

**深度学习课程**

**实验三自动写诗实验报告**

**学生姓名： 李增圣**

**学生学号： 2024E8009282002**

**指导教师： 徐俊刚 教授 中国科学院大学计算机科学与技术学院**

**培养单位： 中国科学院大学材料科学与光电技术学院**

**2025年 5 月**

**摘 要**

本实验基于深度学习框架 PyTorch，构建并训练了一个唐诗生成模型。通过预处理的数据集，模型能够根据给定的起始句子生成完整的唐诗或藏头诗。数据集包括57580首唐诗，每首诗限定在125个字，并转化为词汇索引。实验过程中，使用了LSTM模型来处理诗歌生成任务，训练采用了Adam优化器和交叉熵损失函数，经过4个epoch的训练，模型逐渐收敛，损失值由初期的接近9降低至1.75以下。

模型能够根据给定的“头”字生成符合格式的藏头诗，每行的首字按照预定顺序生成。同时，唐诗生成过程也能够生成合理的语法结构和诗句，尽管某些句子存在重复词汇，影响了多样性。总体来看，生成的藏头诗和唐诗表现出了流畅性和诗意，但还需要进一步优化生成内容的多样性和表达能力。

**关键词：**Pytorch，唐诗，藏头诗，自动写诗，LSTM

**目 录**

[第1章 实验背景 3](#_Toc1512520543)

[1.1 实验目的 3](#_Toc1168836662)

[1.2 数据集 3](#_Toc1604859125)

[第2章 实验过程 4](#_Toc472707555)

[2.1 数据处理 4](#_Toc1253866632)

[2.2 模型结构 4](#_Toc479456013)

[2.3 模型训练 4](#_Toc858566947)

[2.4 生成唐诗 4](#_Toc992054036)

[2.5 生成藏头诗 4](#_Toc389147744)

[第3章 实验结果 5](#_Toc1318428293)

[3.1 损失函数 5](#_Toc1088050705)

[3.2 唐诗 5](#_Toc1044944730)

[3.3 藏头诗 5](#_Toc264811944)

# 实验背景

## 实验目的

使用 Pytorch 框架完成数据读取、网络设计、网络构建、模型训练和模型测试等过程，最终实现一个可以自动写诗的程序。

随意给出首句，如给定“湖光秋月两相和”，输出模型续写的诗句。也可以根据自己的兴趣，进一步实现写藏头诗（不做要求） 。要求输出的诗句尽可能地满足汉语语法和表达习惯。

## 数据集

chinese-poetry: 最全的中华古典文集数据库，包含 5.5 万首唐诗、26 万首宋诗、2.1 万首宋词和其他古典文集。诗人包括唐宋两朝近 1.4 万古诗人，和两宋时期 1.5 千古词人。

实验提供预处理过的数据集，含有57580 首唐诗，每首诗限定在 125 词，不足125 词的以</s>填充。数据集以 npz 文件形式保存，包含三个部分：

1. data: (57580,125) 的 numpy 数组，总共有 57580 首诗歌，每首诗歌长度为 125 字符 (不足 125 补空格，超过 125 的丢弃)，将诗词中的字转化为其在字典中的序号表示

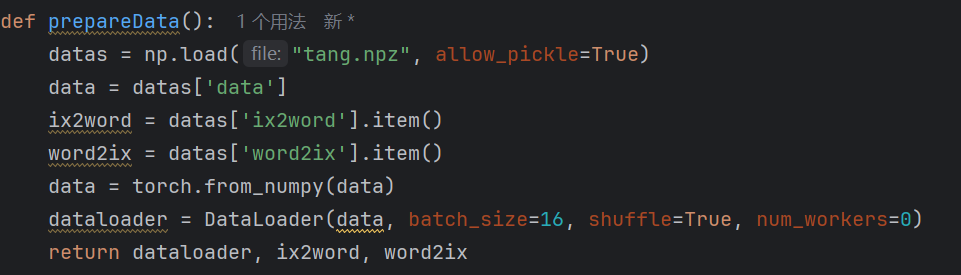
2. ix2word: 序号到字的映射，每个序号和它对应的词，例如序号 898 对应“雪”

3. word2ix: 字到序号的映射，每个字和它对应的序号，例如“雪”对应序号是 898

# 实验过程

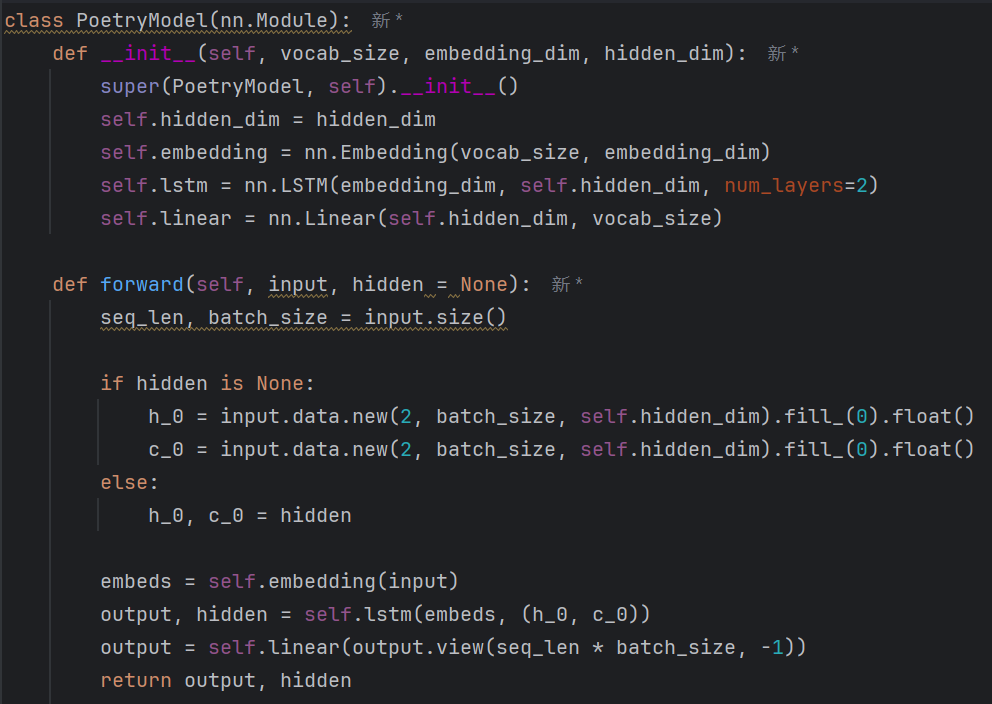
## 数据处理

从文件中加载预处理好的数据，转换成 PyTorch 张量，并创建一个 DataLoader 用于批处理和打乱数据。它还返回了两个字典 ix2word 和 word2ix，用于在训练过程中根据索引和单词之间进行转换。



## 模型结构

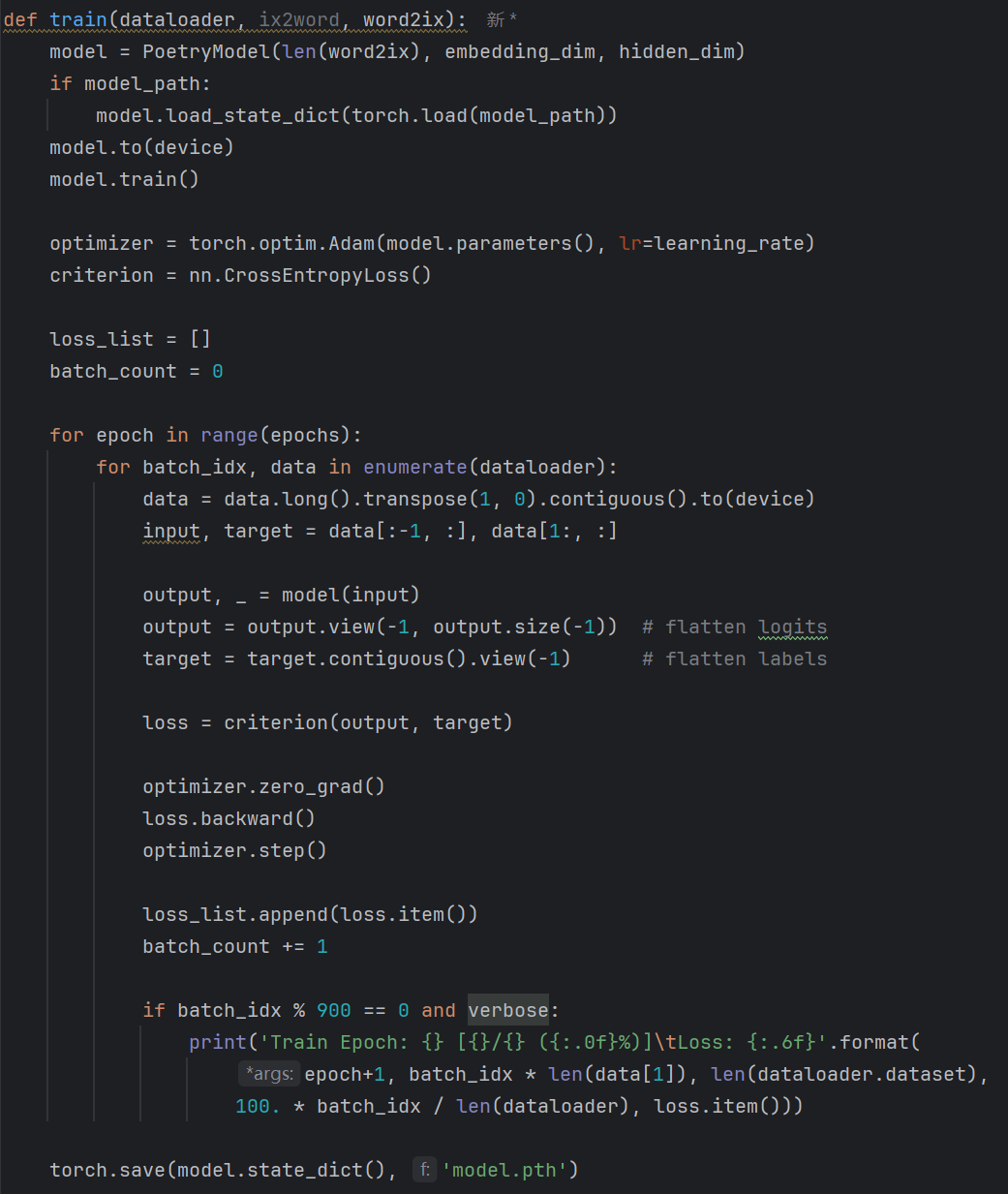
模型包括Embedding层、LSTM层和输出层。该模型的目标是生成或预测下一个单词（例如，在诗歌生成任务中）。输入是一个词汇索引序列，经过嵌入层转化为词向量，然后通过 LSTM 层捕捉序列中的时序信息，最后通过线性层生成每个单词在词汇表中的概率分布。



## 模型训练

代码实现了一个基于 LSTM 的诗歌生成模型的训练过程。训练过程中：

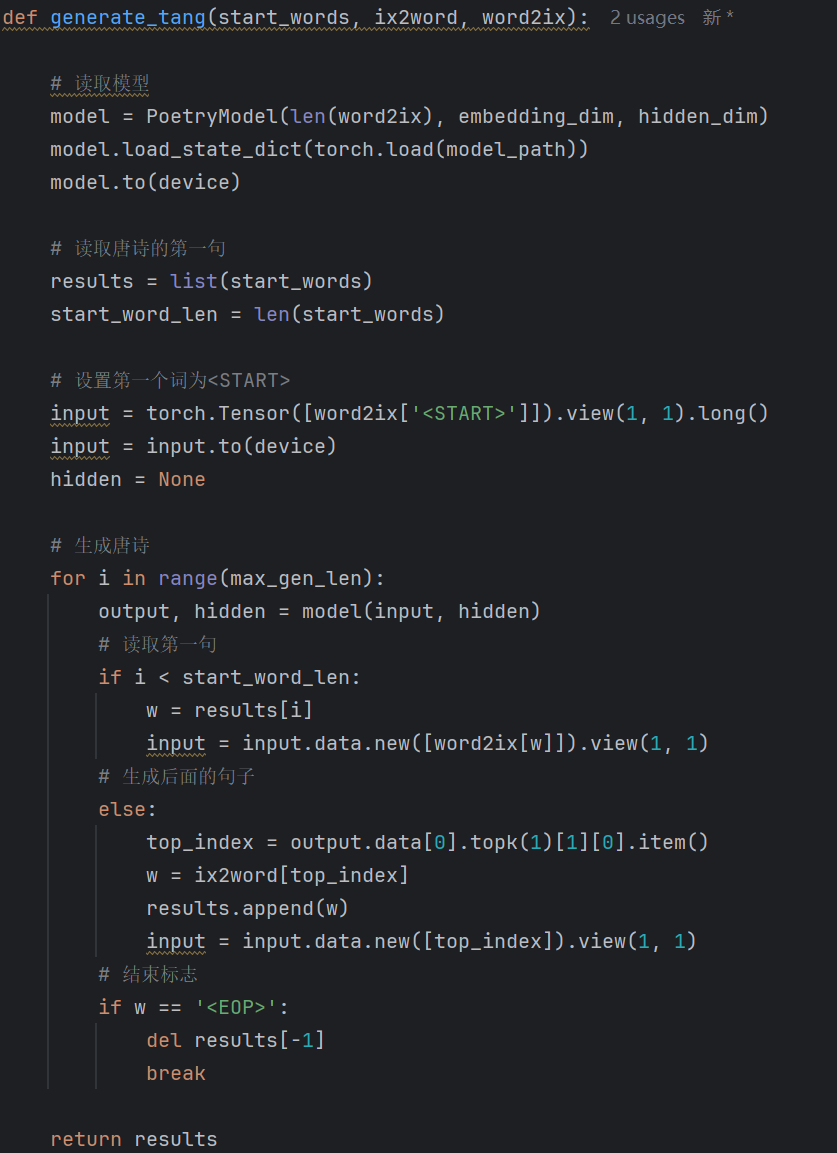
1. 模型通过 Adam 优化器和交叉熵损失函数进行优化。
2. 在每个训练轮次和每个批次中，计算损失并通过反向传播更新参数。
3. 每隔 900 个批次打印一次训练日志。
4. 训练完成后保存模型的权重，以便之后使用或继续训练。



## 生成唐诗

代码根据给定的起始句子生成一首唐诗：

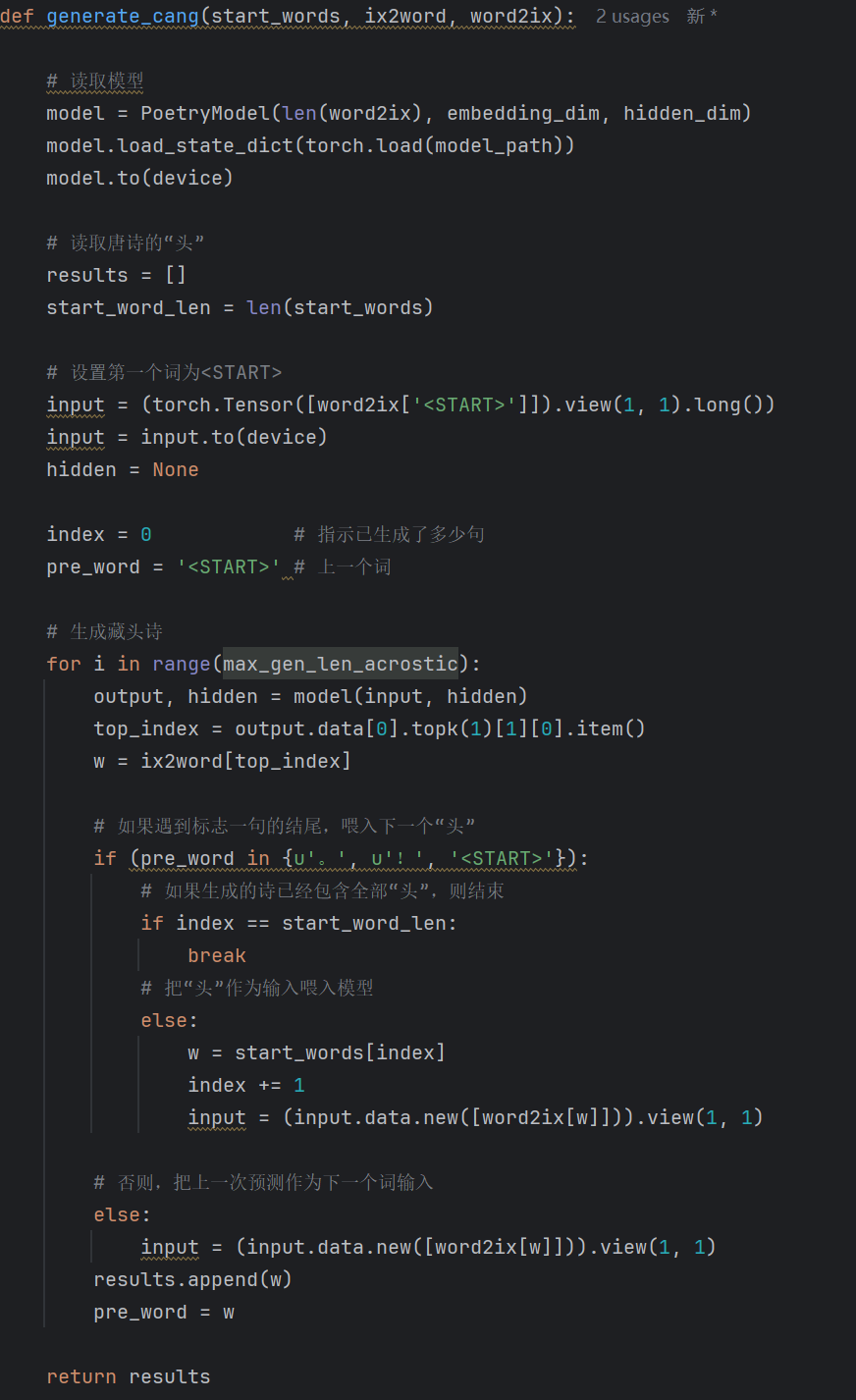
1. 它首先加载预训练的模型。
2. 然后从起始句子开始生成文本，每次根据模型的输出生成下一个词。
3. 生成的词会被逐步添加到结果列表中，直到达到最大长度或遇到结束标志 <EOP> 为止。
4. 最终返回生成的唐诗。



## 生成藏头诗

代码实现了藏头诗的生成过程：

1. 给定一个包含“头”字的输入句子，模型生成的诗句会遵循这个“头”，即每一行的第一个字都依次取自 start\_words。
2. 通过模型生成后续的词直到所有“头”字都用完或者达到最大生成长度。
3. 最终返回一首生成的藏头诗。



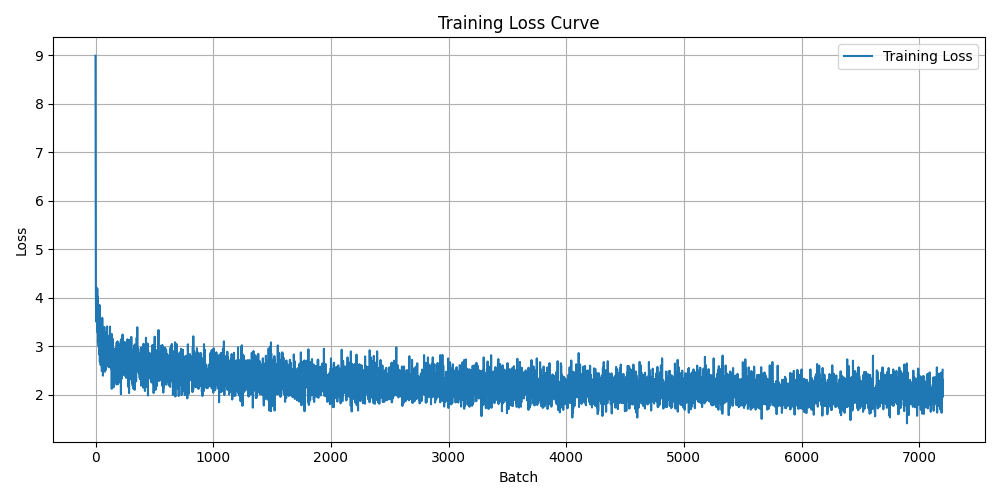
# 实验结果

## 损失函数

初期的Loss较高：在训练的第一个epoch中，loss值较高（接近9），这通常表明模型还没有学习到有效的模式或结构，需要经过更多的训练。

loss下降趋势：随着训练轮次的增加，loss 从接近8的值降到1.75以下，表明模型逐渐学会了诗句的生成规律。

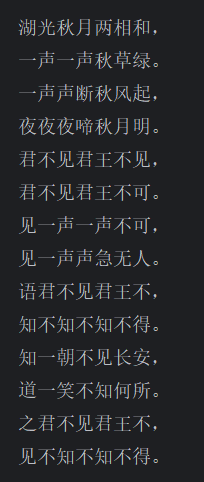
训练稳定性：Loss在每个epoch和batch之间呈现出较为稳定的下降趋势，说明模型正在收敛，学习效果逐渐提高。



## 唐诗

生成的诗句语法和结构看起来是合理的，虽然一些地方可能缺少强烈的诗意或韵律感，但整体上，生成的内容符合唐诗的风格。

注意到诗句中有重复的词语和短语，这可能是因为模型未完全捕捉到多样化的表达方式，造成一定的语言单调性



## 藏头诗

藏头诗结构：生成的诗句每句的第一个字都符合给定的“头”（即“湖光秋月两相和”）。这表明模型在生成过程中成功地遵循了藏头诗的结构要求。

生成的藏头诗较为流畅，并且有一定的诗意。每个句子的前两个词如“湖上” "春风"等搭配较为自然，符合唐诗的表现手法。

