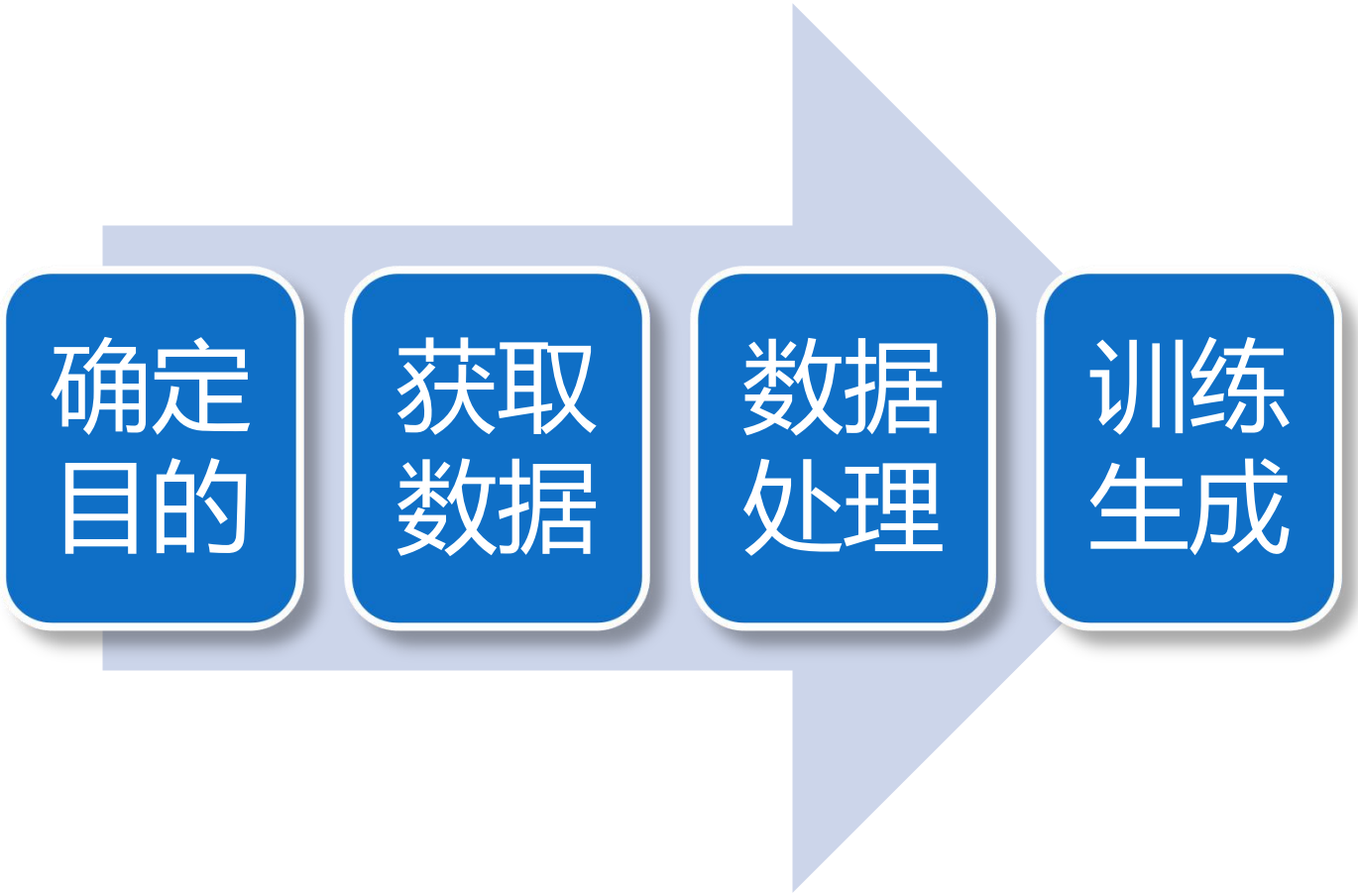
西安鼓乐项目概述

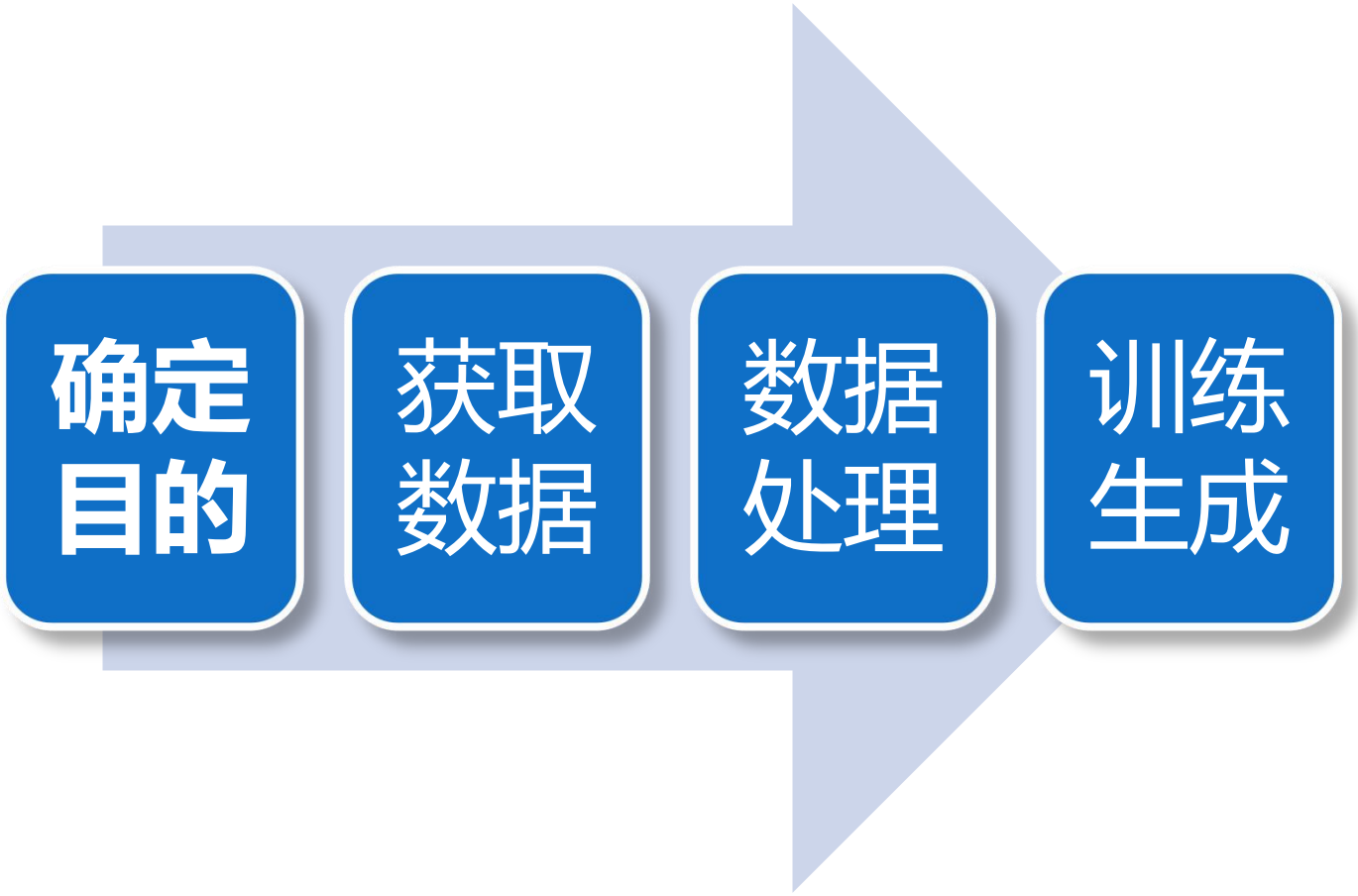
章奕林

2019年5月15日

总体流程



确定目的



研究背景和目标

研究背景——传统文化保护

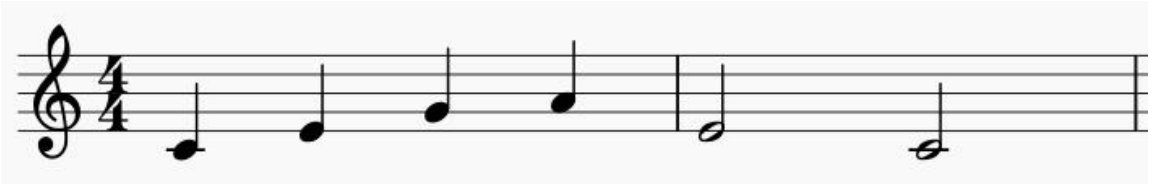
- 无法仅靠书面乐谱完成传承
- 只有经过民间艺人的韵曲才能演奏
- 民间鼓乐社中能够韵曲的老艺人相继辞世，乐社里还有很多乐谱无人能韵

目标

- 借助机器学习，完成自动韵曲
- 发现西安鼓乐的韵曲特点、风格规律
- 具有交互式功能，辅助人工作曲

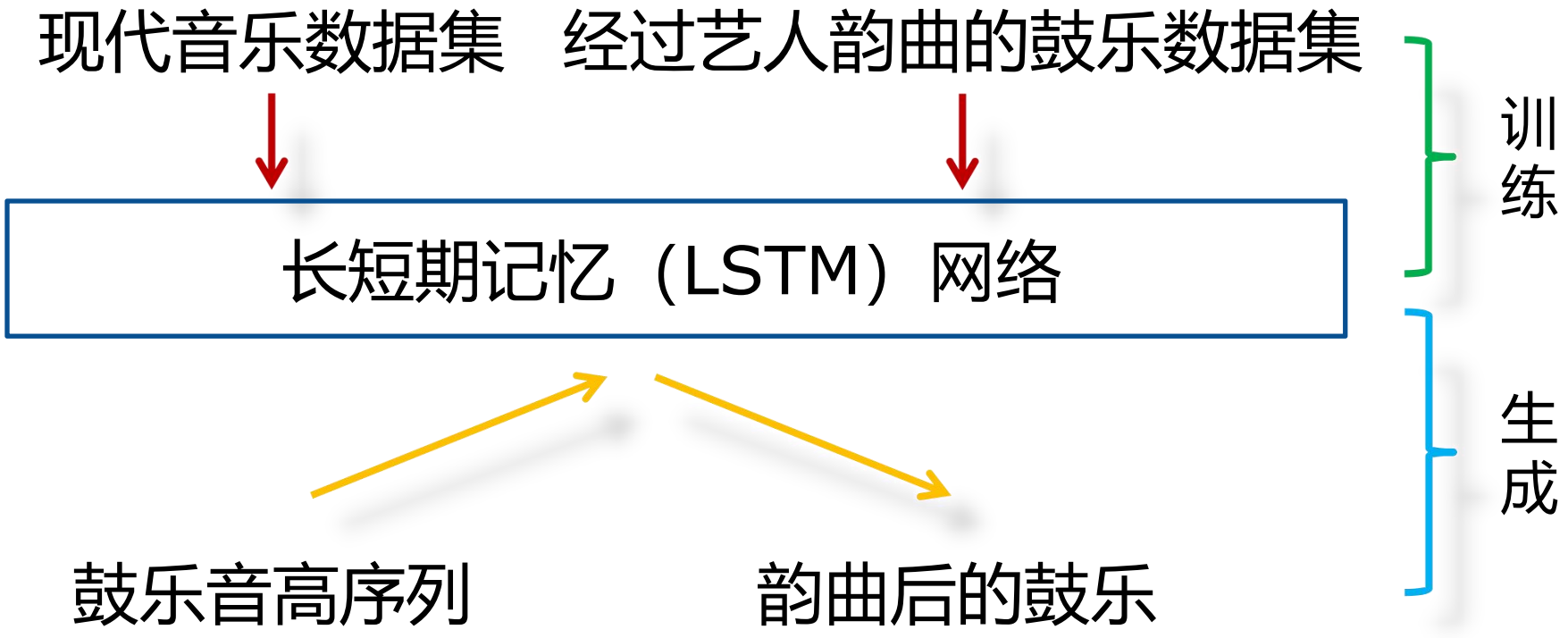
项目具体目的

- 鼓乐韵曲
  - 生成谱外音
  - 给定音高序列，生成时值（节奏）
- 输入：音高序列
  - [C, E, G, A, E, C]
- 输出：音符序列

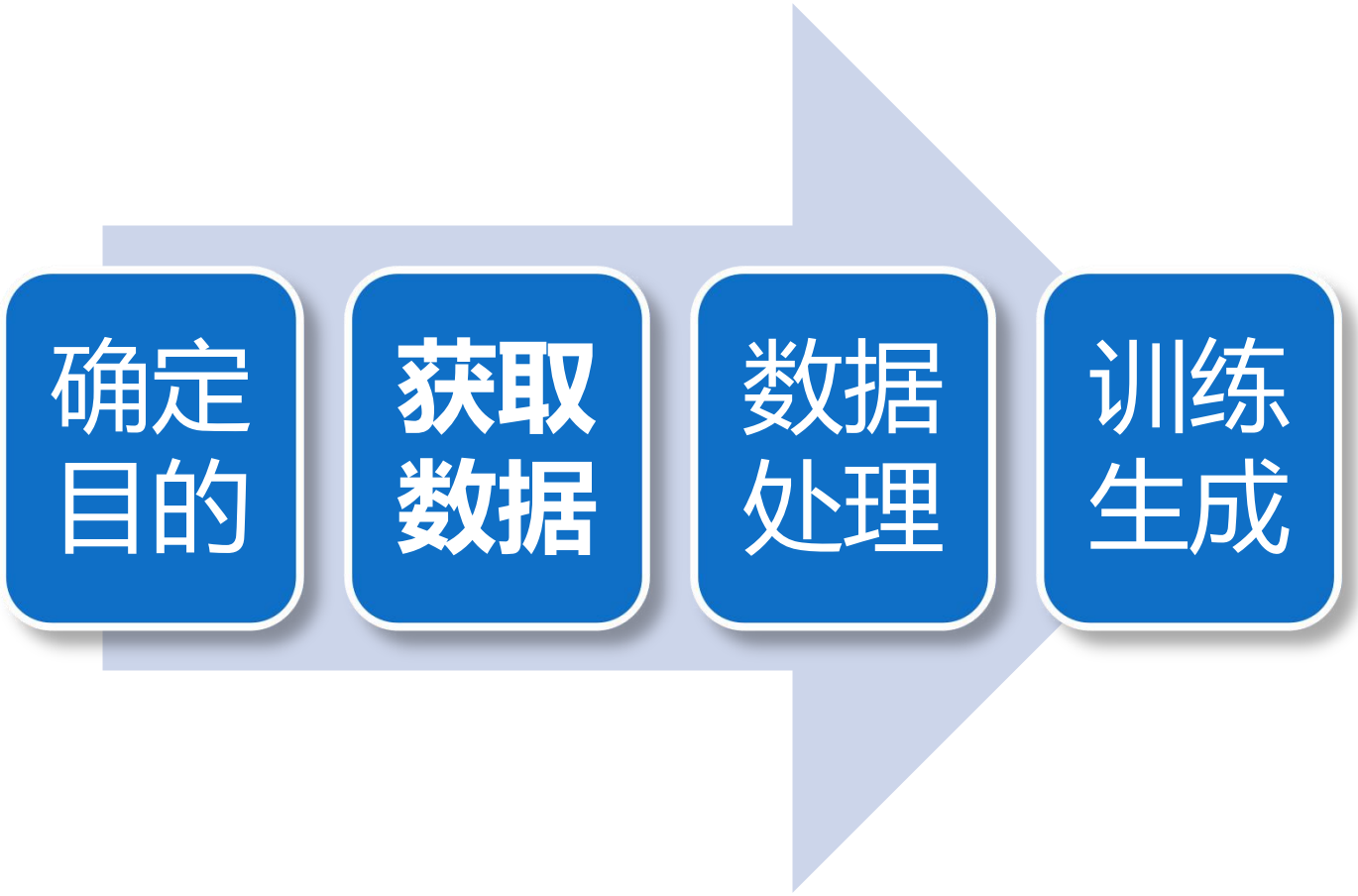


- 随机性（创造性）、不可交互

技术手段



获取数据



获取数据——现代音乐

- The Lakh MIDI Dataset v0.1
- https://colinraffel.com/projects/lmd/
- **Clean MIDI subset** - A subset of MIDI files with filenames which indicate their artist and title (with some inaccuracy), as used in a few of my papers.

获取数据——西安鼓乐

- 《西安鼓乐古曲谱集(四调八拍坐乐全套)》
  - 作者，译者：马西平
  - 古谱来源：赵庚辰老艺人手抄谱
  - 其中刘字调、尺字调、商字调用古谱、五线谱、简谱三种方式记载
- MusicXML 格式

MIDI

- Musical Instrument Digital Interface
- MIDI中包含了实时演奏信息和控制信息
- 以二进制形式表示
- 需要使用第三方工具进行解析
  - 音序器（Sequencer)
  - 编程库（如Python的mido库)

数据处理

数据处理目的

- **MIDI音符 → 向量编码**
- **MIDI音符序列→ 向量序列**

- 分为两步：

1. 数据预处理
2. 获取训练用数据集

编程库的选择

- Python语言
- 对底层MIDI处理细节有一定的封装
- 提供高层功能，便于信息检索

- mido库：处理底层MIDI信息
- pretty_midi库：基于mido库，提供高层功能
  - http://craffel.github.io/pretty-midi/

数据预处理

- **原始MIDI → 单音旋律MIDI**
- raw_midi → processed_midi

- 去除打击乐通道
- 量化（单位：十六分音符）
- 旋律提取

- `midi_preprocess.py`

数据预处理——去除打击乐通道

- `MidiData.dropdrum()`

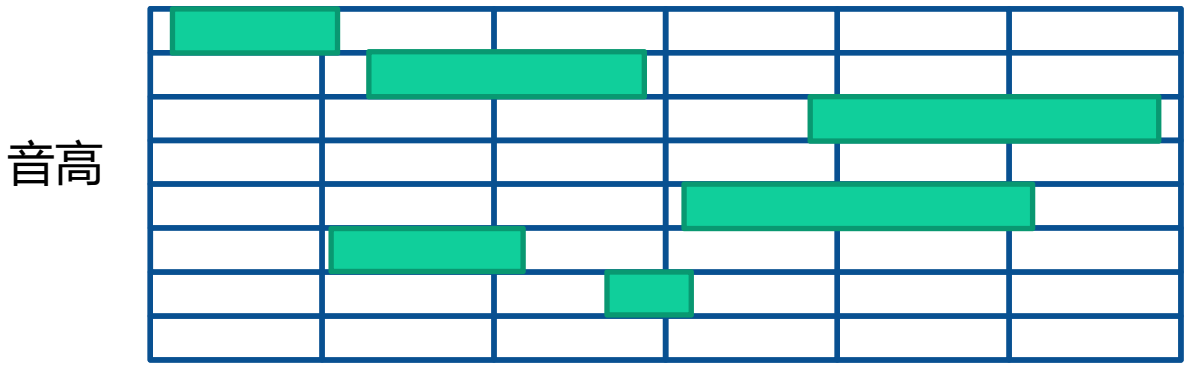
- 使用 pretty_midi 中，instrument 对象的 is_drum 属性来判断是否是打击乐通道。
- 打击乐通道在 MIDI 中一般为10号通道。

- 通道（Channel）≠ 轨道（Track）
- 通道（Channel）对应 乐器（Instrument）

数据预处理——量化

- `MidiData.quantize()`

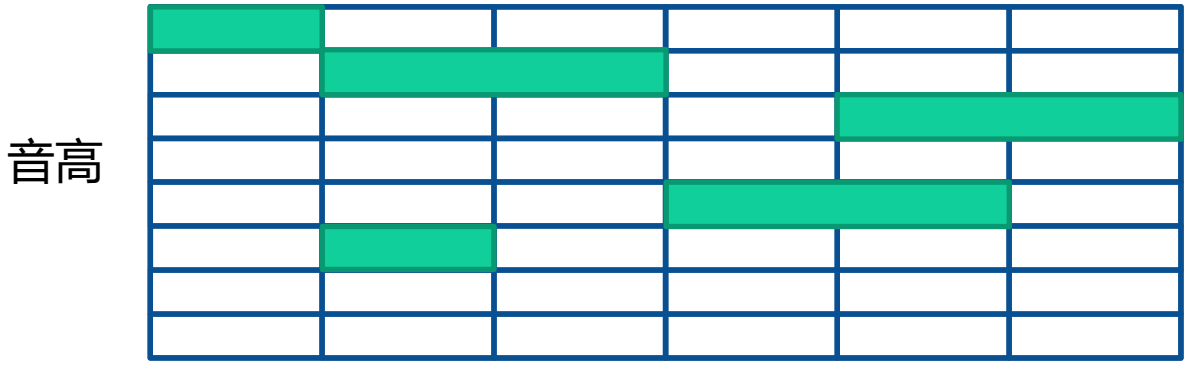
- 将音符的起止位置对齐到网格
- 以十六分音符长度为网格单位
- 剔除小于单位长度的音符



数据预处理——量化

- `MidiData.quantize()`

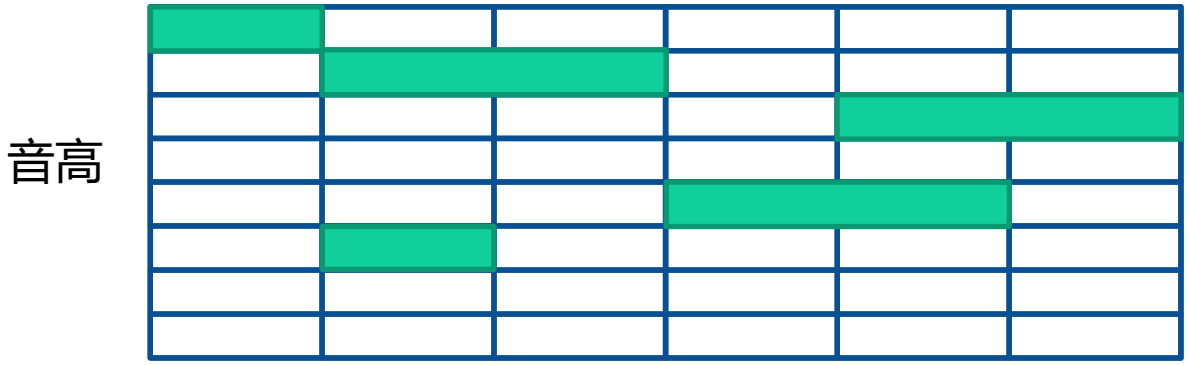
- 将音符的起止位置对齐到网格
- 以十六分音符长度为网格单位
- 剔除小于单位长度的音符



数据预处理——旋律提取

- `MidiData.skyline()`

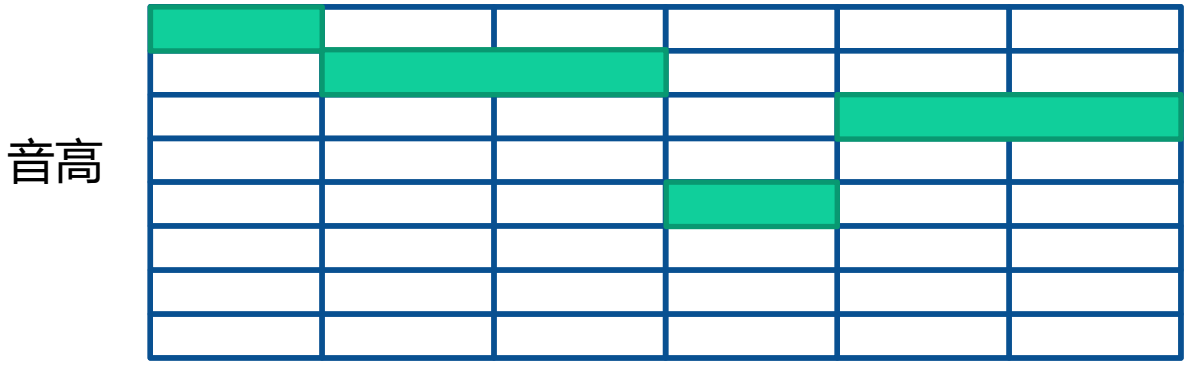
- 使用 Skyline 算法提取旋律
- 提取最高音



数据预处理——旋律提取

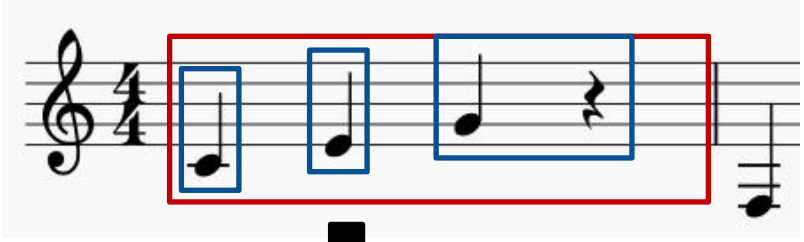
- `MidiData.skyline()`

- 使用 Skyline 算法提取旋律
- 提取最高音



获取训练用数据集

- **单音旋律MIDI → 编码后的音符序列**
- preprocessed_midi → dataset
- [音程，时值，休止时值]



- [0, 4, 0], [4, 4, 0], [3, 4, 4]

3个音符对象

- `get_dataset.py`

获取训练用数据集

- 但这三个属性需要有范围的约束

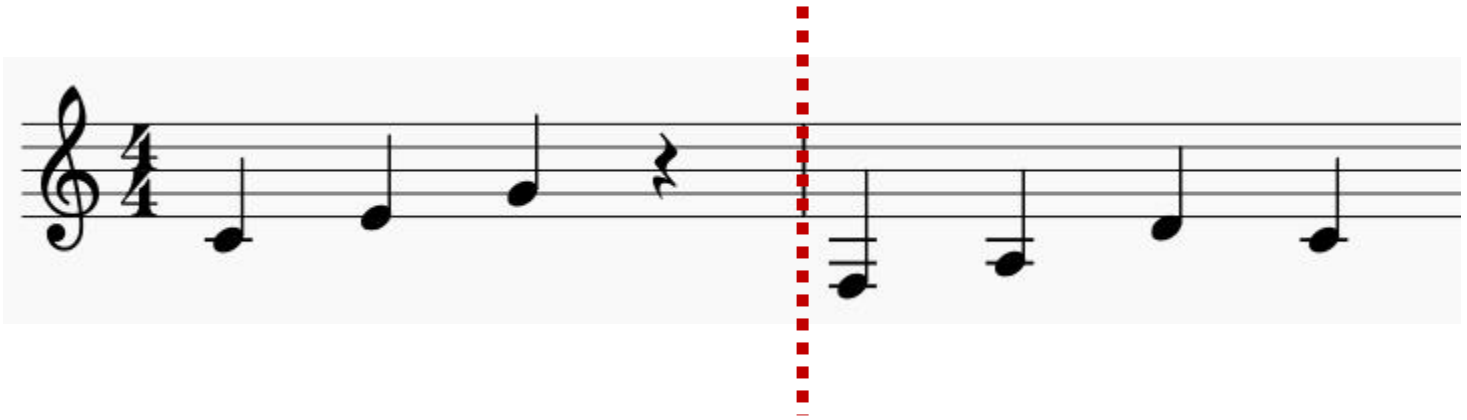
[0 0 0 … 1 … 0 0 0 0 … 1 … 0 0 0 0 … 1 … 0]

音程　　时值　　休止时值

- 规定如下：
- 音程：±12
- 时值：16
- 休止时值：16

获取训练用数据集

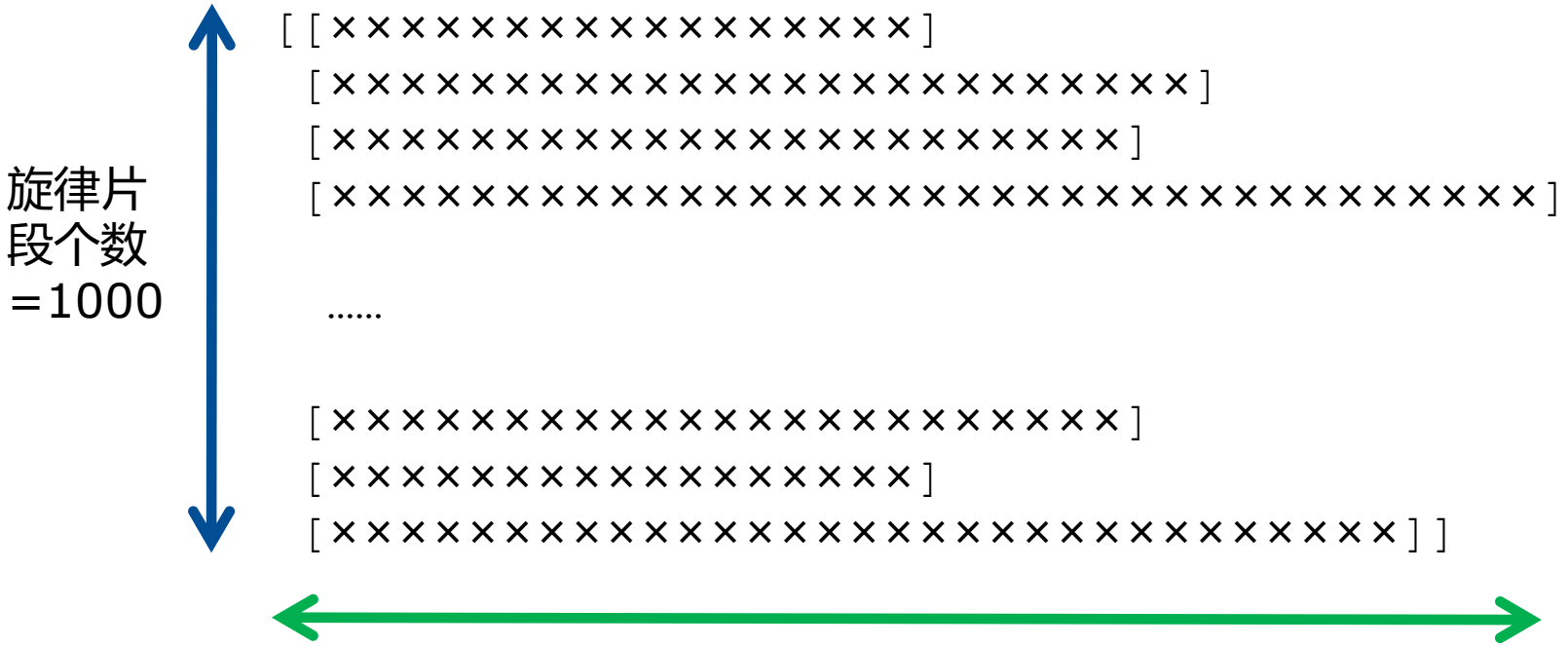
- 在不满足处截断，截取旋律片段



- 规定 LSTM 网络的步数为16
- 只选取音符对象数大于等于17的旋律片段

获取训练用数据集

- 一个数据文件（.pkl后缀）的构成
- 下面以 × 代表音符对象



获取训练用数据集

- 将所有数据切分为：
  - 训练集（training set)
  - 验证集（validation set)
  - 测试集（test set)
- 在 dataset 目录下建立三个子目录
  - train
  - valid
  - test
- 按照给定比例，随机选取文件到三个目录

鼓乐数据集获取

- 与现代音乐MIDI的处理手法基本相同

- 不同点：
- 先由 MusicXML 转换为 MIDI
- 只需要量化，无需剔除打击乐和旋律提取
- 一个数据文件只包含一首曲子的所有旋律片段

- `xadrum_preprocess.py`
- `xadrum_get_dataset.py`

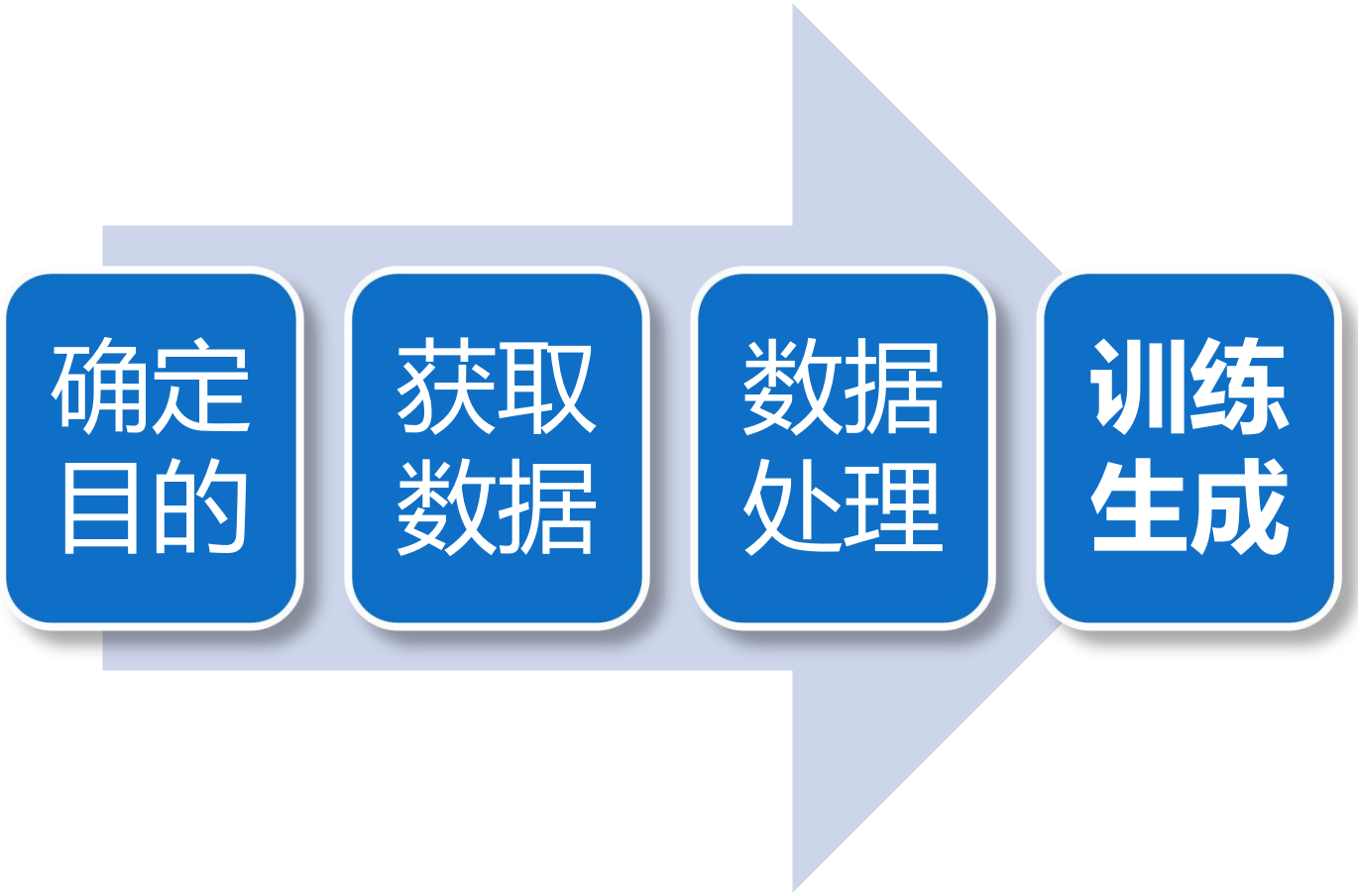
回顾

- 目的：**MIDI音符序列→ 向量序列**

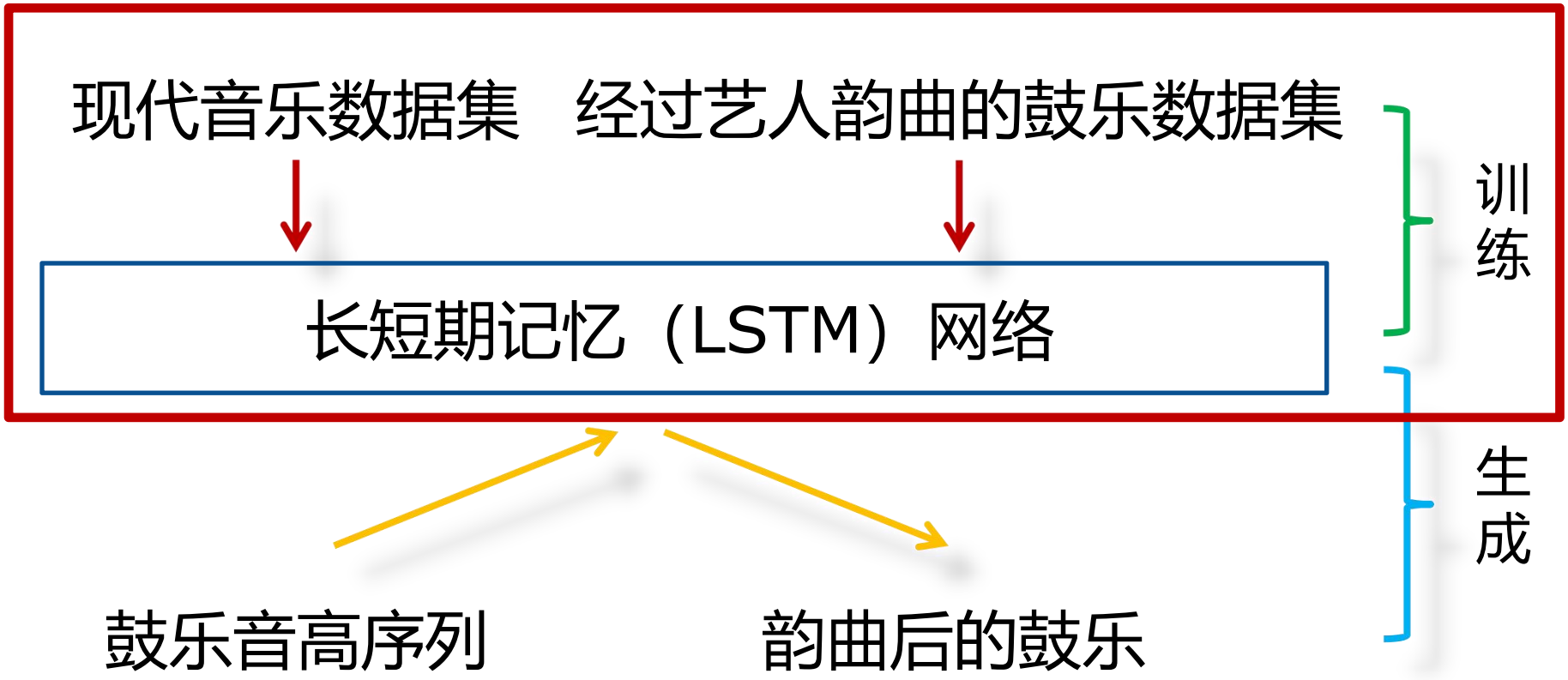
- 现代音乐数据
  - raw_midi（多声）
  - preprocessed_midi（无打击乐、量化、单音）
  - dataset（编码后的旋律片段的集合）

- 鼓乐数据
  - xadrum_midi（单音）
  - xadrum_processed_midi（量化、单音）
  - xadrum_dataset（编码后的旋律片段的集合）

训练生成

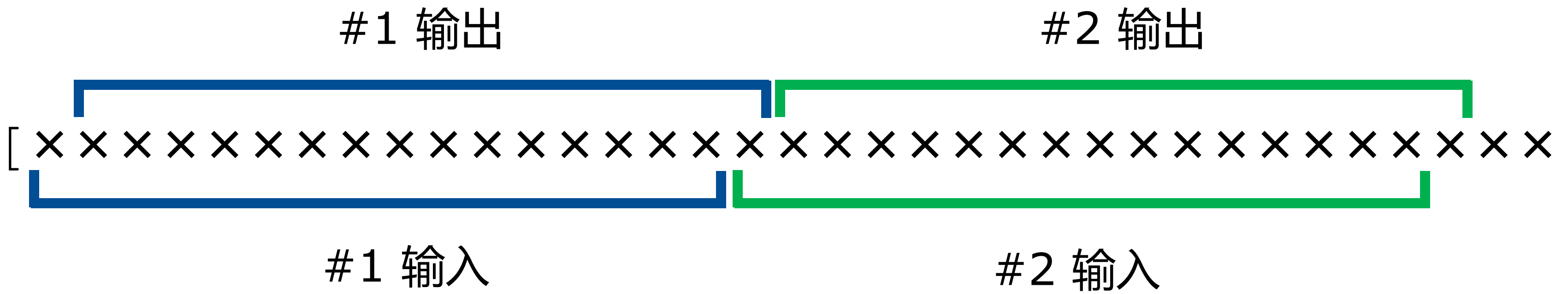


训练部分



获取输入和输出

- 在旋律片段中进行切片（×代表音符对象）



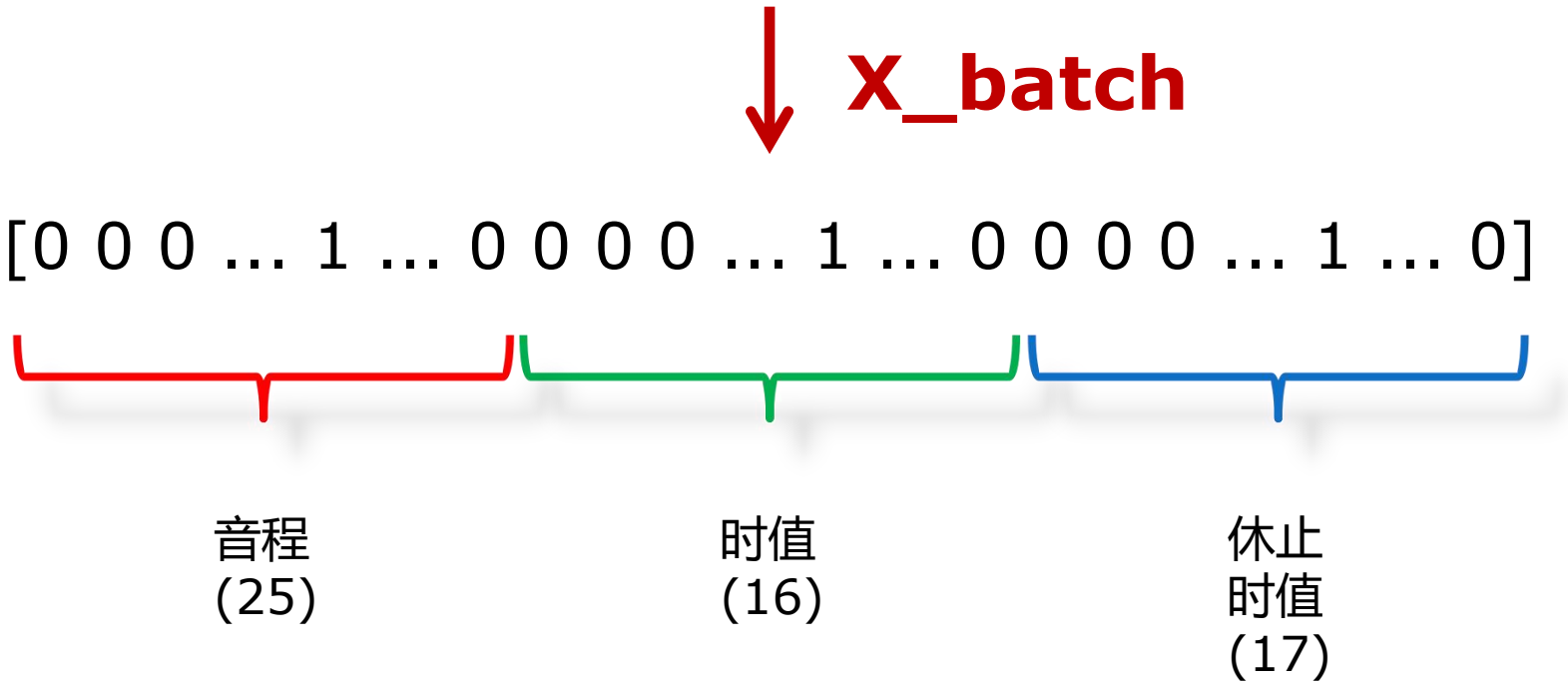
X_batch=
[multihot(#1 输入),
  multihot(#2 输入),
  multihot(#3 输入),
  ......
  multihot(#n 输入)]

y_batch=
[multihot(#1 输出),
  multihot(#2 输出),
  multihot(#3 输出),
  ......
  multihot(#n 输出)]

音符对象向量 → multihot向量

[音程，时值，休止时值]

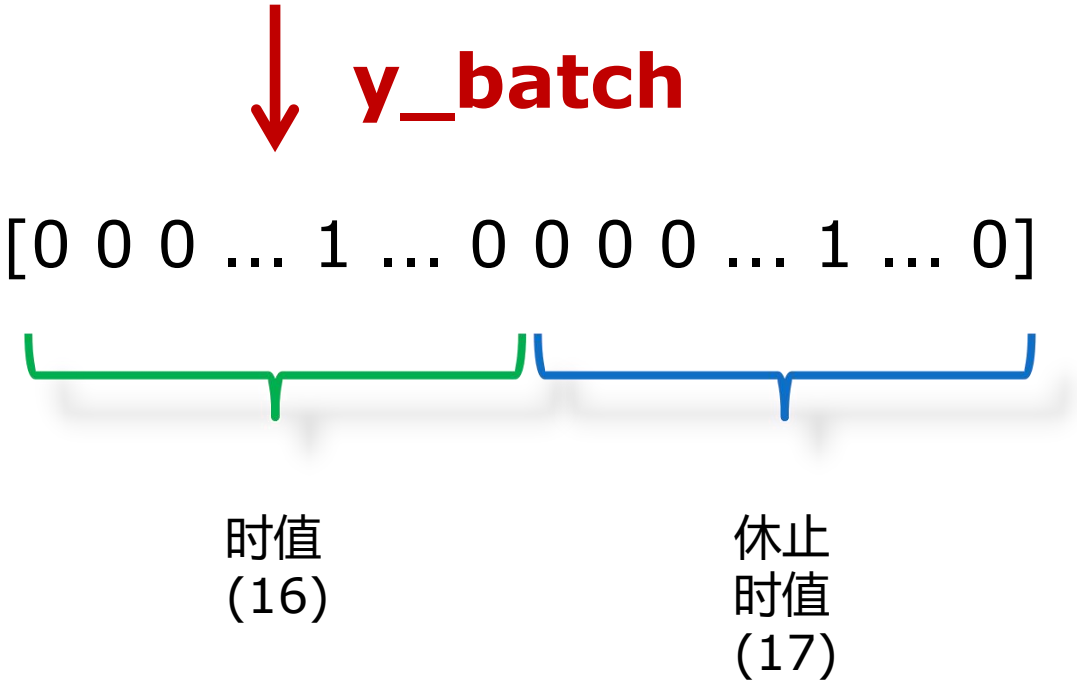
例如：[4, 4, 2]



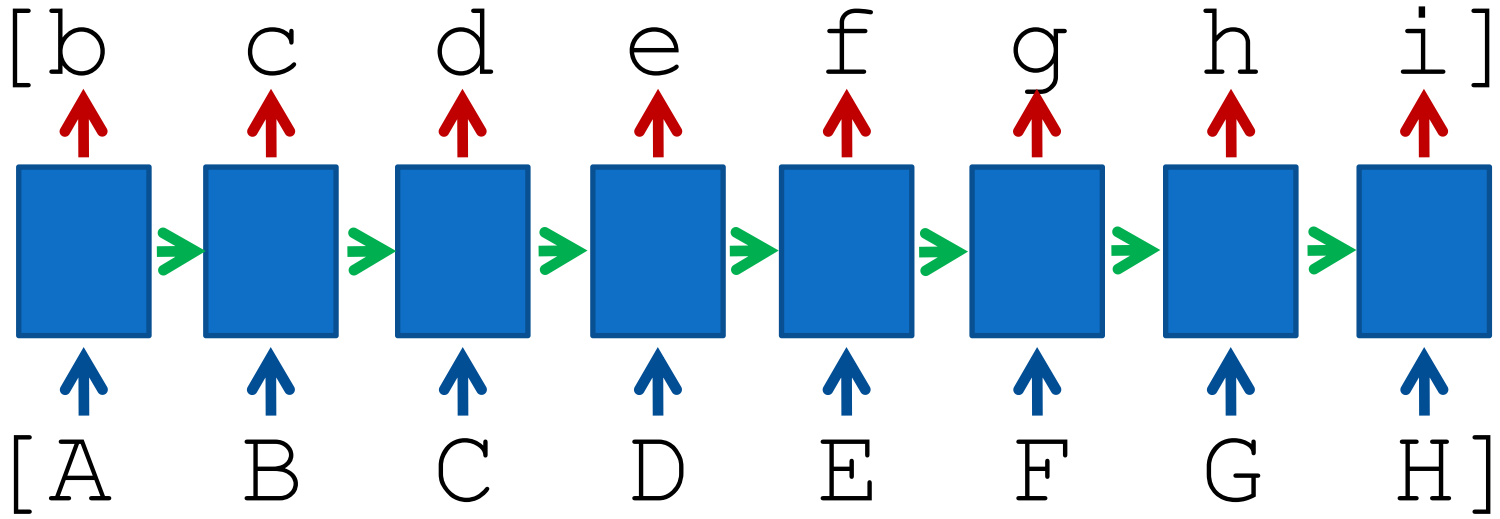
音符对象向量 → multihot向量

[音程，时值，休止时值]

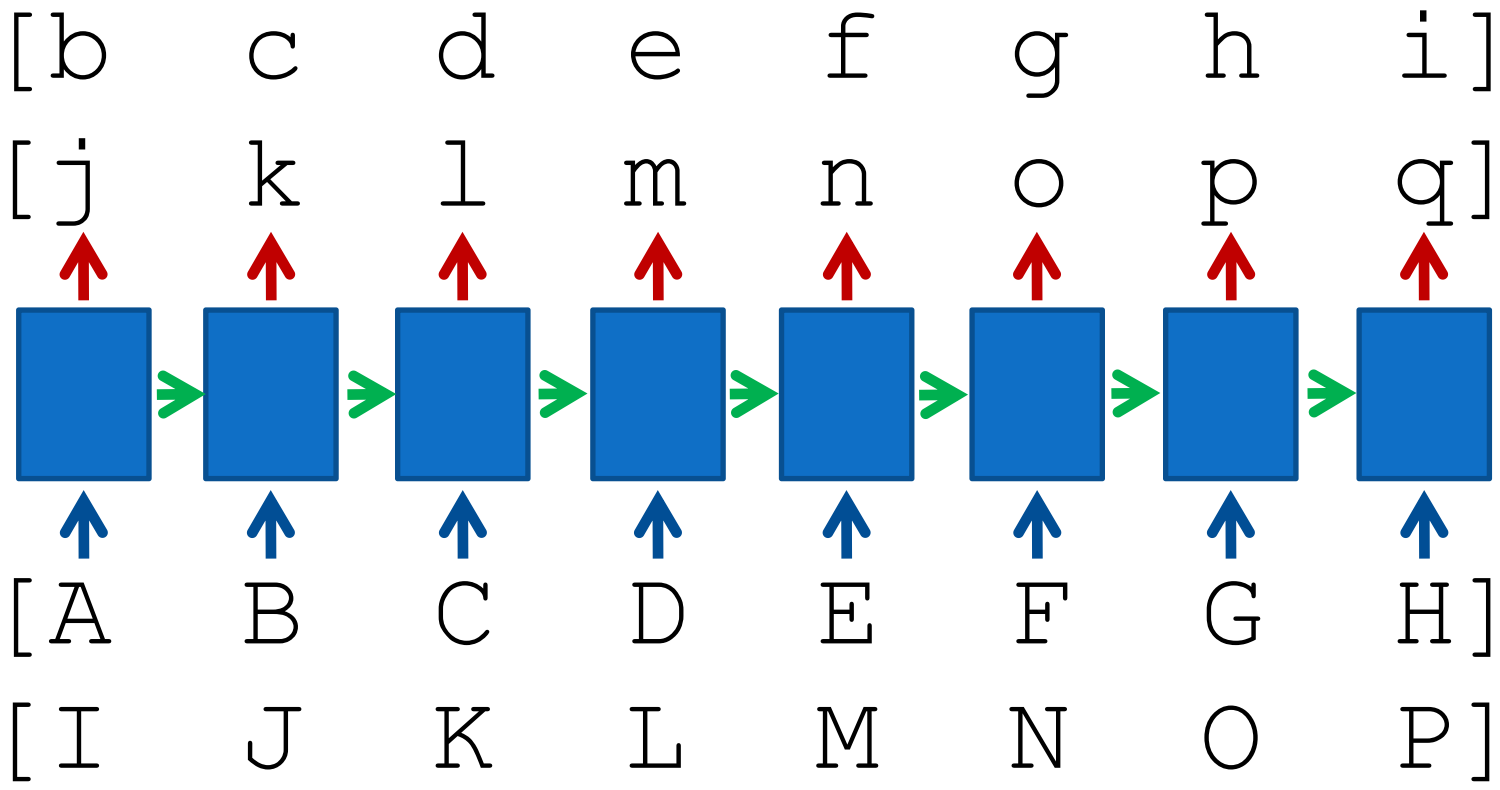
例如：[4, 4, 2]



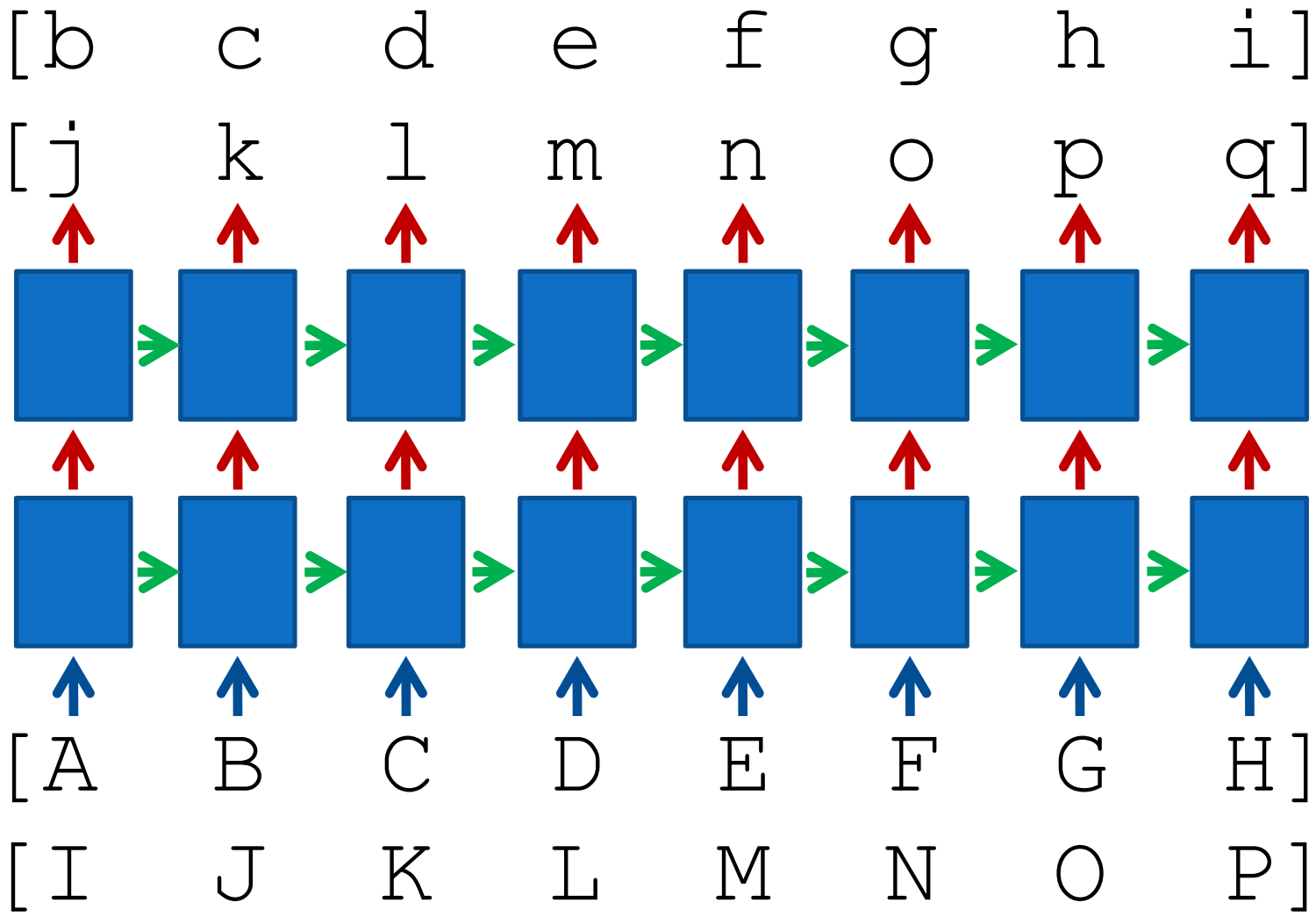
模型训练——基础RNN



模型训练——Batch



模型训练——多层



模型训练——参数设定

模型参数

- 神经元维度：256
- LSTM 层数：4
- dropout 率：30%

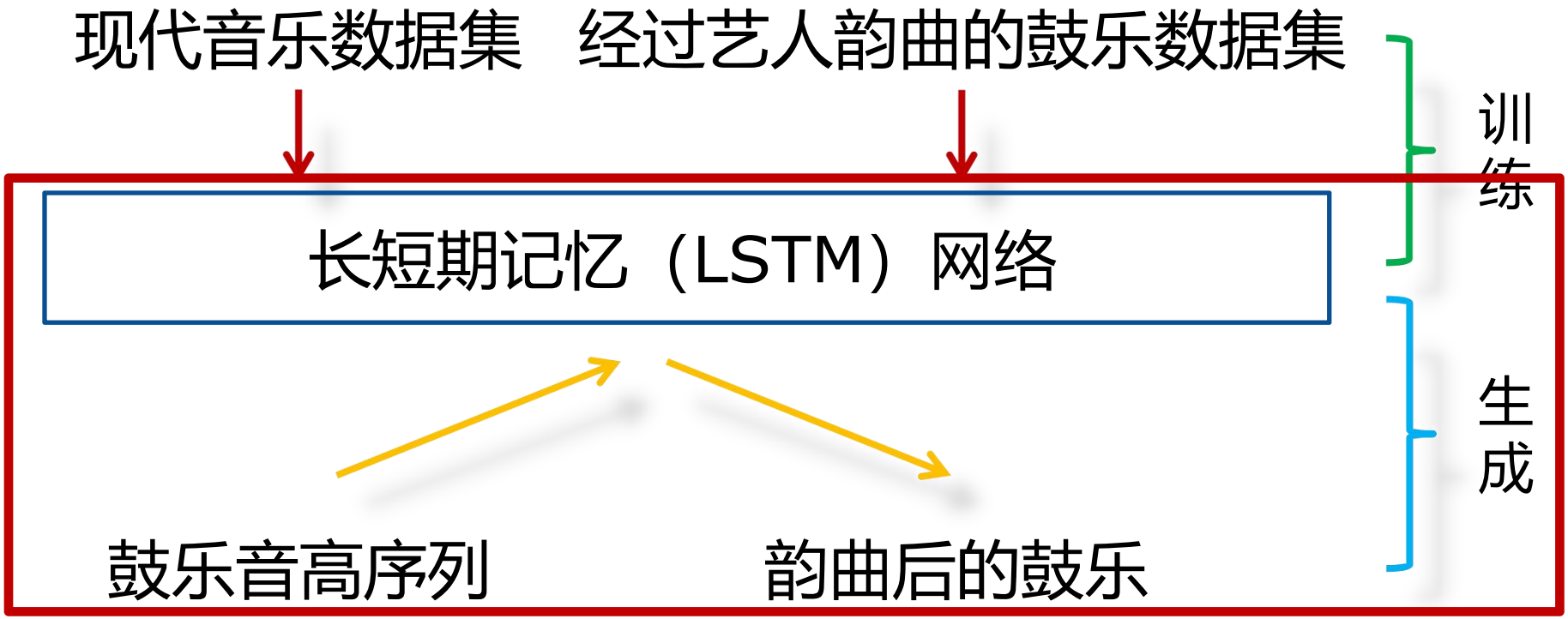
源代码文件

- `lstm_model.py`
- `lstm_model_continue.py`

训练数据参数

- 现代音乐：
  - batch size：40
  - 迭代次数：50轮
- 鼓乐：
  - batch size：5
  - 迭代次数：100轮

生成部分

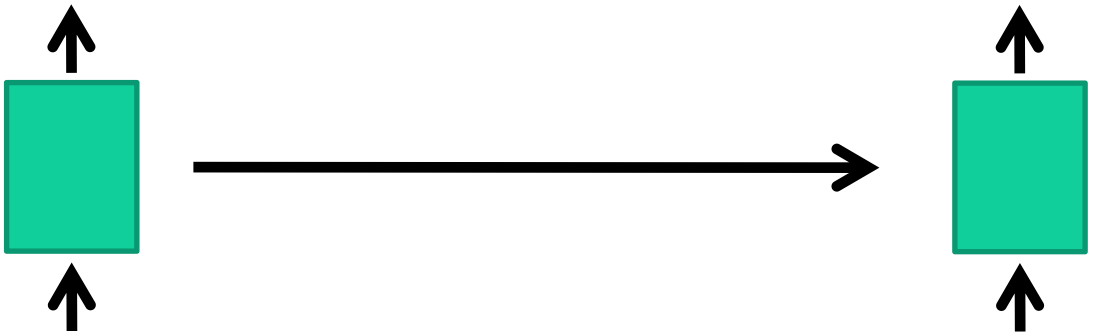


生成节奏

输入、输出都是 multihot 向量，这里只是简化表示

[b时值，b休止时值]

[c时值，c休止时值]



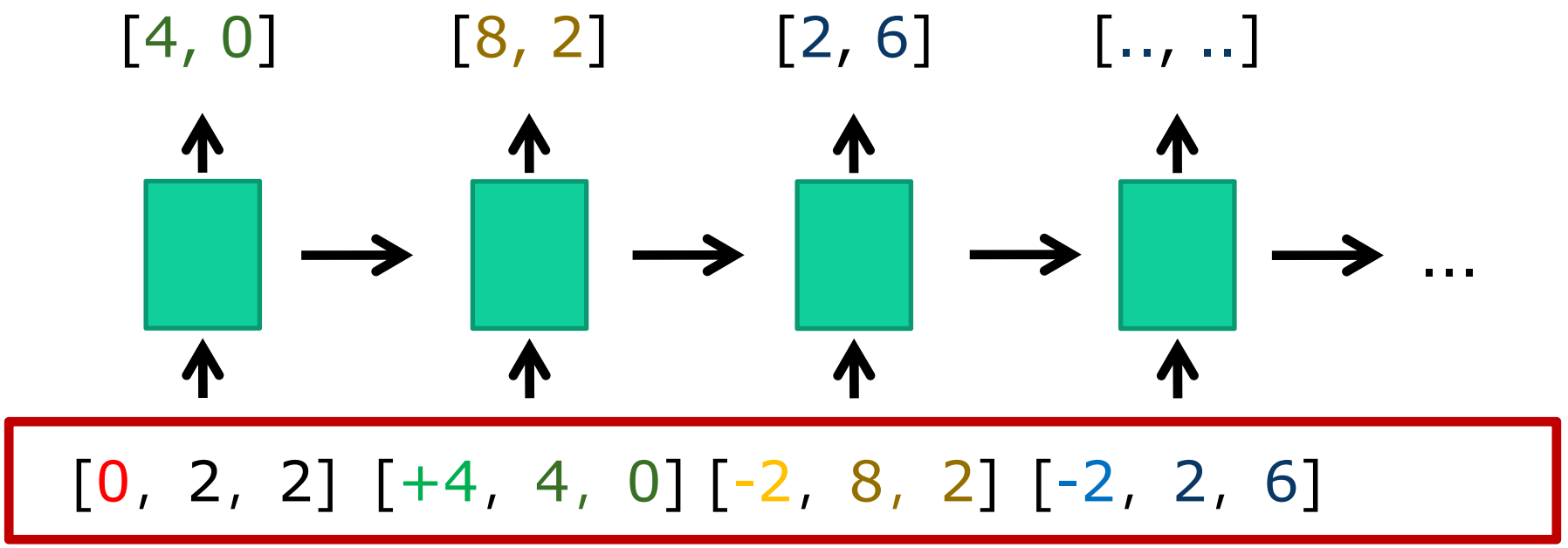
[A音程，_时值，_休止时值] [B音程，b时值，b休止时值]

音程序列：[A音程，B音程，C音程，D音程，……]

音高序列：[A音高，B音高，C音高，D音高，……]

生成节奏

输入、输出都是 multihot 向量，这里只是简化表示

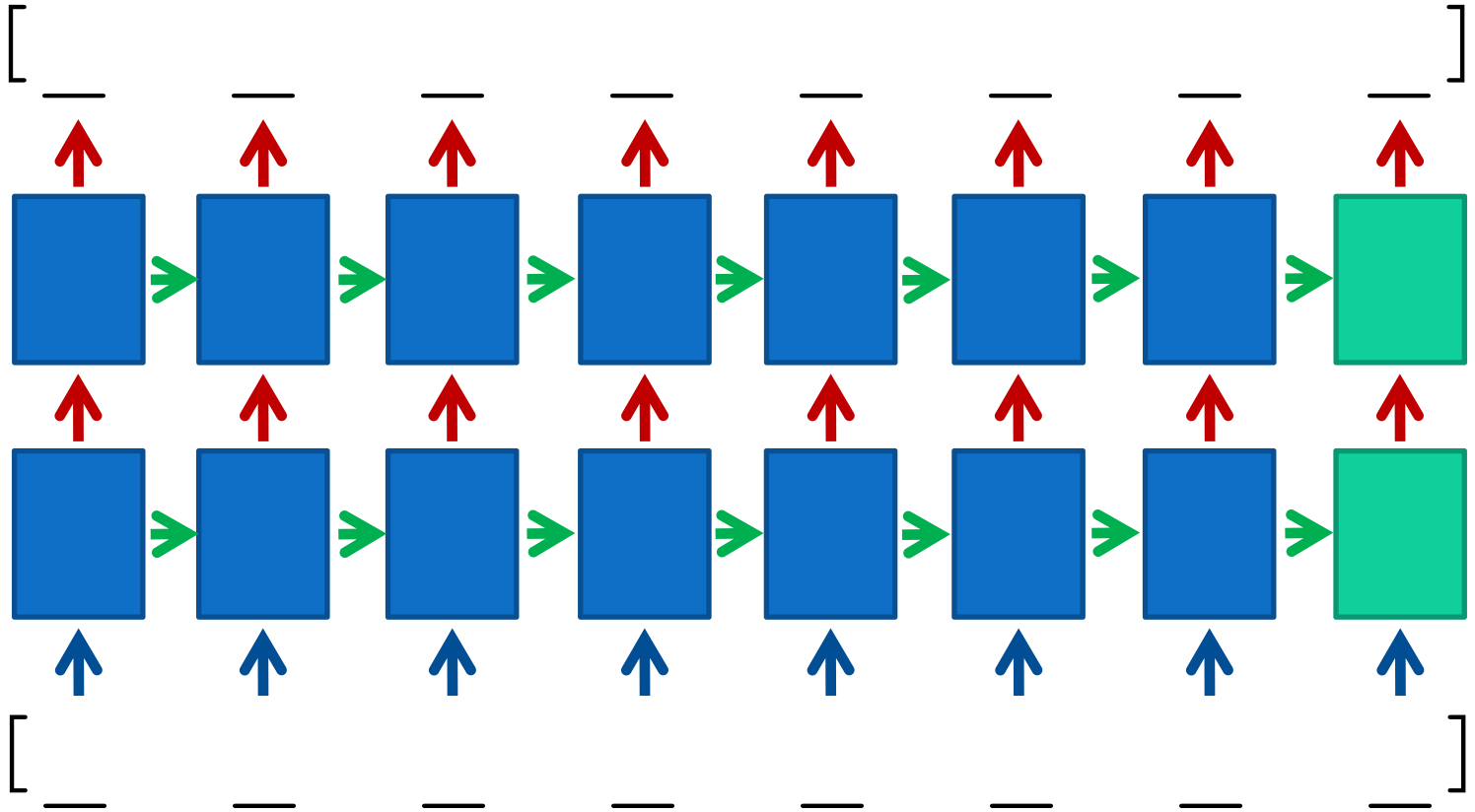


音程序列：[0, +4, -2, -2, ……]

音高序列：[60, 64, 62, 60, ……]

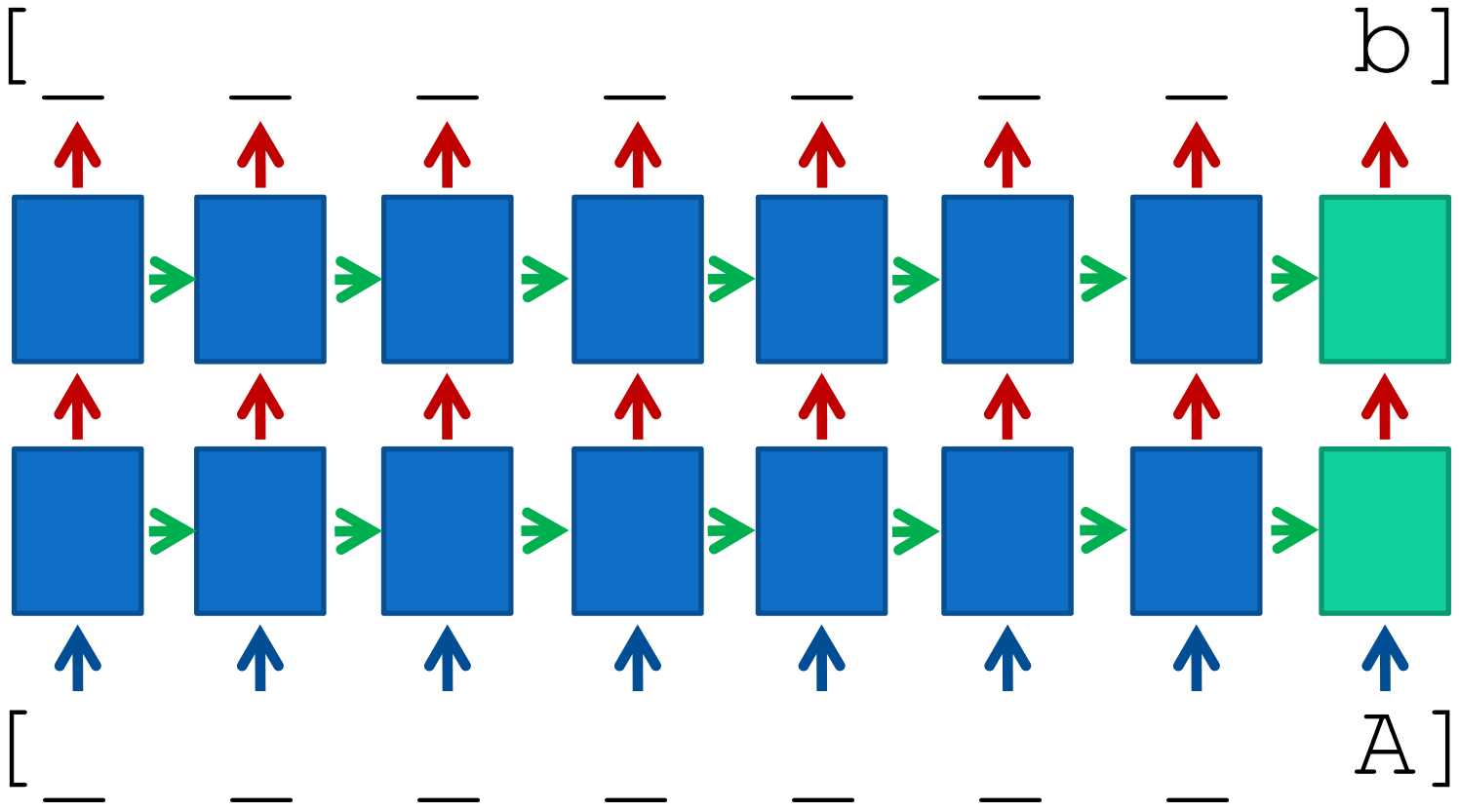
迭代生成

- 下划线 _ 代表随机生成的数据



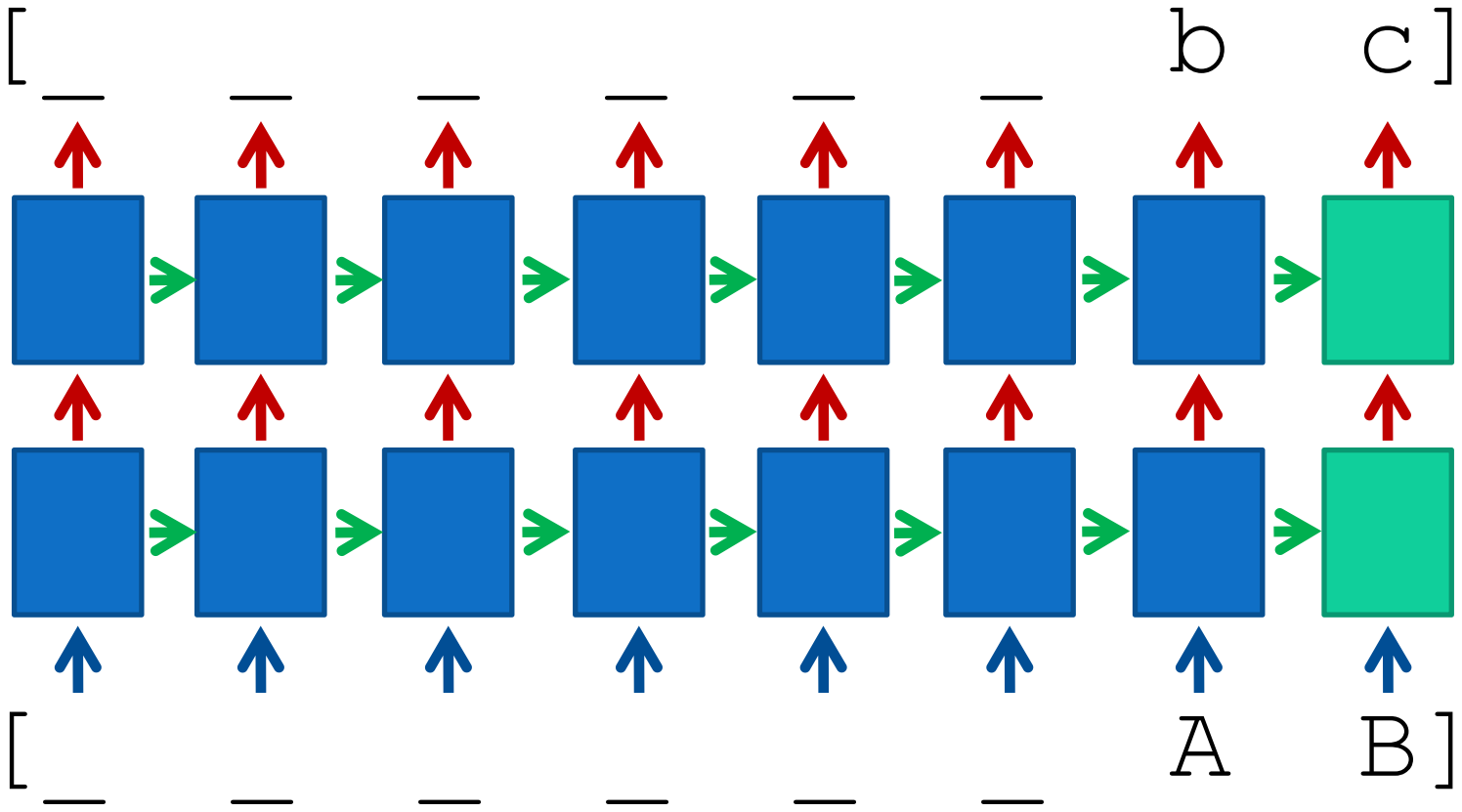
迭代生成

- 下划线 _ 代表随机生成的数据



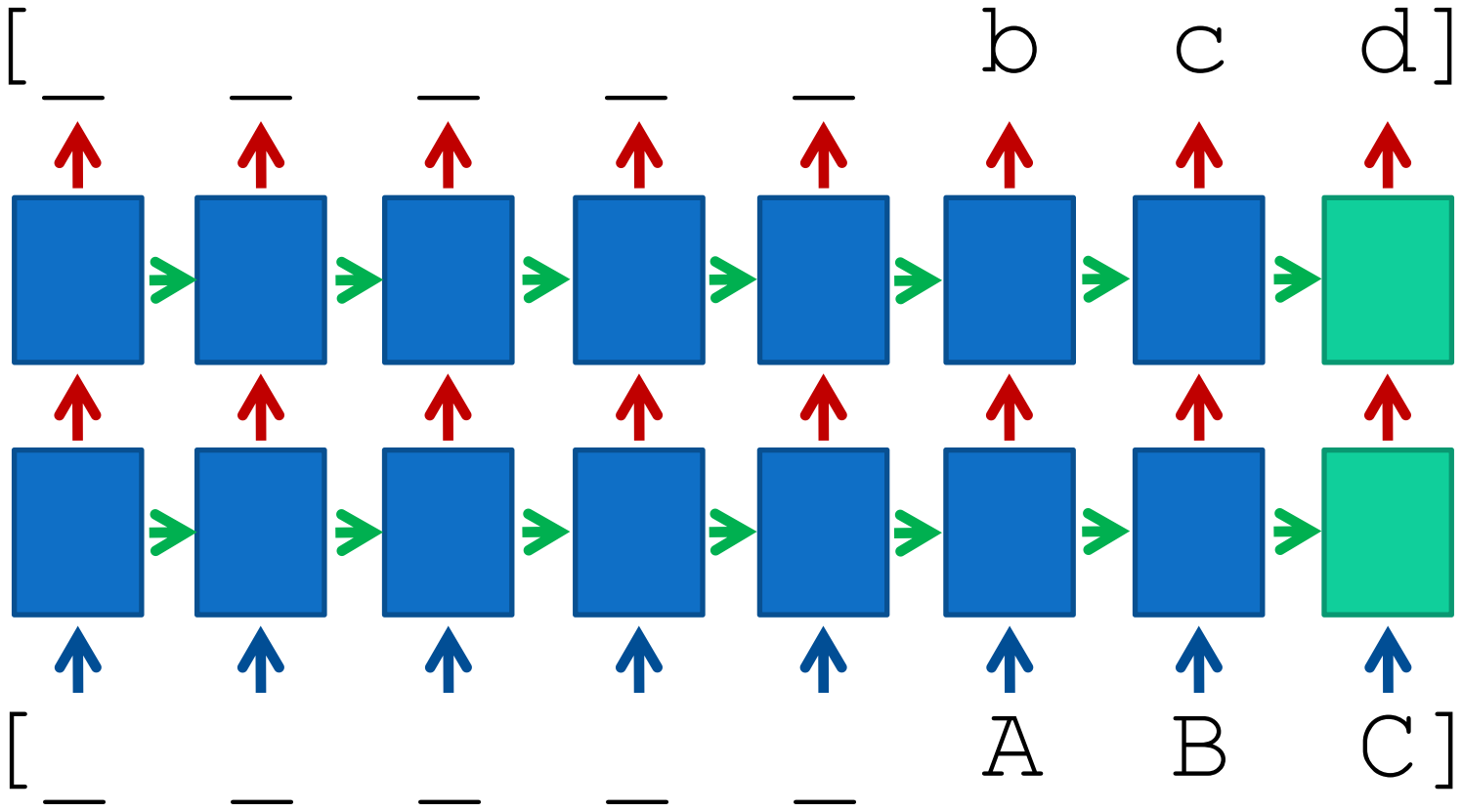
迭代生成

- 下划线 _ 代表随机生成的数据



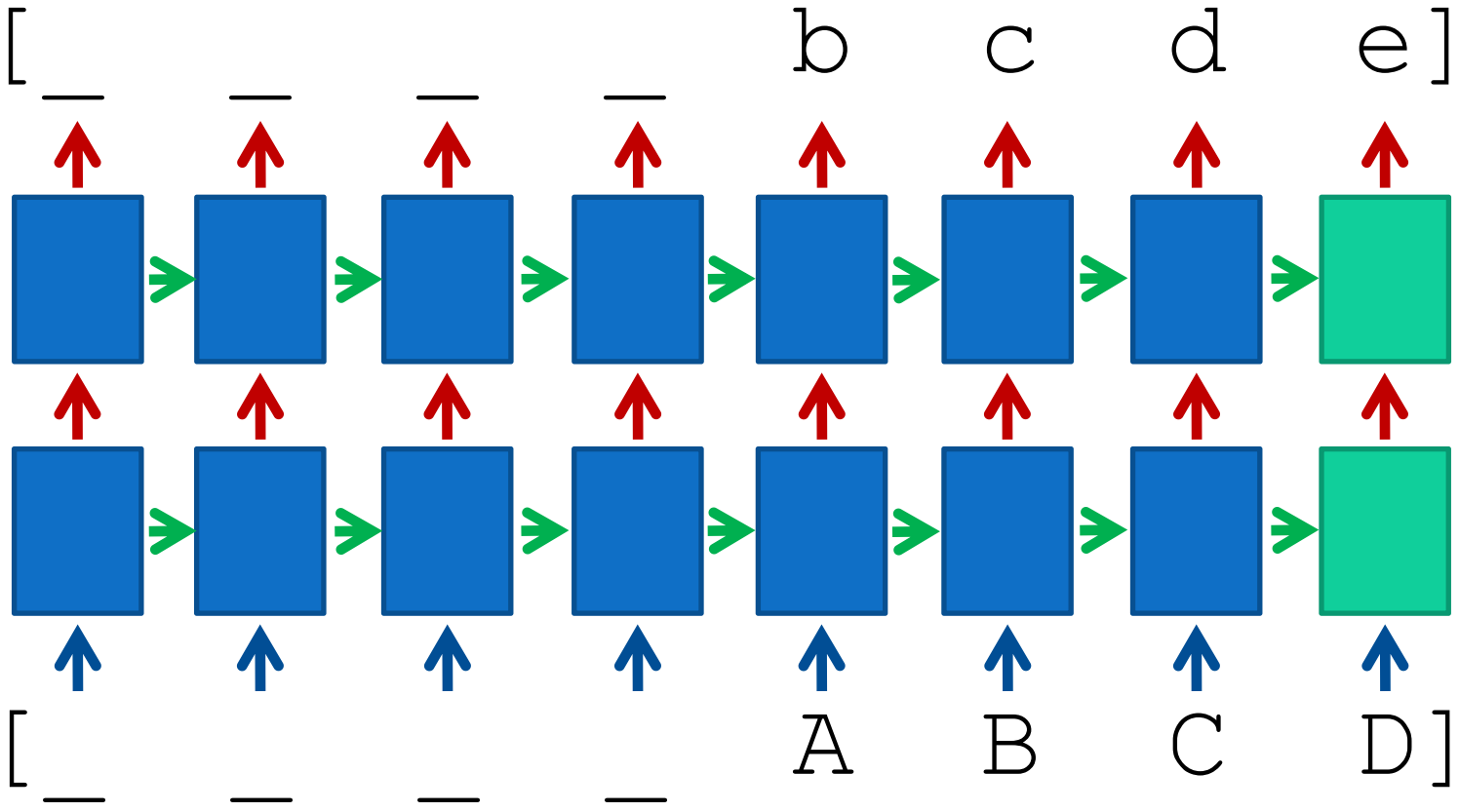
迭代生成

- 下划线 _ 代表随机生成的数据



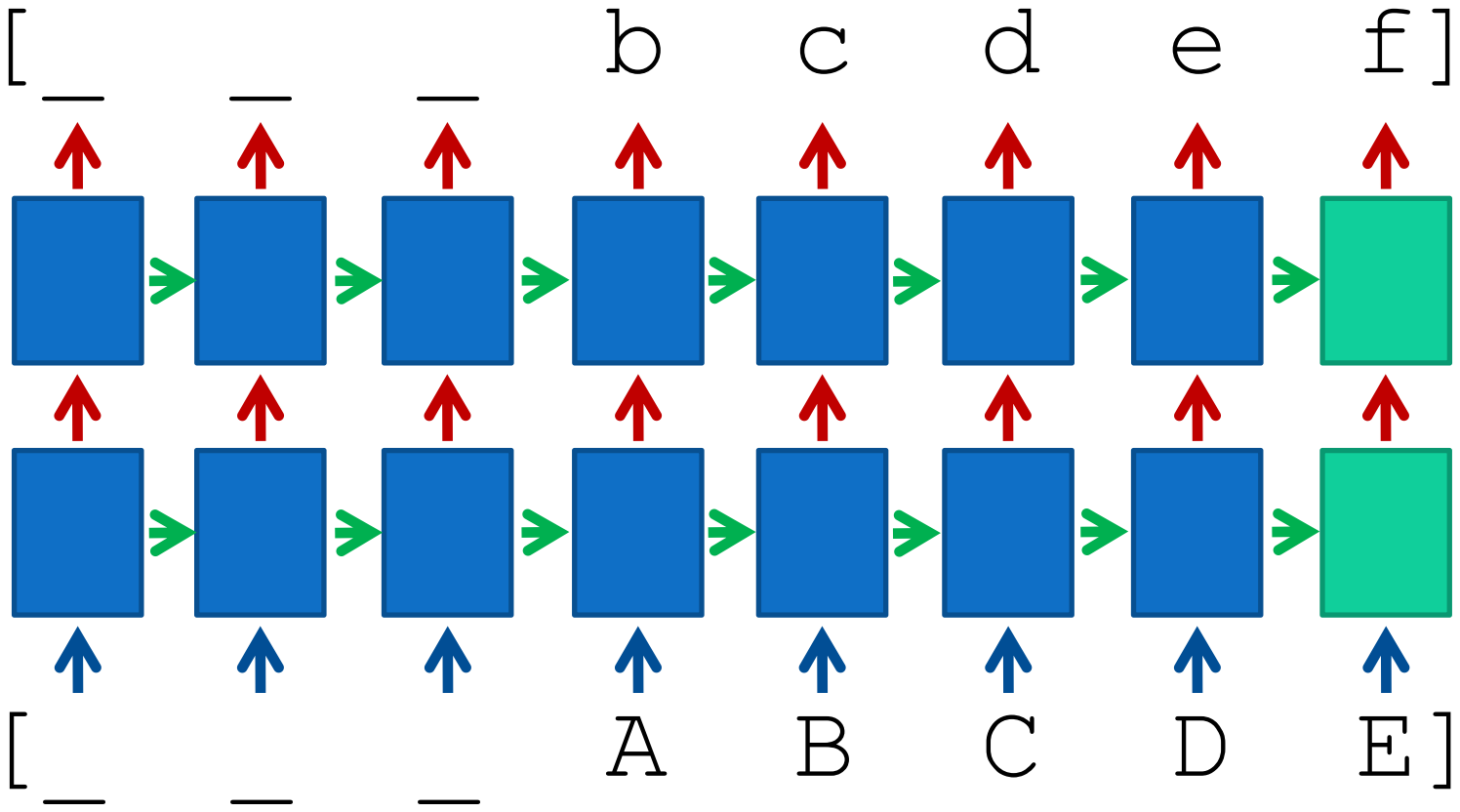
迭代生成

- 下划线 _ 代表随机生成的数据



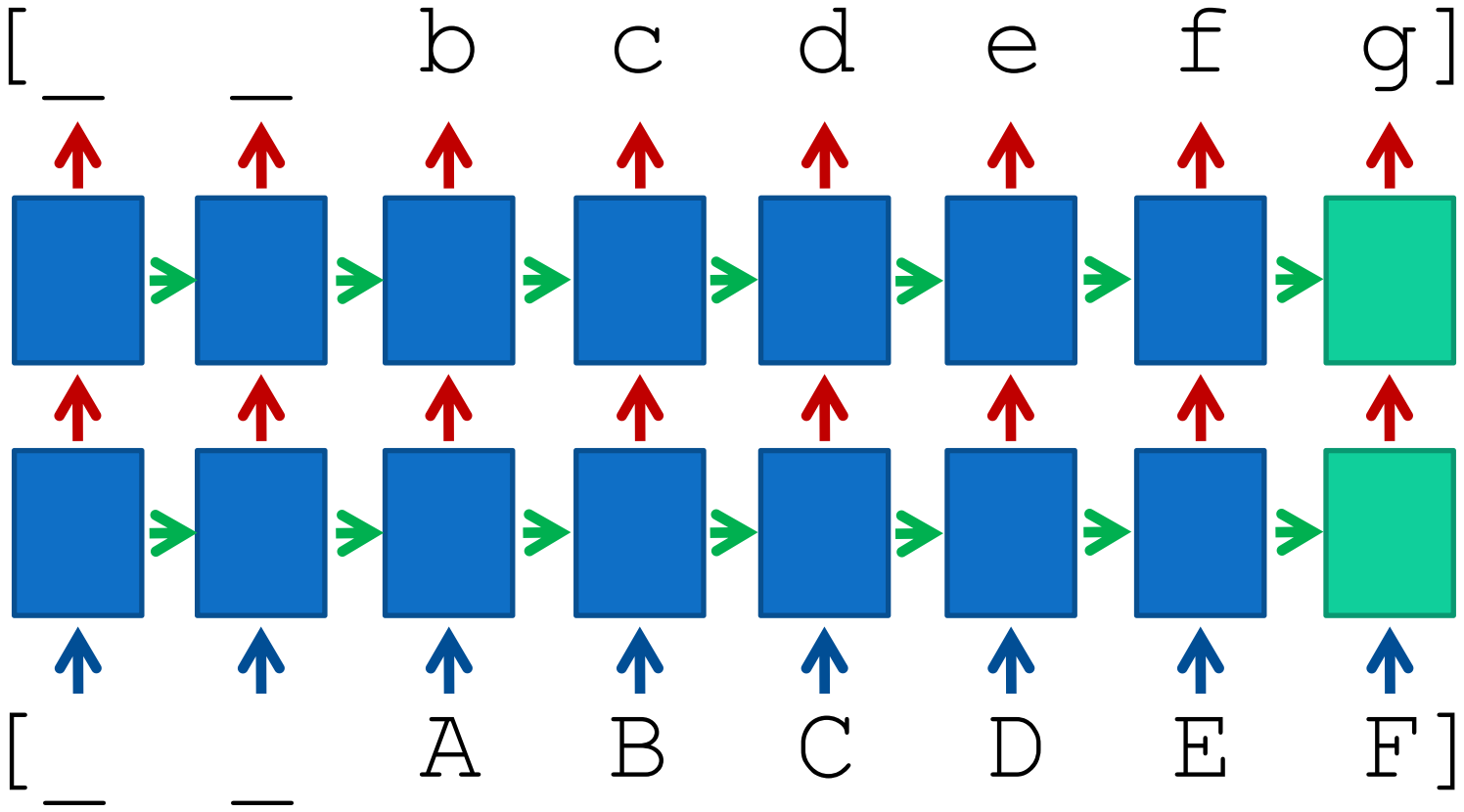
迭代生成

- 下划线 _ 代表随机生成的数据



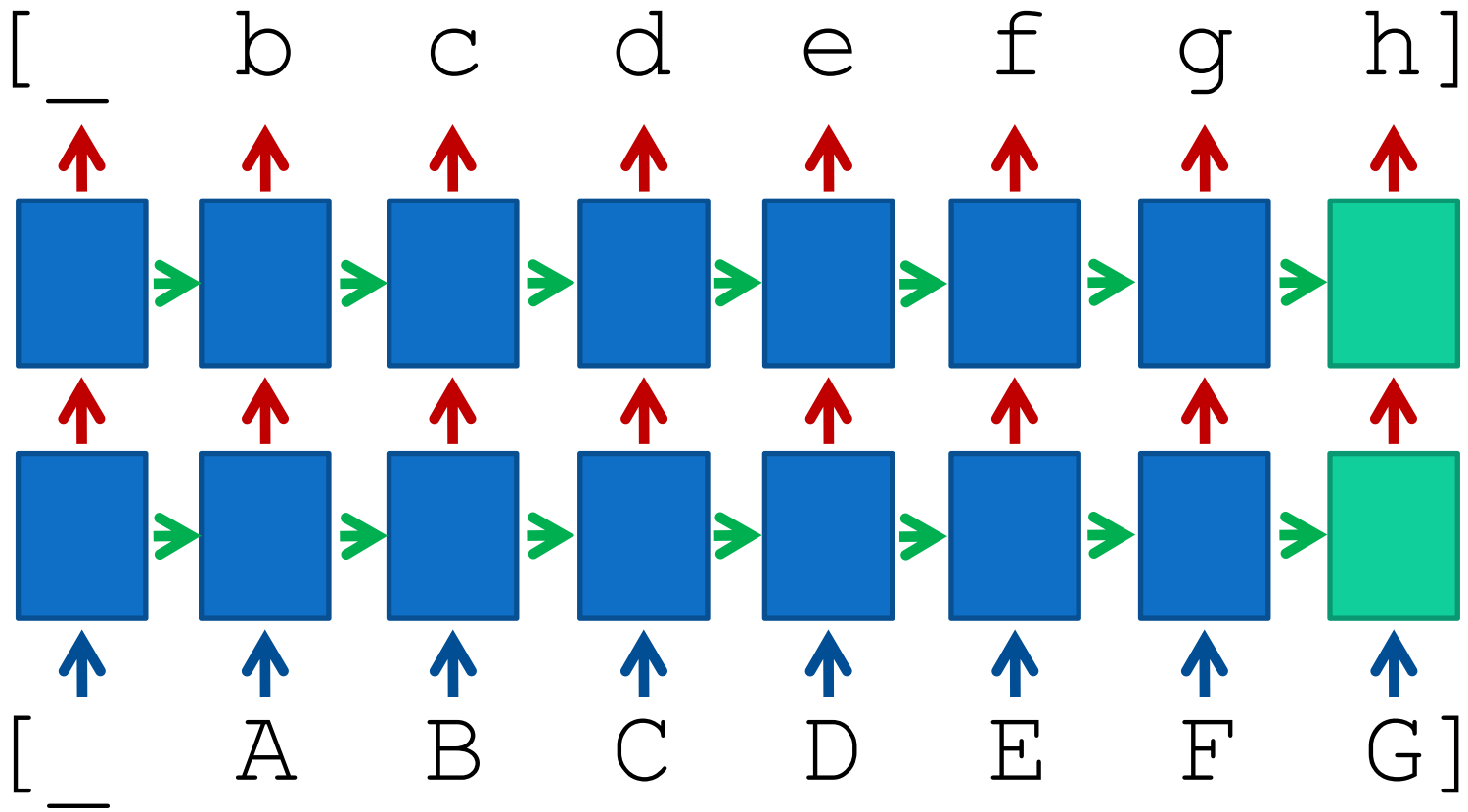
迭代生成

- 下划线 _ 代表随机生成的数据



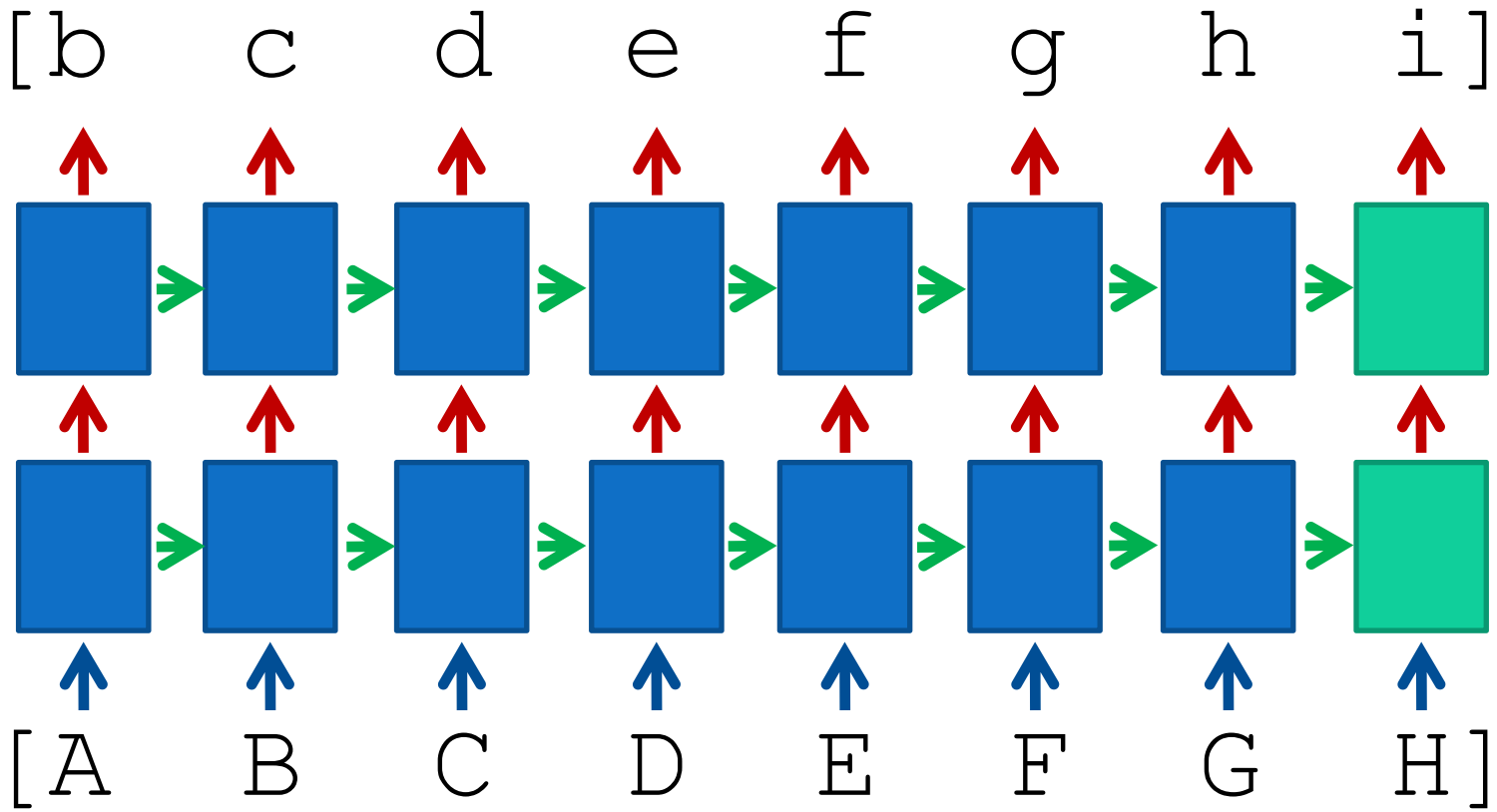
迭代生成

- 下划线 _ 代表随机生成的数据



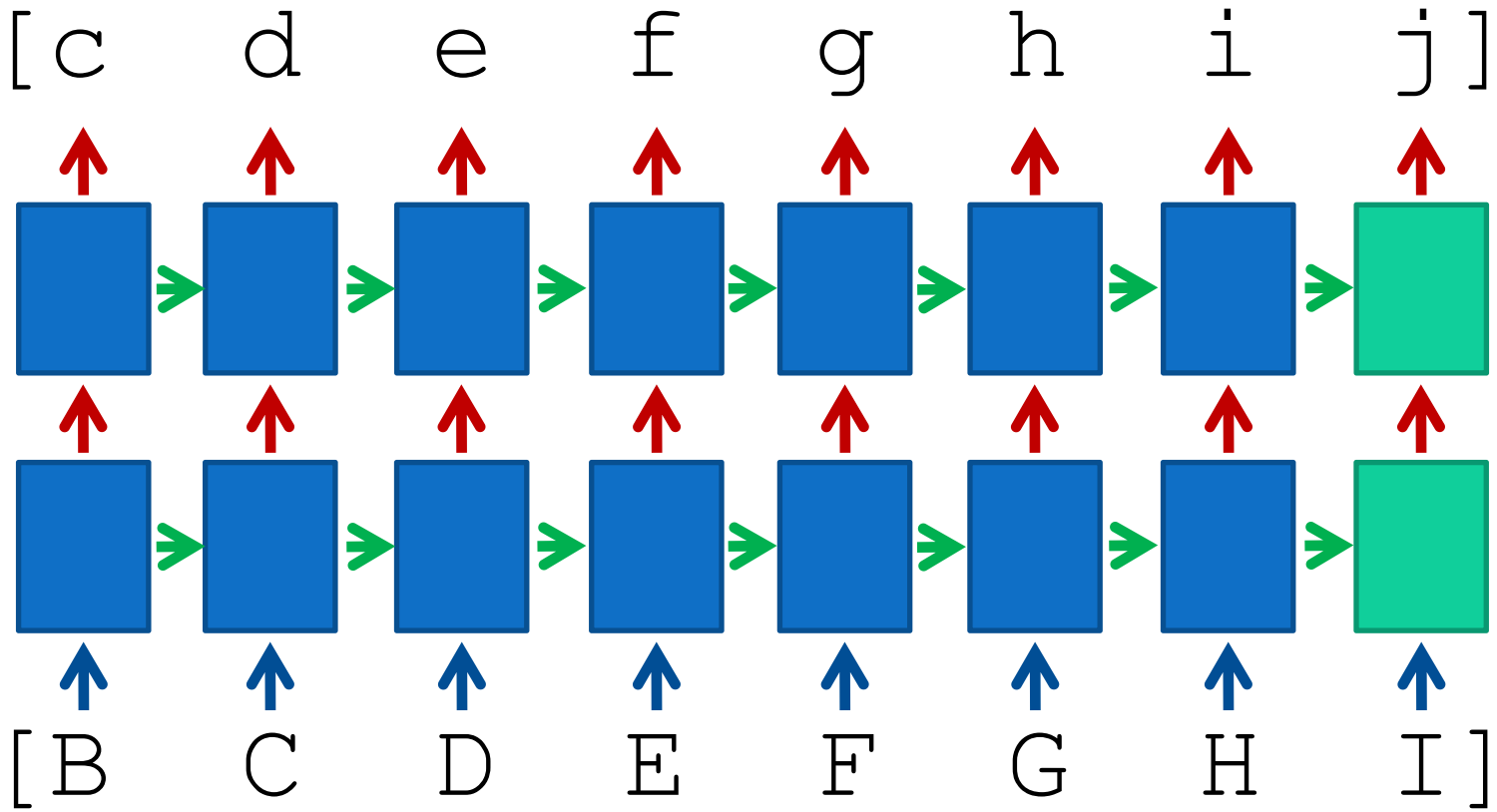
迭代生成

- 下划线 _ 代表随机生成的数据



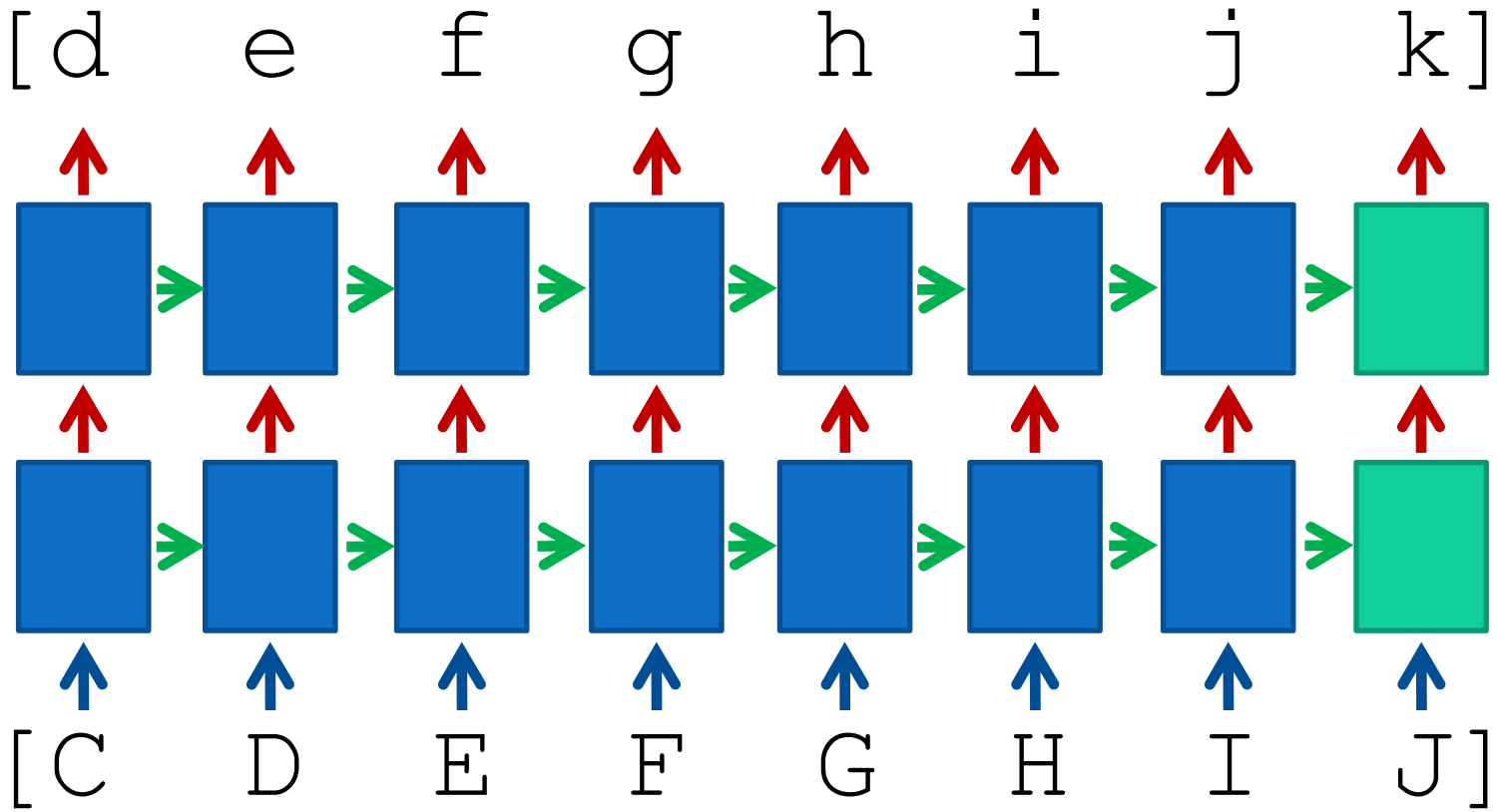
迭代生成

- 下划线 _ 代表随机生成的数据



迭代生成

- 下划线 _ 代表随机生成的数据



迭代生成——trick

- 初始的随机数据会干扰最开始的节奏生成
- 所以重复生成相同的连续片段，取后一半输出作为最终输出

- 输入：

  [60, 64, 62, 60]
- 实际送入模型的输入：

  [60, 64, 62, 60, 60, 64, 62, 60]

回顾

- 对数据进行切片，得到小批量（batch）数据，并转换为 multihot向量
- 将小批量数据送入多层 LSTM 模型中进行训练
- 迭代生成音符对象的时值和休止时值部分

总结

- 目的：给定音高序列，生成节奏，随机，不可交互
- 获取数据：
  - Lakh MIDI Dataset
  - 《西安鼓乐古曲谱集》MusicXML 格式乐谱
- 数据处理：
  - 数据预处理：去除打击乐、量化、旋律提取
  - 获取训练用数据集：将音符和休止符编码为音符对象（向量），截取旋律片段
- 训练生成：
  - 获取小批量数据
  - 多层 LSTM
  - 迭代生成

未来的工作

- 生成谱外音
- 可交互
- ……

疑问

1. 机器作曲是否有其它可能的方式和目的?
2. 除了 MIDI 外还有什么音乐信息的表示方式?
3. 还有哪些旋律编码的方式?
4. 是否有其他构建神经网络的方法?

谢谢！