Music Generation with RNNs

实验目的

理解序列建模与 RNN/LSTM 原理，掌握 MIDI 转序列的音乐数据数值化过程；熟悉模型结构、训练流程，能通过调参提升音乐生成质量；

生成至少两段可播放的.mid 旋律文件

实验方法

1. 数据集：使用 MIDI 格式音乐数据，通过 music21 库解析为音符序列与和弦表示。
2. 模型结构：采用三层 LSTM 网络结构，每层使用 tanh 激活函数，输出层使用 softmax 预测下一个音符的类别。
3. 实验设置：在保持相同数据集与模型框架的前提下，对比原始参数与修改参数模型的表现

调参过程与结果分析

原始参数

```
### Hyperparameter setting and optimization ###

vocab_size = len(vocab)

# Model parameters:
params = dict(
    num_training_iterations = 6000,    # Increase this to train longer
    batch_size = 8,    # Experiment between 1 and 64
    seq_length = 100,    # Experiment between 50 and 500
    learning_rate = 5e-3,    # Experiment between 1e-5 and 1e-1
    embedding_dim = 256,
    hidden_size = 1024,    # Experiment between 1 and 2048
)
```

调参参数

```
vocab_size = len(vocab)

# Model parameters:
params = dict(
    num_training_iterations = 6000,    # Increase this to train longer
    batch_size = 12,    # Experiment between 1 and 64
    seq_length = 100,    # Experiment between 50 and 500
    learning_rate = 1e-3,    # Experiment between 1e-5 and 1e-1
    embedding_dim = 256,
    hidden_size = 1024,    # Experiment between 1 and 2048
)

# Checkpoint location:S
```

结果分析：首先降低了学习率：先前的学习率（0.005）就像模型“大步快跑”。在训练中，它能很快地接近目标，但是因为它步子很复杂，很容易在最模式解的“山谷”底部来回“跨过”，导致损失值上下波动所以这里降低了学习率，让相当于模型“小步慢走”。在训练一步中，模型能够非常精细地、一个脚印地探索最优解从而：使得损害的曲线非常平滑，从图上可以清楚地看到，曲线在最新的几乎没有表情，非常稳定收敛到更优的点，精细的调整让模型能够找到一个精确的损失值，这意味着模型对音乐规律的结构更加清晰。

批次大小：批量大小（Batch Size）决定了模型每次更新自己之前“看”多少个样本，将批次大小提高到12，意味着模型每次综合了12个不同样本的信息后，才如何更新自己。这就像做决策前“多问了几个人”的意见，使学习的方向更加稳定和准确。

轮次大小：因为降低了学习率所以有可能导致未学到最优解就提前结束，所以将轮次提高到6600，给了“小步慢走”的模型足够的时间去充分学习和收敛。

心得体会

通过本次实验，我深刻体会到超参数调整需在 “模型性能” 与 “训练效率” 间找平衡，控制变量法是厘清参数影响的关键。同时直观理解了 LSTM 捕捉音乐时序依赖的原理，也明白音乐质量评估需结合 Loss 曲线与听觉感受，这种 “技术 + 艺术” 的结合思维，为我后续深度学习实践带来了新启发。

实验反思问题

1. 模型为什么能学会“旋律规律”？

   RNN/LSTM模型就像一个有记忆力的艺术家。普通的神经网络模型在预测下一个语音时，只能看到当前的语音并不知道前面弹了什么。而RNN/LSTM模型的核心优势在于它的“记忆机制”，而音乐本质上是一个时间序列，音符出现顺序是关键。RNN结构是专门设计用来处理这种序列数据的在每一步都会接受到上一步的传参，通过这种“记忆机制”模型可以学习到长期的依赖关系，在训练过程中，模型通过不断调整内部参数（权重），学习去预测在给定历史序列，词汇表中每一个音符出现的概率。那些在真实音乐中经常出现的旋律模式，其对应的概率就会被模型学高，因此便可以在之后对音乐进行预测

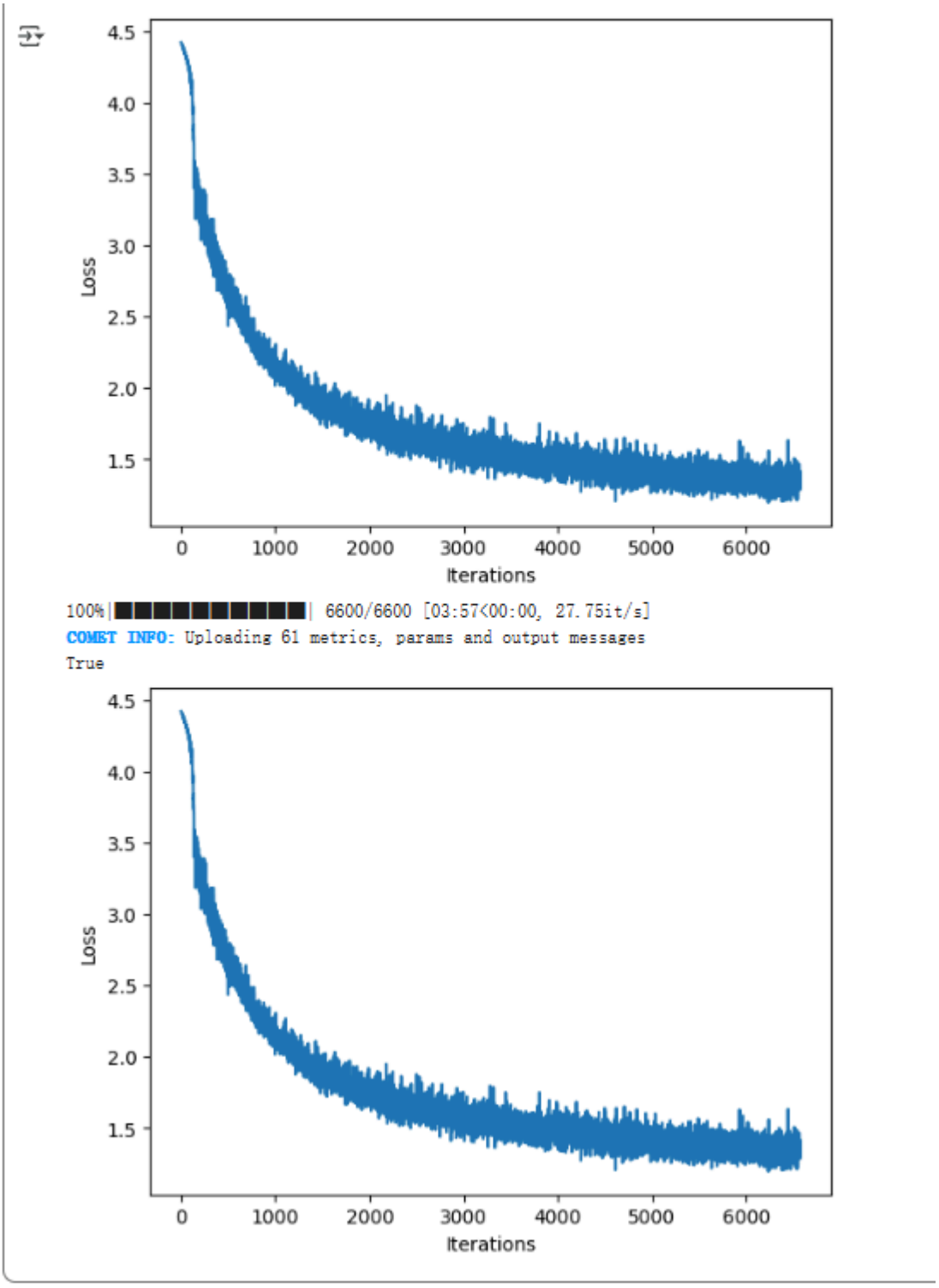
2. 为什么温度参数（temperature）会影响生成多样性？

   因为在最后一步，模型并不会直接输出一个音符，而是为刻度表中的每一个可能的字符（音符、符号等）都给出一个分数，这个分数叫做**logit**。分数大，代表模型认为这个字符是下一个的可能性更大。经过softmax将分数转换为概率：为了让这些分数变得有意义，我们用`softmax`函数将它们转换成一个概率分布。转换后，所有字符的概率值都在0到1之间，且总和为1。所以温度参数通过缩放logit值，直接控制了`softmax`输出概率分布的形状（梯度或角度），从而调节了采样过程的随机性，最终决定了生成内容的多样性

3. 你的改进在哪些方面提升了音乐的自然度或节奏感？

   改进1：调参学习速率0.001，批次大小12， 轮次6600

结果：



首先降低了学习率：先前的学习率（0.005）就像模型“大步快跑”。在训练中，它能很快地接近目标，但是因为它步子很复杂，很容易在最模式解的“山谷”底部来回“跨过”，导致损失值上下波动所以这里降低了学习率，让相当于模型“小步慢走”。在训练一步中，模型能够非常精细地、一个脚印地探索最优解从而：使得损害的曲线非常平滑，从图上可以清楚地看到，曲线在最新的几乎没有表情，非常稳定收敛到更优的点，精细的调整让模型能够找到一个精确的损失值，这意味着模型对音乐规律的结构更加清晰。

批次大小：批量大小（Batch Size）决定了模型每次更新自己之前“看”多少个样本，将批次大小提高到12，意味着模型每次综合了12个不同样本的信息后，才如何更新自己。这就像做决策前“多问了几个人”的意见，使学习的方向更加稳定和准确。

轮次大小：因为降低了学习率所以有可能导致未学到最优解就提前结束，所以将轮次提高到6600，给了“小步慢走”的模型足够的时间去充分学习和收敛。

改进2：提升温度1.0到1.5

通过提高温度，我提升了音乐的“节奏感”和“趣味性”：当我将生成阶段的温度参数从默认的`1.0`提高到`1.5`时，生成的音乐风格发生了显着的变化。

分析：高温让模型不再局限于那些最常见的音符和节奏模式，增加了采样的随机性如同上面所讲的高温直接控制了`softmax`输出概率分布的形状（梯度或角度），从而调节了采样过程的随机性，最终决定了生成内容的多样性所以在听感上这使得生成的音乐在节奏上更加丰富的多变。相比于基准模型中规中矩的四分音和八分音组合，高温版本出现了更多有趣的切分音和节奏变化，使得音乐不再单调。虽然牺牲了部分稳定性（偶尔有几个音听起来很“怪”），但整体的操作性和“惊喜感”大大增强。

4. 如何判断“音乐质量”的好坏？是否存在客观指标？

   好的音乐质量通常体现在以下几个方面：

   - 旋律性（Melodiousness）：旋律是否优美、动听、易于记忆
   - 节奏感（Rhythm）：是否节奏稳定、有趣、有律动感
   - 结构感（Structure）：音乐是否听起来有组织？是否有重复的主题、发展的乐句和完整的结尾，而不是一段无始无终的随机序列
   - 和声（若何声部）/调性：对话之间的布置是否协和？是否围绕一个中心调性展开
   - 新颖性/创意（新颖性/创意）：音乐听起来就像无数旋律的拼接，还是有自己独特的风格和出人意料的转折

     但是不存在一个能够完全替代人类听感的单一侦查指标，“好因为听”本身就包含着复杂的情感和文化因素。所以无客观指标