北京郵電大學

雏雁计划项目

结题报告

项目编号：Y1700016　项目级别：A

项目类别：目标导向类□　自主探索类√
校企合作类□　导师科研类□

项目中文名称：机器学习编曲

项目名称（英文）：Machine Learning Based Music Arrangement

项目负责人：吴雨松

联系电话：13911356479

E—Mail:wuyusongwys@gmail.com

指导教师：杨辉华

E—Mail:yhh@bupt.edu.cn

立项时间：2017.5

结题时间：2018.5

2018 年　5 月　13　日

机器学习编曲

摘　要

随着机器学习与人工智能技术的发展，基于机器学习的模型得以有能力代替人类进行复杂的工作，如人脸识别，围棋博弈等。音乐，作为反映人类现实生活情感的一种艺术，其创作也是有一定规则与规律，并可以被机器学习模型所学习并模仿。本项目就基于机器学习模型的计算机音乐创作与编曲进行了研究与探索，制作完成了两个音乐生成模型。这两个音乐生成模型分别为基于长短期记忆网络（Long Short-Term Memory, LSTM）的单声部 -单音生成模型与双声部 -和弦生成模型。本项目就两个模型分别在混合数据集与巴赫风格数据集上进行训练，并进行了音乐生成。

关键词　机器学习　深度学习　人工智能　作曲

目　录

第一章　引言

1.1　项目研究选题背景

在音乐作曲中，作曲家在谱写旋律与音乐格调是需要付诸极高的创造力与灵感。但通常在进行和声编写与编曲过程中，其乐理规则相对固定死板，并很大程度上依据经验完成，不一定需要极强的创造力。现今的编曲行业大多是由大师写出一段旋律之后交由编曲师完成的，编曲师一般会遵循一定的乐理规律，并运用经验化的编曲方法来谱写，然而乐理规律复杂繁多，甚至有很多人类还没有完全严谨的总结，经验化的工作枯燥而劳累。而机器学习这一方法恰恰适用于规律人为总结难度大、操作机械化的数据处理。本项目针对这一问题，意图完成一个自动编曲软件，可以学习人们未曾发现的编曲规律并根据旋律生成伴奏，减轻编曲师的负担。

截至立项时，业界并无成熟的机器学习音乐编曲项目。对于使用传方法辅助编曲的软件，如 Wolfram 的曲调生成器和编曲软件 Band in a box，它们运用乐理上人们总结的编曲规律来生成旋律，音乐效果不错，但由于人为总结并编程的规则简单有限，其音乐千篇一律，也局限于流行音乐的很少几个声部。在机器学习与音乐创作结合的方面，相关研究方向集中在自动音乐作曲。在音乐作曲方面，1991 年 David Cope 曾做过首个相关项目“机器谱曲”，但其算法只是根据特征值将音乐库中的音乐切成碎片打乱重排；对于楚航等人的研究，其通过机器学习分析图片，输入文字唱出歌曲，但效果欠佳，音乐有不和谐之处；谷歌曾组织人工智能音乐研究项目 Magenta，其完成的项目声部单一，并旨在生成旋律，与本项目的目的不同。

总的来说，现有的基于机器学习的研究效果欠佳，同时并无针对编曲的研究。基于这个背景，本项目将参考前人做过的相关探索研究，并重点参考 Google 开源项目 Magenta。

1.2 项目研究的意义

本项目旨在制作一种基于机器学习中的长短期记忆网络（LSTM）编曲软件。其目的是让机器学会古典交响乐的伴奏方式，达到用户输入一段旋律，软件可以输出一段带有伴奏的旋律续写的效果。

本项目选用 MIDI 格式的乐曲库，实现了长短期记忆网络的 MIDI 接口，搭建和修改神经网络结构，训练出了一系列表示曲库中音乐旋律和伴奏之间的关系的参数模型。

对于无专业音乐知识的业余音乐爱好者，本软件帮助其将旋律谱成完整的曲目，便于演奏与编排。对于专业的音乐工作者，本软件可作为一个辅助作曲的工具，令其更方便地完成配器工作。

1.3 项目研究的创新点

本项目的创新点在于采用大数据高性能机器学习算法设计、研制无成熟的同类机器学习多声部编曲系统等。

第二章　乐理及数据表示

2.1　乐理

十二平均律：十二平均律，又称“十二等程律”，是一种音乐定律方法，将一个纯八度平均分成十二等份，每等分称为半音。因此，一个纯八度由 12 个音组成，它们分别为 $C, C\sharp/D\flat, D, D\sharp/E\flat, E, F, F\sharp/G\flat, G, G\sharp/A\flat, A, A\sharp/B\flat, B$。因此，对于一个八度中的 12 个音，每个半音之间为相邻关系。若将一个八度中十二个音按 1-12 编号，我们可以认为每个相邻半音之间的编号大小相差为 1。

音符时值：音符时值，也称为音符值或音值，在乐谱中用来表达各音符之间的相对持续时间。在音乐中，音符的时值按大小分为完全音符、二分音符、四分音符、八分音符、十六分音符、三十二分音符等。其时长关系为：一个完全音符等于两个二分音符；等于四个四分音符，八个八分音符；十六个十六分音符，三十二个三十二分音符。上述时长关系只是音符时值的比例，在演奏中，音符演奏的具体时间还由当前的速度决定。

2.2　数据表示

本项目中使用的数据表示方法为“钢琴卷帘（Piano Roll）”法。具体来讲，对于在一段时间演奏的音乐，对其按相对频率采样。对于每个采样点，生成采样点对应的时间点的音高数据。对于采样频率，本项目在时间上按每个四分音符 16 个均匀采样点的采样频率进行对音高的采样，即将六十四分音符作为最小时间单位。对于每个时间点上的音高数据，本项目使用独热表示法：该方法首先需要统计表示范围内所有音高的数量 N，然后给这 N 个音高分别编号为 $1, 2, \ldots, N$，最终使用一个仅第 k 维为 1 的 N 维向量来表示编号为 k 的词。例如，在音符空间 $\Omega = \{C_4, D_4, E_4\}$ 中，“C_4”的独热编码为 $[1, 0, 0]$，“D_4”的独热编码为 $[0, 1, 0]$，“E_4”的独热编码为 $[0, 0, 1]$。对于每个时间点的音高数据，本项目用一个 128 维的独热向量储存。其中，对于音高中央 C“C_4”，在独热向量中的维度为第 60 维。因此，本项目可以表示的音高空间范围为 C_0 到 G_8。

第三章　预处理及数据集

3.1　预处理

数据预处理是为了剔除不符合要求的数据种类，并将音乐由连续转化为离散的向量形式，便于后续的处理。本项目中预处理程序的运行流程为，对指定路径内的所有数据文件，排除无效文件格式、过短曲目及含有无关乐器的文件后，对每个符合要求MIDI 文件转换成矩阵形式储存。对于 MIDI 格式的文件，首先将其按每个四分音符 16 个均匀采样点的采样频率进行对音高的采样，然后将每个采样点对应的音高以一个维数为 128 的独热码向量储存，这个向量的每一维表示一种音高，采样点对应音高所在维的值为 1，其余维为 0。

3.2　数据集

本项目经过收集公开 MIDI 数据集及自主建立并整理 MIDI 数据集，共收集近一百五十万个 MIDI 文件。对这些 MIDI 文件进行上述预处理，本项目建立了两个数据集。一个是人为挑选的巴赫风格的巴赫小型数据集，由 412 首巴赫创作的音乐，共约 2000 万个数据点组成，风格鲜明、数据量小，用于进行初期的验证性试验以及训练和测试单一风格的音乐。另一个是混合了流行乐、电子乐、古典乐等多风格音乐的混合大型数据集，在对公开数据集进行预处理后随机挑选。由 2000 多首音乐，共约 4000 万个数据点组成，风格混杂、相对数据量大，用于训练和测试混合风格的音乐。由于过大的数据集不利于神经网络快速有效的学习与模型的调整迭代，我们最终选择了规模相对大型的数据集大小。

我们将数据集分割为训练集与测试集，用于评估神经网络的训练效果，两者比例为 9:1。

第四章　时间序列模型

4.1　循环神经网络

循环神经网络（Recurrent Neural Networks，RNN）是一种用于预测时间序列的神经网络。它的核心由一个深度神经网络组成，输入上一个时间点向前传递的一个隐藏状态 h_{t-1} 与当前时间点的数据 X_t，并输出当前时间点的隐藏状态及 h_t。这种结构使得它可以将当前时间点之前的输入数据的一部分信息传递到当前时间点的处理，使其拥有时间上的“记忆”能力。网络前向传播时，其循环状态与在时间上的展开如图 4-1 所示。

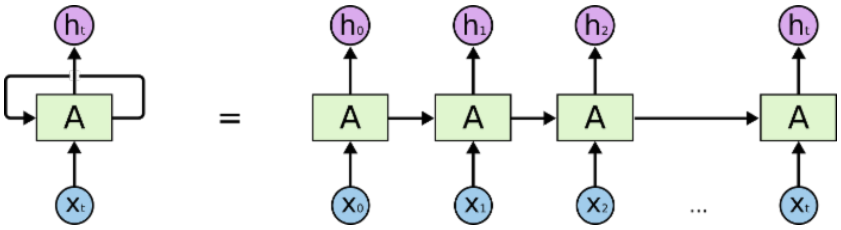


图 4-1 循环神经网络

其前向传播的公式为

$$h_t = g(W[h_{t-1}, X_t] + b) \qquad \text{式（4-1）}$$

其中 W 为隐藏层参数，b 为偏置层参数，$g()$ 为激活函数。

4.2　长短期记忆网络

长短期记忆网络（Long Short-Term Memory, LSTM）是一种改进型循环神经网络，其在一定程度上解决了循环神经网络在训练时的梯度消散及梯度爆炸问题。因此，这种改进型模型可以在很长的时间序列上进行梯度反向传播，在一定程度上解决了递归神经网络的长期依赖问题，可以很好的预测间隔长时间的时间序列。长短期记忆网络的结构简图如图 4-2 所示。长短期记忆网络内部由三个“门”组成，分别为“遗忘门“、”更新门（也称输入门）“、”输出门 “，分别由三个神经网络组成。与递归神经网络类似，长短期记忆网络在每个时间点的正向传播时输入上一时间节点的神经元状态 C_{t-1} 及隐藏状态 h_{t-1}，并输入当前时间点的数据 X_t，并输出当前时间点的两个状态 C_t 及 h_t。长短期记忆网络的正向传播过程的逐步推导如下：

首先进行遗忘门的运算。h_{t-1} 及 X_t 输入神经网络 f 并经过 $sigmoid$ 激活函数进行运算，运算结果输出一个 0 到 1 之间的数值 f_t。对于这一步，我们可以认为遗忘门所包含的神经网络筛选了前一个时间点隐藏状态 h_{t-1} 中信息的取舍。其中，数字越大则保留的越多。

其示意图与运算公式如图 4-3 所示。

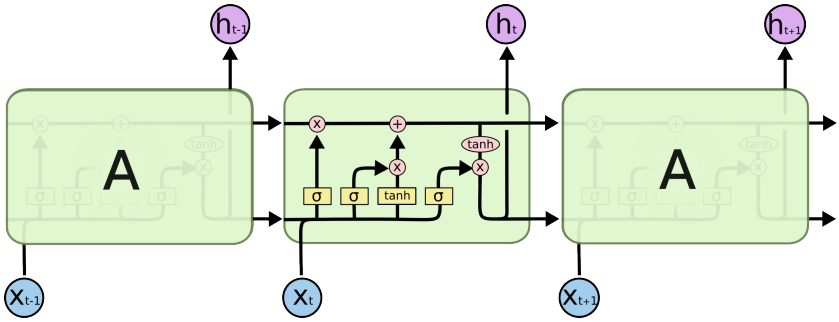
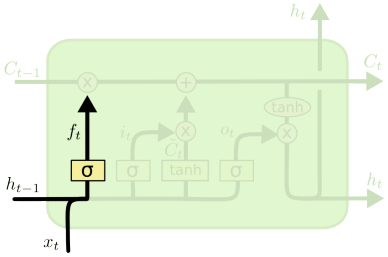


图 4-2 长短期记忆网络结构



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

图 4-3 第一步

其中，$sigmoid$ 激活函数的运算公式如下：

$$sigmoid(X) = \frac{1}{(1 + e^{-X})} \qquad \text{式（4-2）}$$

第二步与第三步进行更新门的运算。首先，与遗忘门类似，将 h_{t-1} 及 X_t 输入神经网络 i，并经过 $sigmoid$ 激活函数，输出一个 0 到 1 的数值 i_t。在这之后，将 h_{t-1} 及 X_t 输入神经网络 C，并使用 tanh 激活函数进行输出一个相同维度的矩阵 $\tilde{C}_t$，此举主要是用于对 h_{t-1} 进行一个非线性变换。

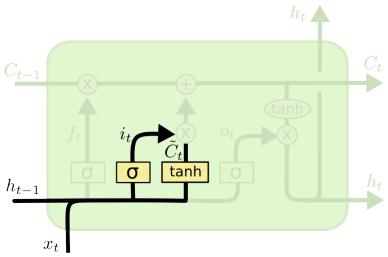
其中，tanh 激活函数的运算公式如下：

$$\tanh(X) = \frac{e^{X} - e^{-X}}{e^{X} + e^{-X}} \qquad \text{式（4-3）}$$

在这之后，使用遗忘门输出的权重 f_t 乘以 C_{t-1}，并加上更新门计算的权重 i_t 乘以非线性变换结果 $\tilde{C}_t$，得到当前时间点的神经元状态 C_t。此举解释为是将当前时间点的输入信息更新到当前时间点的神经元状态 C_t 中。其示意图与运算公式如图 4-4 与 4-5 所示。

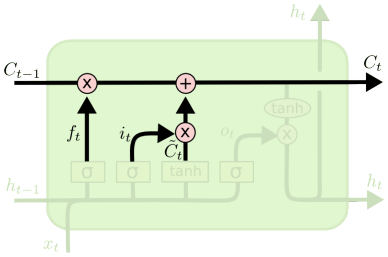
最后，将 h_{t-1} 及 X_t 经过输出门的神经网络 o，经过 $sigmoid$□□ 输出一个 0 到 1 的数值 o_t，将 o_t 与上一步得出的当前时间点的神经元状态 C_t，得到当前时间点的隐藏状态 h_t。

通常，长短期记忆网络的输出就是 h_t 或 h_t 经过全连接层的分类器。因此，此举可以被认为是确定输出的内容 h_t 中包含多少当前时间点的神经元状态 C_t 的内容。其示意图与运算公式如图 4-6 所示。



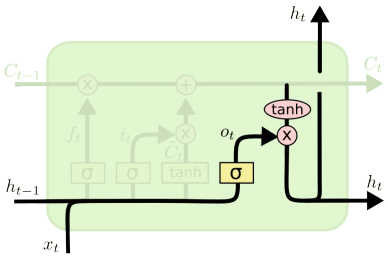
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

图 4-4 第二步



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

图 4-5 第三步



$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

图 4-6 第四步

多对一（many-to-one）结构：多对一结构根据是长短期神经网络输入输出序列的时间点数量而划分的多种结构，如一对一（one-to-one）、一对多（one-to-many）、多对多（many-to-many）中的一种。它们的结构示意图如图 4-7 所示。

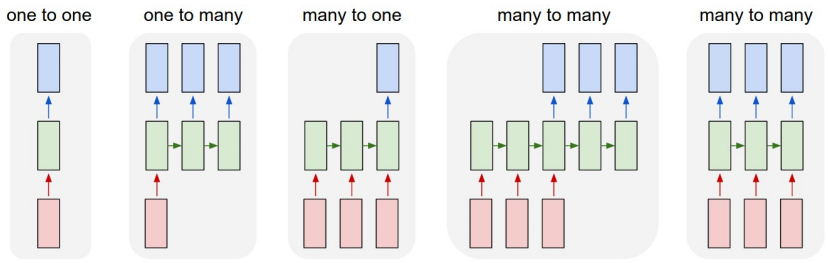
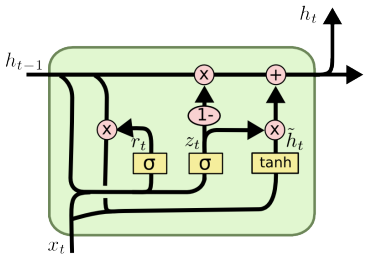


图 4-7 输入输出结构

具体来讲，多对一结构的长短期神经网络的前向传播结构为：每一个时间点输出的隐藏状态只在最后一个时间点的神经元经过一个全连接层分类器进行输出，而在这之前的时间点的神经元的输出只传递到下一个神经元。

4.3 GRU 单元

GRU（Gated Recurrent Unit）是一种长短期记忆网络的变体。其主要的改进在于组合遗忘们和输入门为一个“更新门”，合并了神经元状态和隐层状态。因此，对于 GRU 单元的长短期记忆网络，其输入只有上一个时间节点的一个隐藏状态输入 h_{t-1} 与当前时间点的数据 X_t。这使得前向传播和反向传播的训练更加快速和高效。本项目中模型使用的长短期记忆网络皆为带有 GRU 单元的长短期记忆网络。它的结构简图及前项传播计算公式如图 4-8 所示。



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

图 4-8 GRU 结构

第五章　本项目模型介绍

5.1　模型

针对不同的复杂程度，本项目构建了两个神经网络模型。

5.1.1　单声部 -单音输出模型

第一个模型为单声部 -单音输出模型，该模型完成的任务是输入一段单音旋律，输出该旋律的续写。

它是由双层长短期记忆网络组成的时间序列预测模型，有 128 个隐藏层，输入为 256 个连续采样点长度，即四个小节长度的音高序列，输出为下一个采样点的音高。本模型采用上述多对一结构，在输出时通过一个带有 softmax 激活函数的 128 维全连接层来获得每个音高上的概率预测值。

其中，对于一个 k 维的矩阵 X，$softmax$ 激活函数的运算公式如下：

$$softmax(X) = \frac{e^{X_j}}{\sum_{k=1}^{k} e^{X_k}} \quad for\ j = 1, \ldots, k \qquad \text{式（5-1）}$$

对于经过 $softmax$ 函数计算的向量，其每一维度的值在 0 到 1 之间，所有维度值的总和为 1。我们可以认为，经过上述模型，我们经过模型计算出的是对于一个长度为 t 时间序列 X，其第 t+1 个时间点 X_{t+1} 在 X 的条件概率:

$$P(X) = P(X_{t+1}|X_t, \ldots, X_1) \qquad \text{式（5-2）}$$

在训练时，我们采用交叉熵损失函数，计算模型预测的下一采样点的音高矩阵的概率与数据集中下一采样点的音高的交叉熵损失值，作为损失大小，进行反向传播，从而更新模型的参数。对于交叉熵损失函数，通过神经网络输出值 $\hat{y}_i$ 与标签 y_i 计算损失值 E_i 的公式为：

$$E_i = -y_i \log \hat{y}_i - (1 - y_i) \log \hat{y}_i \qquad \text{式（5-3）}$$

对于第一个时间节点的隐藏状态输入，本模型使用全零向量进行隐藏层状态的初始输入。

5.1.2　多声部 -多音输出模型

第二个模型为基于开源音乐生成模型的多声部 -多音输出模型。它由旋律输层出与和弦输出层两个部分组成。旋律输出部分与上述单声部 -单音输出模型相似，将输入的 256 个连续采样点长度的音高序列进行预测，输出下一个时间的预测值。和弦输出部分的输入为 256 个连续采样点长度的旋律，输出则为为旋律搭配的和弦。和弦的数据表示方法为独热向量，维度数为 128。在训练时，我们分析当前时间点输入旋律附近一段时间的音高组成，作为训练时旋律的标签值。

旋律输层出与和弦输出层皆采用 64 个隐藏的双层长短期记忆网络，损失函数皆为交叉熵损失函数。因为需要同时训练两个神经网络，本模型隐藏层数量较单声部 -单音输出模型有所缩减。同样，对于第一个时间节点的隐藏状态输入，本模型的两个层皆使用全零向量进行隐藏层状态的初始输入。

第六章　训练和输出

6.1　训练

6.1.1　单声部 -单音输出模型

本模型在巴赫小型数据集上进行了训练。本模型在训练时的参数如下：学习速率为 0.00001，并使用小批量梯度下降法（Mini Batch Gradient Descent）进行训练以及参数的更新，其中批尺寸为 400，即一次迭代中使用 400 组数据进行训练。本模型在巴赫小型数据集上训练了单声部 -单音输出模型，迭代 114000 次后，训练集预测准确率约为 95%，测试集预测准确率约为 80%~85%。

6.1.2　多声部 -多音输出模型

本模型分别在巴赫小型数据集与大型混合数据集上进行了训练。本模型同样使用小批量梯度下降法进行训练以及参数的更新，其中批尺寸为 80。在使用巴赫小型数据集训练时，本模型训练时的参数如下：学习速率为 0.001，迭代 20000 次后，训练集预测准确率约为 78%，测试集预测准确率约为 76%。在使用混合大型数据集训练时，本模型训练时的参数如下：学习速率为 0.001，迭代 20000 次后，训练集预测准确率约为 70%，测试集预测准确率约为 65%。

6.2　输出

按概率随机预测：本项目对于两个模型的神经网络的预测皆采用随机预测的方式，即，对每一个音符，按模型 softmax 层计算出的每一个音符的预测概率进行随机。此举一方面避免了模型的输出陷入循环或者进行重复预测，另一方面也使得输出的音乐更富有变化。

循环输出：本项目的两个模型皆输出采用循环输出的方式，将输出作为下一次循环的输入进行输出。具体步骤的伪代码形式如下。

算法 1 模型的循环输出

输入： $[X_{t-256},\ldots,X_t]$
输出： $[Y_{t+1},\ldots,Y_{t+n}]$

for $i = 1 \to n$ **do**
　　$Y_{t+1} \leftarrow$ 正向传播 ($[X_{t-256},\ldots,X_t]$)
　　将 Y_{t+1} 加入列表 Y 的末尾
　　$[X_{t-256},\ldots,X_t] \leftarrow [X_{t-255},\ldots,X_t,Y_{t+1}]$
end for
return 列表 Y

通过循环输出，本项目可以实现任意长度的音乐输出。在样本输出时，单声部 -单音输出模型默认输出 256 长度，即 4 个小节长度的音乐，多声部 -多音输出模型则默认

输出 512 长度，即 8 个小节长度的音乐。

6.3 测试成果及性能分析

作为典型实验，我们分别对两个模型输入了“小星星”、“致爱丽丝”两种旋律作为输入序列。输出旋律见附录文件。我们认为，输出的旋律已初见艺术性。在输出中，我们发现模型创作的旋律很少有不和谐音程出现——这表明模型学习到了旋律前后的音程关系；并且，在输出的旋律中，我们注意到神经网络创作的节奏与输入的节奏相关，并在某种程度上“模仿”了输入旋律——这表明模型学习到了旋律中前一段与后一段的某种长期与短期关系，表明模型确实做到了长短期记忆。对于和弦的伴奏输出，我们发现，模型在输出中，倾向于使用与当前旋律和谐的和弦。并且，对于在巴赫数据集上训练的模型，我们发现，模型的输出在某种程度上可以听到巴赫作品的一些音乐特点，如和弦的转调与旋律音阶的运用等。

第七章　项目研究的成本核算

本项目无成本支出。