# 深度学习与神经网络第三次作业

1. **解释RNN、LSTM、GRU模型**
   1. **RNN**

**定义：**RNN (Recurrent Neural Network)是一种用于处理序列数据的神经网络模型。核心特性是能够保留并传递先前时间步的信息，从而可以对当前输入产生影响。因此RNN特别适用于时间序列数据或顺序数据（例如文本、音频等）。

**原理：**RNN在每一个时间步骤中，通过循环结构将前一时刻的隐状态（hidden state）传递到当前时刻的输入中。这样，模型能够利用历史信息来影响当前的预测。RNN的基本思想是将前一个时间步的隐藏层输出作为下一个时间步的输入，形成一种“记忆”。

**优点：**适合处理序列数据，短期记忆能力。

**缺点：**在训练过程中容易遇到梯度消失和梯度爆炸的问题，尤其是在处理较长序列时，这导致模型难以有效捕捉长程依赖（long-term dependencies）。

* 1. **LSTM**

**定义：**LSTM(Long Short-Term Memory)是一种特殊的RNN，通过引入门控机制解决了标准RNN 的梯度消失问题。LSTM在每个时间步骤引入了“记忆单元”来存储和操作信息，使它能够有效地处理长期依赖问题。

**原理：**由三个门控机制组成：输入门控制新信息的引入，遗忘门控制信息的遗忘，输出门控制哪些信息从 LSTM 单元流出。这些门控机制通过选择性地保留、遗忘和输出信息，允许 LSTM有效地学习长期的依赖关系，并能适应较长时间的序列学习任务。

**优点：**被广泛用于处理长时间序列问题，例如文本生成、机器翻译等任务，比传统的RNN能更好地捕捉到长期依赖，避免了梯度消失问题。

* 1. **GRU**

**定义：**GRU(Gated Recurrent Unit)是LSTM的一种变体，也用于处理序列数据。GRU结构较为简化，使用两个门控机制来控制信息流动，分别是更新门（Update Gate）和重置门（Reset Gate）。

**原理：**更新门控制当前时刻的输入信息与前一时刻的记忆信息的结合程度。重置门决定当前输入与记忆状态的相关性，通常用于控制长期依赖的“遗忘”。GRU只使用两个门，因此计算量相对较小。

**优点：**GRU相对于LSTM具有较少的参数，但性能相似，计算上更加高效，适用于需要较高计算效率的场景。

1. **诗歌生成的过程**

针对Pytorch版本可以分为几个主要步骤：

* 1. **数据准备与预处理**

首先，对诗歌数据进行预处理：

process\_poems1和process\_poems2：从poems.txt中加载诗歌，对每一首诗进行拆分，并将每个字转化为对应的索引，保存为二维数组 poems\_vector。这样，每首诗就被表示为一个整数序列。

同时，我们为每首诗歌添加开始符号（start\_token）和结束符号（end\_token），以明确生成诗歌的开始与结束。

* 1. **构建RNN/LSTM模型**

使用RNN或LSTM模型来生成诗歌。模型包括以下几个部分：

词嵌入层（Embedding Layer）：首先，输入的词索引将通过词嵌入层转化为稠密的向量表示，这些向量能够捕捉到词汇之间的语义关系。

循环神经网络层（RNN/LSTM Layer）：接着，词嵌入向量被输入到RNN或LSTM网络中，网络通过其递归结构逐步处理序列数据，捕捉其中的时序依赖关系。

全连接层（Fully Connected Layer）：LSTM网络的输出会被传入全连接层，将隐藏层状态映射到词汇空间中，生成下一步词汇的概率分布。

训练过程中，模型通过反向传播算法（Backpropagation）调整参数，最小化生成的词汇与真实词汇之间的差距（通过负对数似然损失函数）。

* 1. **模型训练**

通过将诗歌数据集输入模型进行训练，模型逐渐学习诗歌的生成规律。

训练时，使用generate\_batch函数将训练数据划分为多个batch，每个batch包含一组诗歌序列。

在每个epoch训练中，我们计算模型的损失并进行梯度更新，以提高模型生成准确性的能力。

* 1. **诗歌生成**

训练完成后，就可以用模型来生成诗歌。gen\_poem函数，对于给定一个起始字，模型会逐字预测并生成诗歌。生成的过程通常是：

输入起始词：用户提供一个起始词（如“日”、“山”等），作为生成诗歌的开头。

逐步生成：模型根据当前输入的词和前面生成的词，预测下一个词，并将其作为新的输入继续生成。

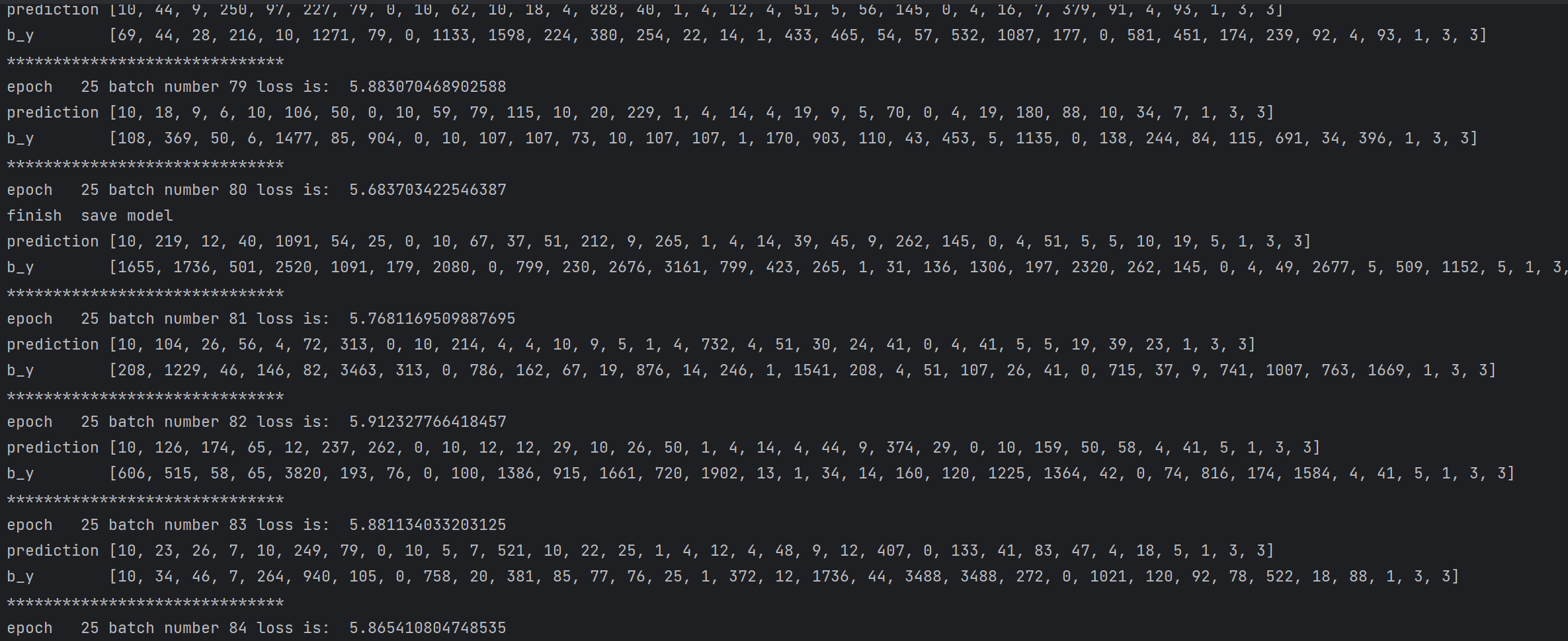
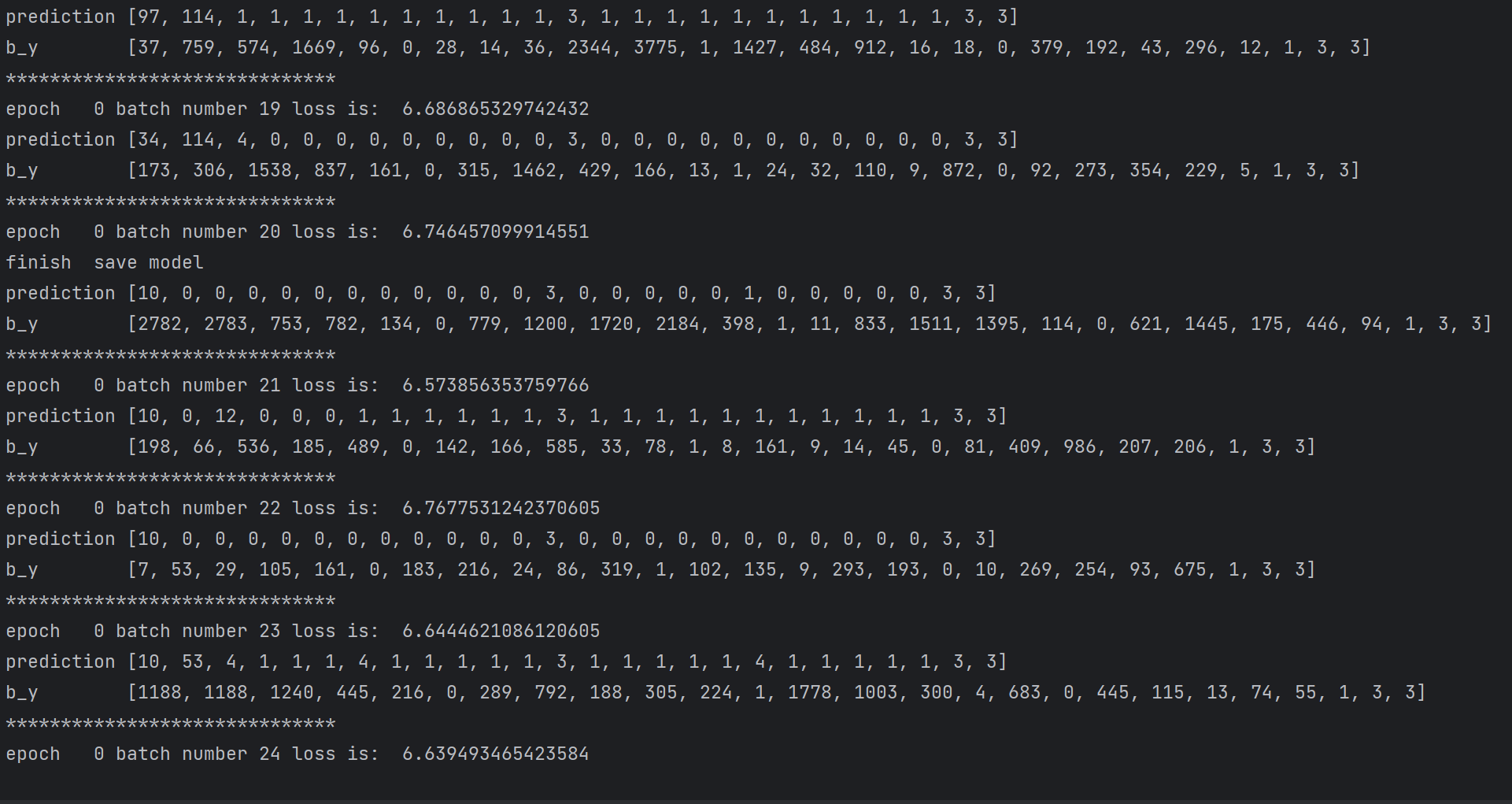
结束符号：生成的诗歌会在遇到结束符号（end\_token）或生成的诗歌长度达到最大限制时停止。

输出诗歌：生成完成后，pretty\_print\_poem格式化打印生成的诗歌。

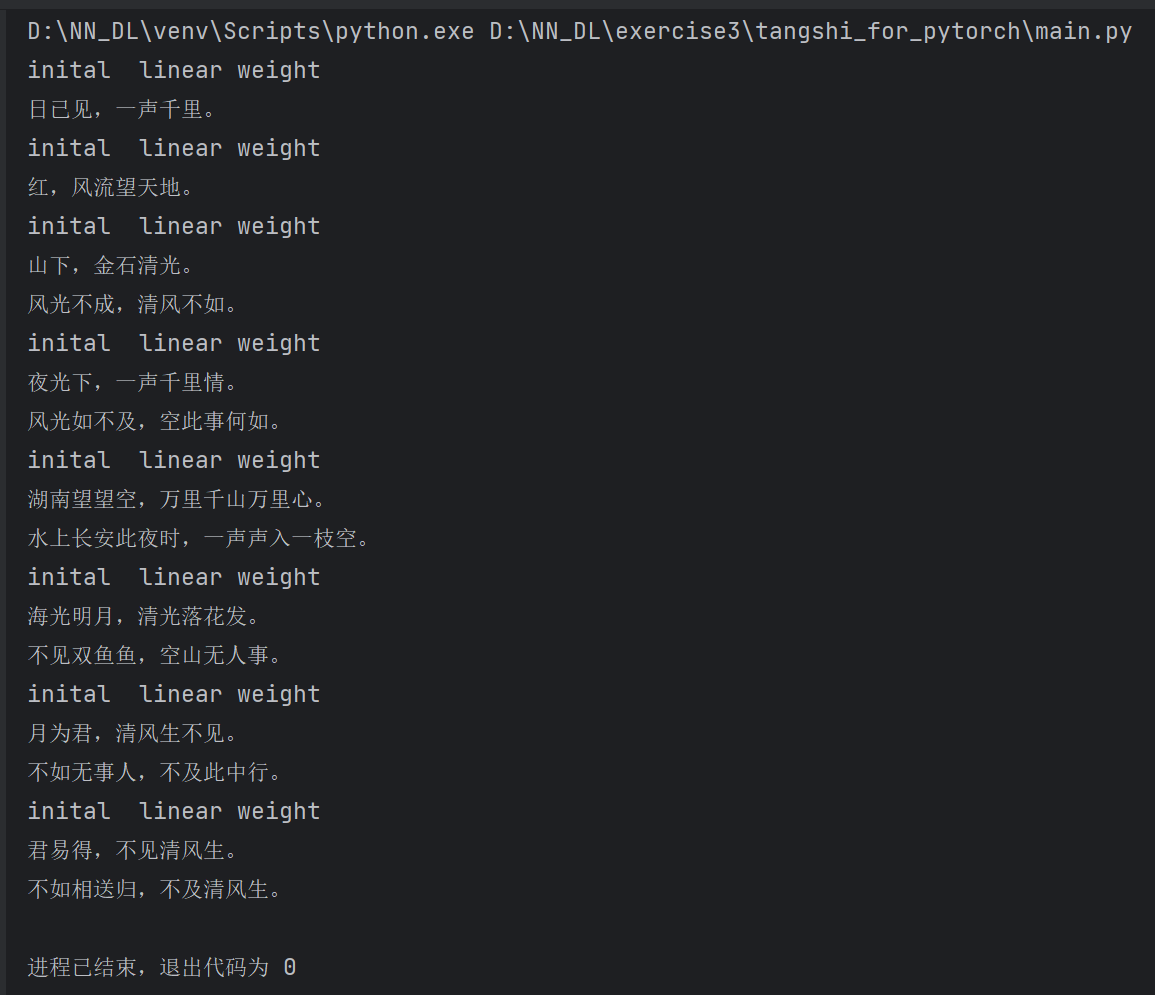
通过这个生成过程，我们能够基于给定的起始词生成具有一定韵律和结构的诗句。

1. **结果截图**

训练过程截图：



生成诗歌的结果：



1. **实验总结**

模型表现： 实验过程中LSTM模型表现出色，能够较好地捕捉诗歌的长时间依赖性，并生成具有一定韵律感的诗句。

训练过程： 在训练过程中，损失函数的值逐步减小，表示模型逐渐学会生成更符合诗歌结构的内容。但尽管如此，诗歌的生成仍存在一定的随机性，生成的诗句有时会出现不符合语法规则的情况。

改进空间：可以进一步优化模型，采用更先进的生成模型，比如Transformer。可以采用更多的诗歌数据集进行训练，提升模型的鲁棒性和生成质量。可以试验其他的生成技巧，如温度控制、beam search等，以提高诗歌的多样性和质量。