo 4:27
- 4. Te pesara  
Los Canarios De Michoacan 3:11
- 5. Just To Worship  
James Fortune, Fiya 4:10
- 6. Let Me Live That Fantasy  
Femke, GMPresents & Jocelyn Scofield 3:11
- 7. Royals (Radio Edit)  
Femke, GMPresents & Jocelyn Scofield 3:11
- 8. Treacherous - Original Demo Rec...  
Taylor Swift 4:00



1. Martyr  
Harvest

low-level #253 (max): bass d...

- 1. Martyr  
Harvest 4:26
- 2. Dollar Dan\$en  
Troo.L.S. & Orgi-E 3:23
- 3. A Trip to Bulgaria  
Dr Peacock 3:37
- 4. Eyes On The Prize  
George & Jonathan 2:10
- 5. Miracle - Breakage's An Inferior...  
Hurts 5:06
- 6. Make Some Noise  
Endymion 3:05
- 7. Makin' It Blend  
Q-Tip, Words 3:55
- 8. Smack  
Vent, Crush 4:34



过滤器14，242，250 和 253的特写图。

- 过滤器 14 似乎探测出颤音歌声 ( vibrato singing )。
- 过滤器 242探测出某种响铃氛围 ( ringing ambience )。
- 过滤器 250 探测出人声大三度 ( vocal thirds )，即多个歌唱者同唱一首歌，但音符相隔大三度 ( 四个半音 )。
- 过滤器 253 探测出各种类型的低音鼓音。

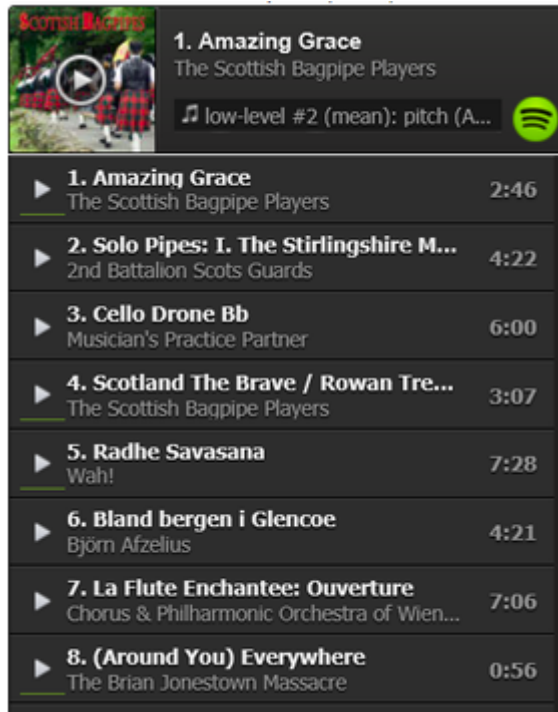
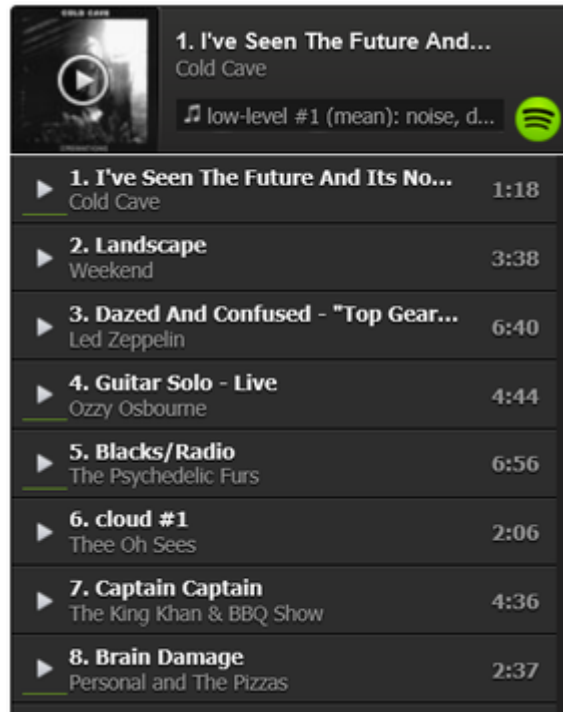
这些播放表中曲目的流派是很不同的，这表示它们主要是从音频信号的低级特性中检测出这些特征的。

### 低级特征播放表：平均激活

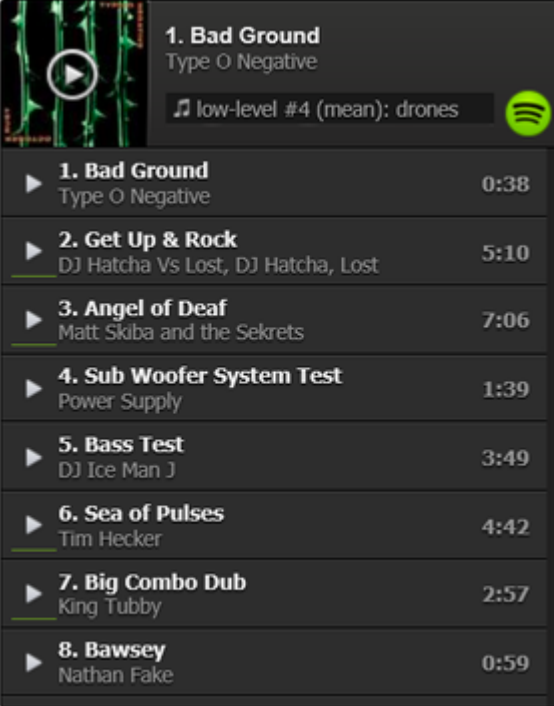
下面四个播放表是用稍微不同的方式获得的：首先对每个曲目计算时域特征的激活平均，然后找出它们中的最大值。这意味着在这些播放表中，涉及的过滤器在分析的30秒钟内一直有效（也就是，它不会只是一个‘峰值’）。这对于检测和声模式更加有用。

过滤器 1：噪音，失真

过滤器 2：音高（A，Bb）



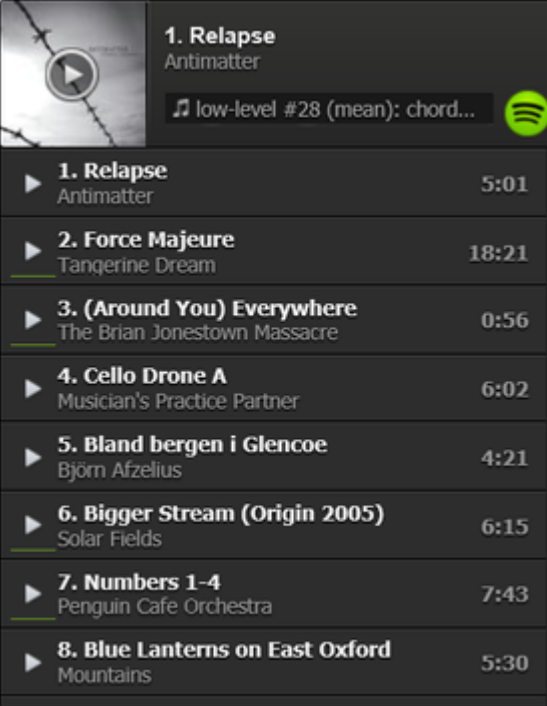
过滤器 4：嗡嗡声 过滤器 28：和声（A, Am）



**1. Bad Ground**  
Type O Negative

low-level #4 (mean): drones

▶ <b>1. Bad Ground</b> Type O Negative	0:38
▶ <b>2. Get Up &amp; Rock</b> DJ Hatcha Vs Lost, DJ Hatcha, Lost	5:10
▶ <b>3. Angel of Deaf</b> Matt Skiba and the Sekrets	7:06
▶ <b>4. Sub Woofer System Test</b> Power Supply	1:39
▶ <b>5. Bass Test</b> DJ Ice Man J	3:49
▶ <b>6. Sea of Pulses</b> Tim Hecker	4:42
▶ <b>7. Big Combo Dub</b> King Tubby	2:57
▶ <b>8. Bawsey</b> Nathan Fake	0:59

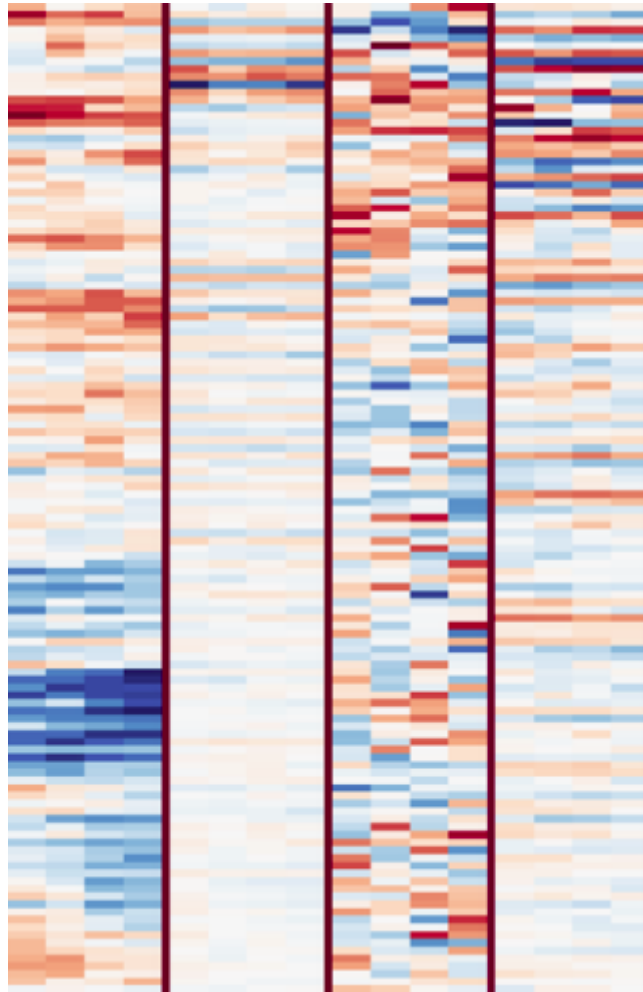


**1. Relapse**  
Antimatter

low-level #28 (mean): chord...

▶ <b>1. Relapse</b> Antimatter	5:01
▶ <b>2. Force Majeure</b> Tangerine Dream	18:21
▶ <b>3. (Around You) Everywhere</b> The Brian Jonestown Massacre	0:56
▶ <b>4. Cello Drone A</b> Musician's Practice Partner	6:02
▶ <b>5. Bland bergen i Glencoe</b> Björn Afzelius	4:21
▶ <b>6. Bigger Stream (Origin 2005)</b> Solar Fields	6:15
▶ <b>7. Numbers 1-4</b> Penguin Cafe Orchestra	7:43
▶ <b>8. Blue Lanterns on East Oxford</b> Mountains	5:30





过滤器1, 2, 4和28的特写图。

- 过滤器1 检测噪音和（吉他）失真音。
- 过滤器2好像检测到一个特殊音高：一个低音Bb。它有时也检测出A声（低半音），因为梅尔声谱的频率分辨率还没有高到足以区分这两个音。
- 过滤器 4 检测各种低音嗡嗡声（drones）。
- 过滤器 28 检测A和弦。看起来它既检测小音阶也检测大音阶版本，所以它可能只检测音高A和E（五度音程）。

我觉得很有趣的是，该网络学会了检测特别的音高和和声。我以前还以为歌曲中准确的音高和和声的出现，不会影响听众的喜爱程度。至于为什么会这样我有点推测：

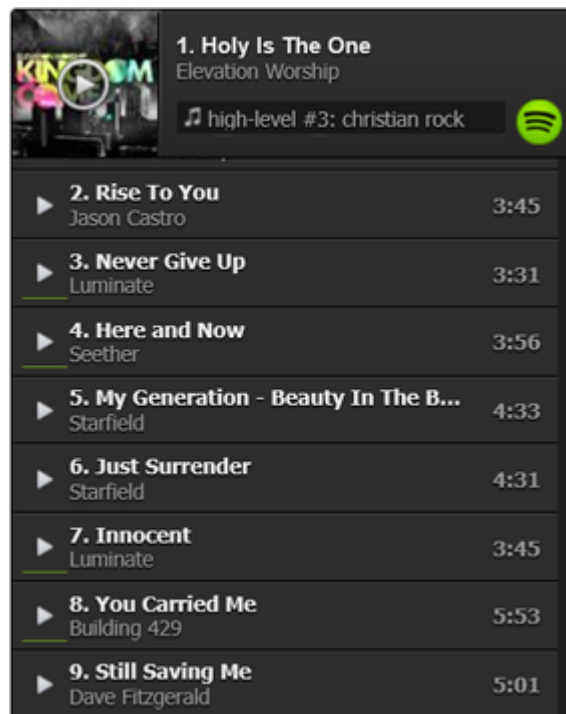
- 用不同的谐音训练了各种过滤器以后，这个网络其实仅仅学习了检测调和性（harmonic）。然后在更高层级上它们被池化到一起，以检测各种音高的调和性。
- 网络学会了在某些流派的音乐中，某个和弦以及和弦进行（chord progressions）比其他和弦更加常用。

我还没有验证上述两点中的任何一点，但看起来后者对于网络有更大的挑战，因此我认为前者的可能性更大。

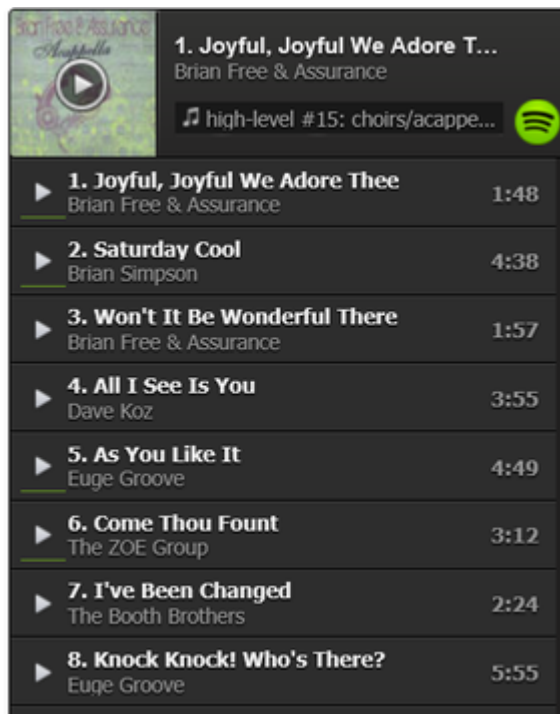
## 高级特征播放表

网络的每一层都从下一层取得特征表现，然后从中提取一组高级特征。在网络最上面的完全连接层，即最靠近输出层的前面一层，学习过的过滤器对某些副主题是非常有选择性的。显而易见，在声谱级可视化这些过滤器的检测结果不是一件简单的事情。下面是六个测试集歌曲的播放表，这些歌曲最大激活了其中的几个高级过滤器。

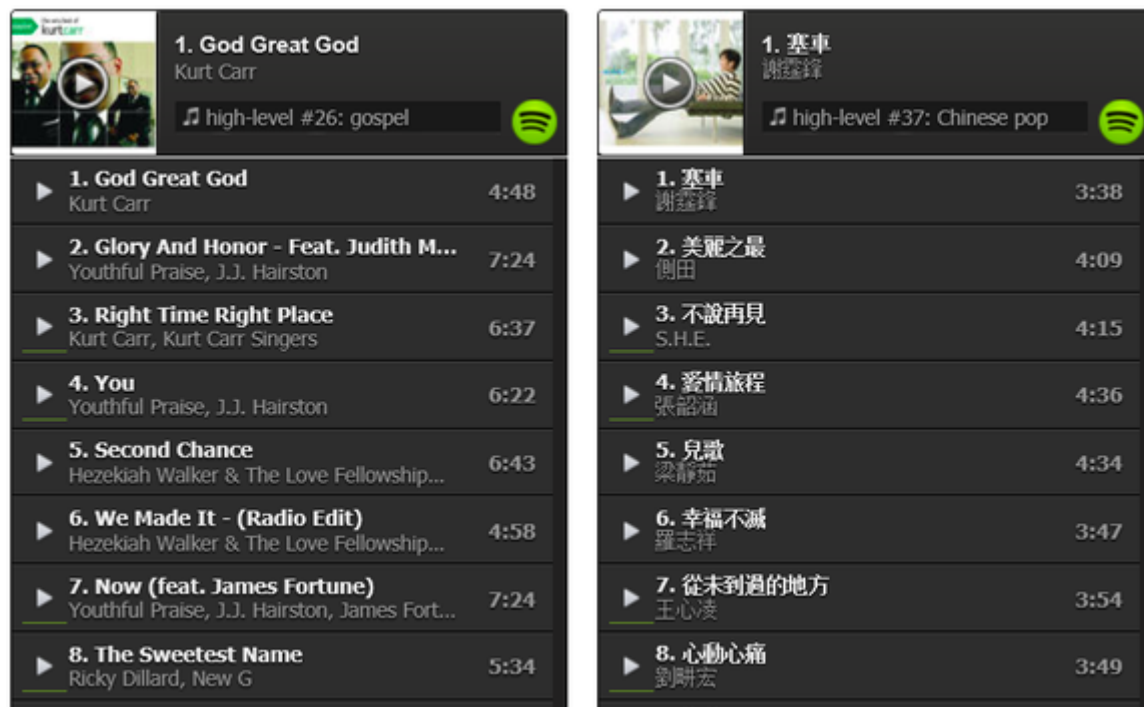
过滤器 3：基督教摇滚（christian rock）    过滤器 15：合唱/无伴奏合唱+时尚爵士



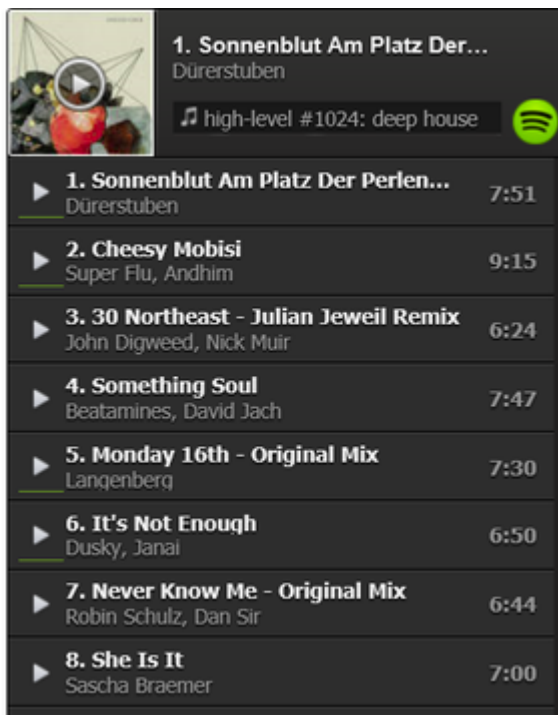
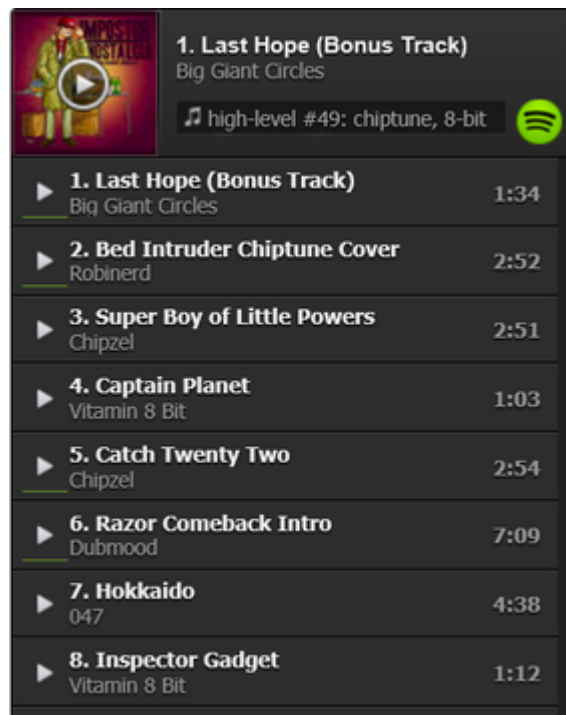
过滤器 26：福音歌



过滤器 37：华语流行



过滤器 49：合成电子乐，8比特过滤器 1024：deep house音乐



很明显，其中每个过滤器都识别一种特定的类型。有趣的是有些过滤器，比如第15号，似乎是多模式的（multimodal）：它强烈地被两种或更多种风格的音乐激活，而那些音乐经常是完全不相关的。大概这些过滤器在结合了所有其它过滤器的激活以后，消除了输出歧义。

过滤器37很有趣，因为它似乎可以识别中文语言。这不是完全不可能的，因为中文的语音库与其他语言相比是很独特的。有其他几个过滤器似乎学习了特定的语言：比如有一个能检测出西班牙语的rap音乐。也有可能性是华语流行音乐存在其它可区分的特性，而那个模型就是检测到了此特性。

我花了一些时间对开始的约50个过滤器作了详细的分析。我想出的其他几个过滤器种类还有：酒廊音乐，雷鬼乐（reggae），暗潮（darkwave），乡村音乐，金属核（metalcore），莎莎舞乐（salsa），荷兰和德国的狂欢节音乐，儿童歌曲，人声电音（vocal trance），朋克（punk），土耳其流行乐，还有我最喜爱的‘exclusively Armin van Buuren’。很明显由于他有那么多的曲目，所以他才有了自己的过滤器。

经过Alex Krizhevsky ImageNet网络学习的过滤器，已经被重复用在各种计算机视觉任务中，并获得了极大的成功。基于这些过滤器的多样性和不变性特性（invariance properties），这些学习音频信号的过滤器，除了能预测隐藏因素之外，也可以用于其他音乐信息检索任务。

## 基于相似度的播放列表

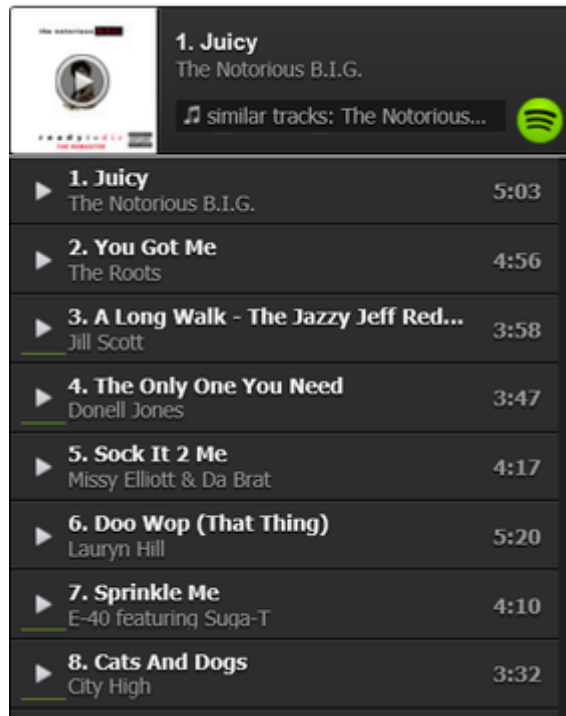
预测的隐因素向量也可以用来查找听起来相似的歌曲。下面是这样产生的几个播放表：首先预测给定歌曲的因素向量，然后从测试集中查找出这样的歌曲，其预测因素向量的余弦距离，靠近给定的歌曲。这样播放列表中的第一个曲目永远是查询曲目本身。

The Notorious B.I.G. – Juicy

Cloudkicker - He would be riding on

（嘻哈舞曲）

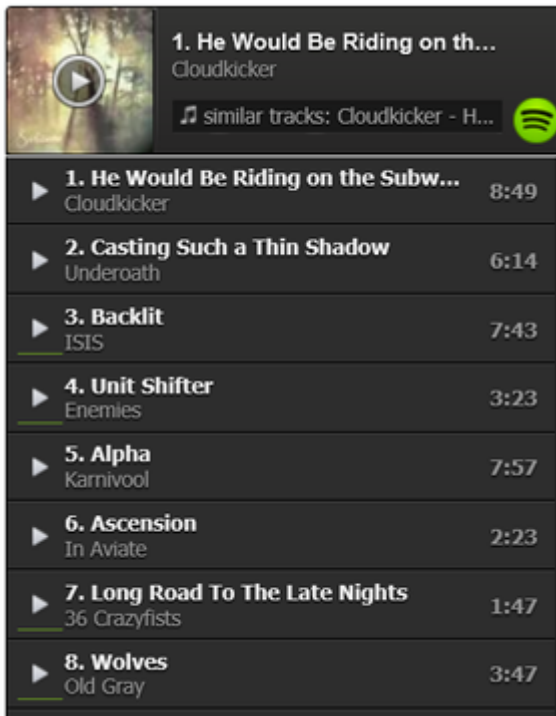
the subway...后现代摇滚，前卫金属）



1. Juicy  
The Notorious B.I.G.

similar tracks: The Notorious...

1. Juicy	5:03
2. You Got Me	4:56
3. A Long Walk - The Jazzy Jeff Red...	3:58
4. The Only One You Need	3:47
5. Sock It 2 Me	4:17
6. Doo Wop (That Thing)	5:20
7. Sprinkle Me	4:10
8. Cats And Dogs	3:32



1. He Would Be Riding on th...  
Cloudkicker

similar tracks: Cloudkicker - H...

1. He Would Be Riding on the Subw...	8:49
2. Casting Such a Thin Shadow	6:14
3. Backlit	7:43
4. Unit Shifter	3:23
5. Alpha	7:57
6. Ascension	2:23
7. Long Road To The Late Nights	1:47
8. Wolves	3:47



Architects - Numbers Count For Nothing

Neophyte - Army of Hardcore

( 金属核 , 硬核 )

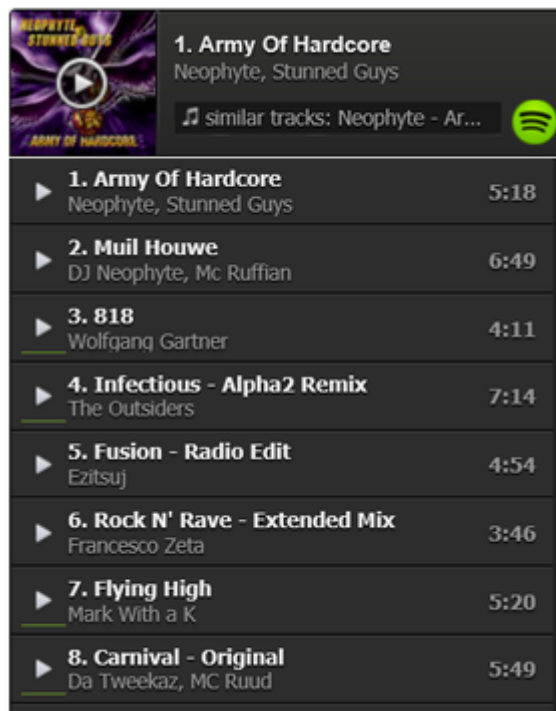
( 硬式电子音乐, 盖巴舞曲 )




1. Numbers Count For Nothing  
Architects

similar tracks: Architects - Nu... 

▶ 1. Numbers Count For Nothing	3:50
▶ 2. Die Knowing	2:13
▶ 3. Persevere and Overcome	2:37
▶ 4. Broken Records	1:42
▶ 5. Rumours	2:52
▶ 6. I've Been Trying To Leave	2:12
▶ 7. Lessons	2:35
▶ 8. A Temperamental Widower	2:47

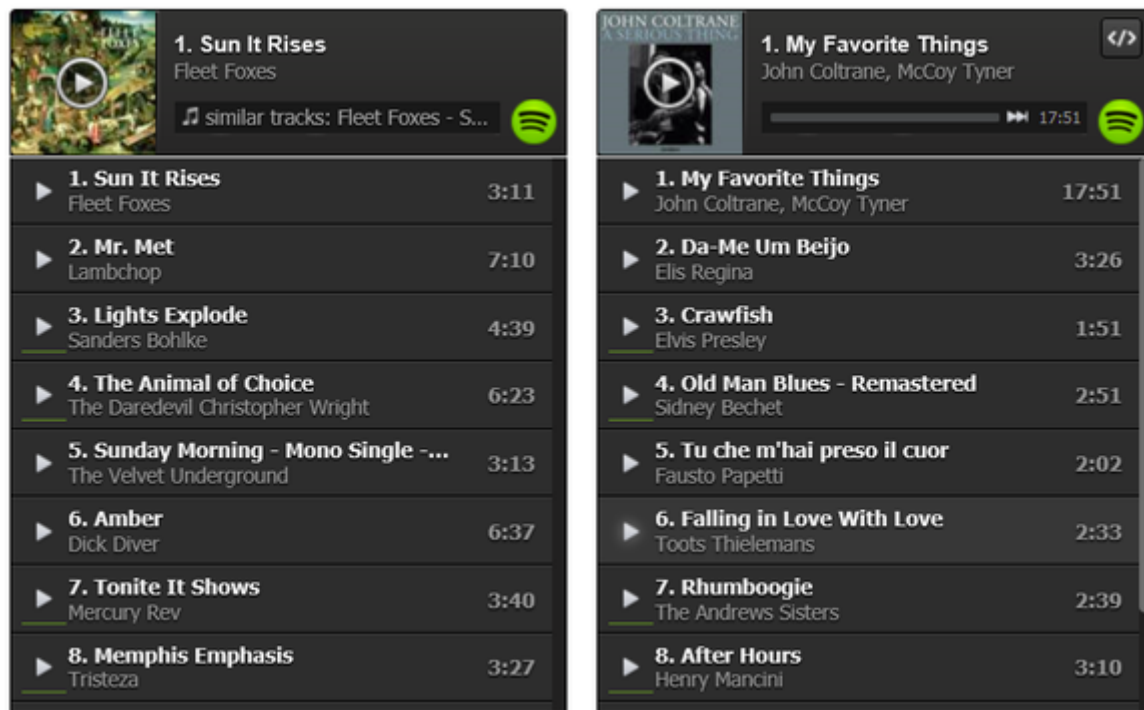


1. Army Of Hardcore  
Neophyte, Stunned Guys

similar tracks: Neophyte - Ar... 

▶ 1. Army Of Hardcore	5:18
▶ 2. Muil Houwe	6:49
▶ 3. 818	4:11
▶ 4. Infectious - Alpha2 Remix	7:14
▶ 5. Fusion - Radio Edit	4:54
▶ 6. Rock N' Rave - Extended Mix	3:46
▶ 7. Flying High	5:20
▶ 8. Carnival - Original	5:49

Fleet Foxes - Sun It Rises ( 独立民谣 ) John Coltrane - My Favorite Things ( 爵士乐 )



绝大多数相似的曲目都是很合适向查询该曲目的歌迷推荐的。当然这些列表并不是很完美的，但是考虑到仅仅依靠音频信号，就获得了这样的结果，应该算是相当不错了。有一个有错误的例子出现在 John Coltrane 的 'My Favorite Things' 播放表中，此播放表的一个不同点是包含几个奇异值（outliers），最明显的地方是在 Elvis Presley 的 'Crawfish' 中。其原因可能是在被分析的音频信号段（从 8:40 到 9:10）包含一段疯狂的萨克斯独奏。如果分析整首歌曲，可能会有比较好的结果。

## 它们有什么用？

Spotify 已经在其推荐管道中使用了一大堆不同的信息源和算法，因此我的工作最明显的应用就是添加成另外一个信号源。当然它也可以用来过滤由其它算法推荐的异常结果。我在前面已经指出，协同过滤算法趋向于在推荐中包含引曲、终曲、翻唱曲和混音曲。这些可以通过基于音频的方法有效地过滤。

我在这项工作中的一个主要目标是可以用它推荐新的，以及尚未流行的音乐。我希望这样提供帮助，那些不太知名和未来的乐队，通过允许Spotify向适合的听众推荐他们的音乐，得到公平的竞争环境。（宣传未来的乐队碰巧也是我一个非营利网站[got-djent.com](http://got-djent.com)的主要目标。）

希望不久它们的部分功能就能开始 [A/B 测试](#)，于是我们可以知道这个基于音频的推荐，能不能在实践中表现非凡。这是我非常兴奋的一件事情，因为它不是在学术界轻松做到的。

## 今后的工作

Spotify收集到的另一种用户反馈形式是用户对电台播放曲目的向上拇指和向下拇指。这类信息对于确定哪些曲目是类似的十分有用。不幸的是其中的噪音也很大。我目前正在尝试在‘[排序学习](#)’（[learning to rank](#)）设置中使用这些数据。我也在实验各种距离度量学习方案，比如[DrLIM](#)。如果有任何很酷的结果我可能会写一篇新的文章。

## 结论

本文中我概述了在Spotify机器学习实习中到目前为止所做的工作。我解释了使用卷积网络做基于音频的音乐推荐的方法，并提出了有关该卷积网络的实际学习效果的心得。有关这个方法更详细的内容，请参考由我和 Aäron van den Oord在NIPS 2013合写的论文‘[基于内容的深度音乐推荐](#)’（[Deep content-based music recommendation](#)）。

如果你对深度学习，特征学习以及它在音乐中的应用有兴趣，可以到我网站的[research](#)中看看，了解一下我在这个领域做过的其它工作。如果你对在音乐推荐中Spotify的方法有兴趣，参考一下[Slideshare](#)和[Erik Bernhardsson](#)在其博客中的介绍。

Spotify是一个很酷的工作场所。他们对其使用的方法很开放（同时允许我写这篇博客文章），这在工业界并不是很常见的。

如果你对本文有任何问题或者反馈意见，请留下你的评论！

**原文链接：**[Recommending music on Spotify with deep learning](#)（文/彭根禄 责编/钱曙光）

本文为CSDN编译整理，未经允许不得转载，如需转载请联系market#csdn.net(#换成@)

顶  
16

踩  
0

推荐阅读相关主题： [算法](#) [音乐](#) [计算机视觉](#) [可视化](#) [西班牙](#) [互联网](#)

### 相关文章

### 最新报道

京东DNN Lab首席科学家：用深度学习搞定80%的客...

APSys 2014，AMD发表基于异构体系的DNN实现与...

Caffe 深度学习框架上手教程

深度学习 vs. 概率图模型 vs. 逻辑学

深度学习 vs. 大数据：神经网络权值的版权属于谁？

KDnuggets热门深度学习工具排行：Pylearn2 居首，...

## 已有8条评论

还可以再输入500个字



有什么感想，你也来说说吧！

您还没有登录! 请 [登录](#) 或 [注册](#)

发表评论

### 最新评论

### 最热评论



请输入2-20字的昵称哦 2015-11-16 19:09

你们看懂了吗

回复



tengshan2008 2015-11-11 16:00

最近在做我的研究生开题，想做基于协同过滤的推荐，想不到好的创新点，想做深度学习的内容，这篇帖子给了我一定的启发，可是怕自己驾驭不了。

回复



yhtang15 2015-02-21 17:55

那些细节基本源自博主发表在NIPS的论文，必须看不懂，不然NIPS就水了 $O(n_n)O\sim$

回复



clanaid 2015-02-21 00:52

实在深奥，不是很懂

回复



EJK 2015-02-20 21:01

Wow, 很棒的东西，mark一下，以便好好学习学习！

回复



QIDoors 2015-02-20 19:33

不明觉厉

回复



z707582070 2015-02-20 18:53

我看不懂，你们能看懂吗？

[回复](#)

mrplplpl 2015-02-20 09:45

mark一下, 学习了

[回复](#)[共1页](#)[首页](#)[上一页](#)

1

[下一页](#)[末页](#)

### 请您注意

- 自觉遵守：爱国、守法、自律、真实、文明的原则
- 尊重网上道德，遵守《全国人大常委会关于维护互联网安全的决定》及中华人民共和国其他各项有关法律法规
- 严禁发表危害国家安全，破坏民族团结、国家宗教政策和社会稳定，含侮辱、诽谤、教唆、淫秽等内容的作品
- 承担一切因您的行为而直接或间接导致的民事或刑事法律责任
- 您在CSDN新闻评论发表的作品，CSDN有权在网站内保留、转载、引用或者删除
- 参与本评论即表明您已经阅读并接受上述条款

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)[网站客服](#)[杂志客服](#)[微博客服](#)[webmaster@csdn.net](mailto:webmaster@csdn.net)

400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved

