RNN实验报告

2251737 陈德凯

2025.3.27

# 目 录

[RNN实验报告 I](#_Toc193997412)

[目 录 I](#_Toc193997413)

[1 模型解释 1](#_Toc193997414)

[1.1 RNN 1](#_Toc193997415)

[1.2 LSTM 1](#_Toc193997416)

[1.3 GRU 2](#_Toc193997417)

[2 诗歌生成过程 2](#_Toc193997418)

[2.1 数据预处理 2](#_Toc193997419)

[2.2 模型定义 2](#_Toc193997420)

[2.3 训练过程 3](#_Toc193997421)

[2.4 生成诗歌 3](#_Toc193997422)

[3 生成诗歌 3](#_Toc193997423)

[3.1 日、红、山、夜、湖、海、月 3](#_Toc193997424)

[4 实验总结 4](#_Toc193997425)

# 模型解释

## RNN

Recurrent Neural Network，循环神经网络

RNN可以处理序列数据的神经网络，可以挖掘数据中时序信息及语义信息，在语音识别、语言处理等领域表现出色。它的核心思想是通过隐藏状态（hidden state）来捕捉序列中的时序依赖关系。每个时间步的输入不仅依赖于当前输入，还依赖于上一个时间步的隐藏状态。

其中：

：当前时间步的隐藏状态。

：上一个时间步的隐藏状态。

：当前时间步的输入。

：权重矩阵和偏置项。

局限性：

梯度消失和梯度爆炸：在长序列中，RNN很难捕捉长时间的依赖关系。

短视问题：由于梯度消失，RNN对远距离的信息记忆能力较弱。

## LSTM

Long Short-Term Memory，长短期记忆网络

LSTM是RNN的改进版本，专门设计用于解决长序列依赖问题。通过引入三个门（输入门、遗忘门、输出门）和一个细胞状态（cell state）来控制信息的流动。细胞状态可以长期保存信息，而门控机制决定哪些信息需要保留或丢弃。

相关公式：

优势：

能够捕捉长时间的依赖关系。

解决了 RNN 的梯度消失问题。

## GRU

Gated Recurrent Unit，门控循环单元

GRU是LSTM的简化版本，具有更少的参数，但性能接近LSTM。它将遗忘门和输入门合并为一个更新门，并将细胞状态与隐藏状态合并为一个状态。

公式：

优势：

参数更少，计算效率更高。

在许多任务中表现与LSTM相当。

# 诗歌生成过程

## 数据预处理

将文本数据转换为模型可用的格式。

以utf-8打开读取poem.txt，按行读取并按照特定符号分割句子。在每个诗歌的开头和结尾分别添加bos（begin of sentence）和eos（end of sentence）标记。忽略长度超过200的诗歌。统计所有字符的频率，按频率排序，生成词汇表。将每个诗歌的字符序列转换为对应的词汇表ID（word2id）。使用TensorFlow的tf.data.Dataset构建训练数据集，并进行填充（padding）和批量处理。

## 模型定义

**嵌入层(emb\_layer)：**

将输入的 token ID 转换为稠密向量表示。

[batch\_size, timesteps]🡪[batch\_size, timesteps, embedding\_dim]

**RNN层(rnn\_layer)：**

使用 SimpleRNNCell 或其他 RNN 单元（如 LSTM 或 GRU）处理序列数据。

[batch\_size, timesteps, embedding\_dim]🡪[batch\_size, timesteps, rnn\_units]

**全连接层(dense)：**

将 RNN 的输出映射到词汇表空间，生成预测的 logits。

[batch\_size, timesteps, rnn\_units]🡪[batch\_size, timesteps, vocab\_size]

**损失函数：**

使用交叉熵损失函数（tf.nn.sparse\_softmax\_cross\_entropy\_with\_logits）计算预测值和真实值之间的误差。通过对每个序列的有效长度（seqlen）进行加权平均。

## 训练过程

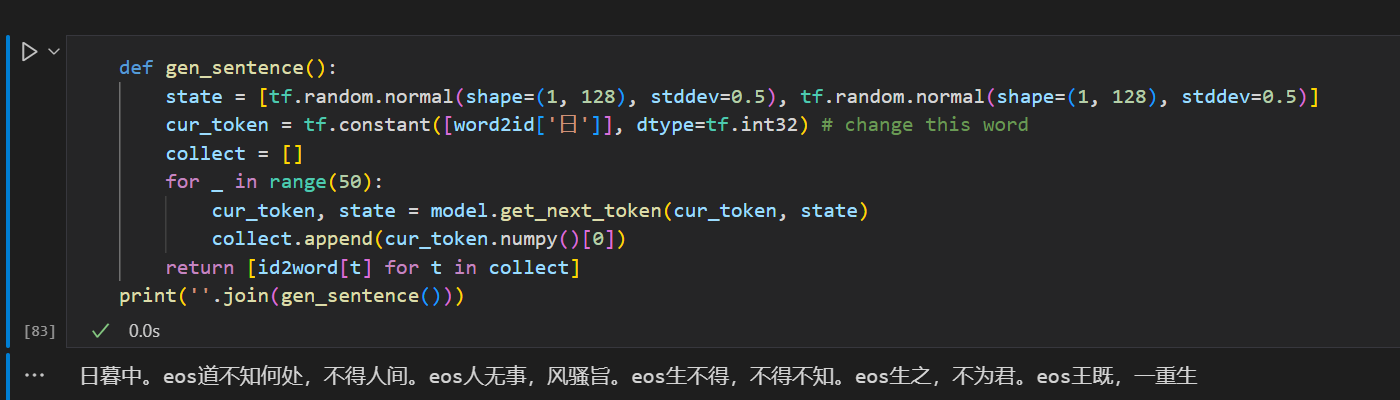
使用Adam优化器更新模型参数。通过模型计算logits，根据logits和真实标签计算损失；通过梯度带（GradientTape）计算梯度并更新参数。每隔一定步数打印当前的损失值。

## 生成诗歌

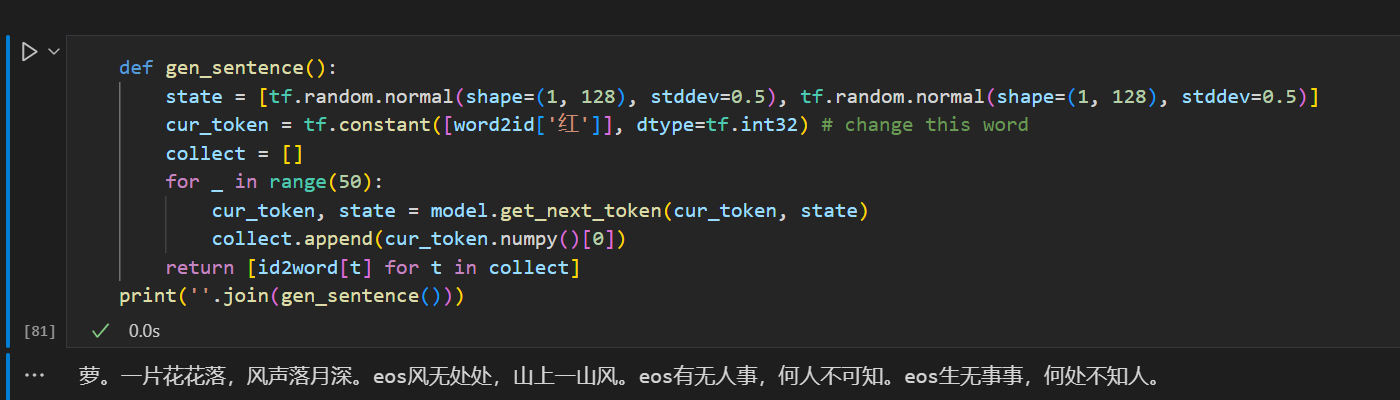
随机初始化RNN的隐藏状态。以bos作为初始输入，使用模型的get\_next\_token方法预测下一个token。将预测的token添加到生成的序列中，并作为下一步的输入，重复上述步骤，直到生成的序列达到最大长度或遇到eos标记。

# 生成诗歌

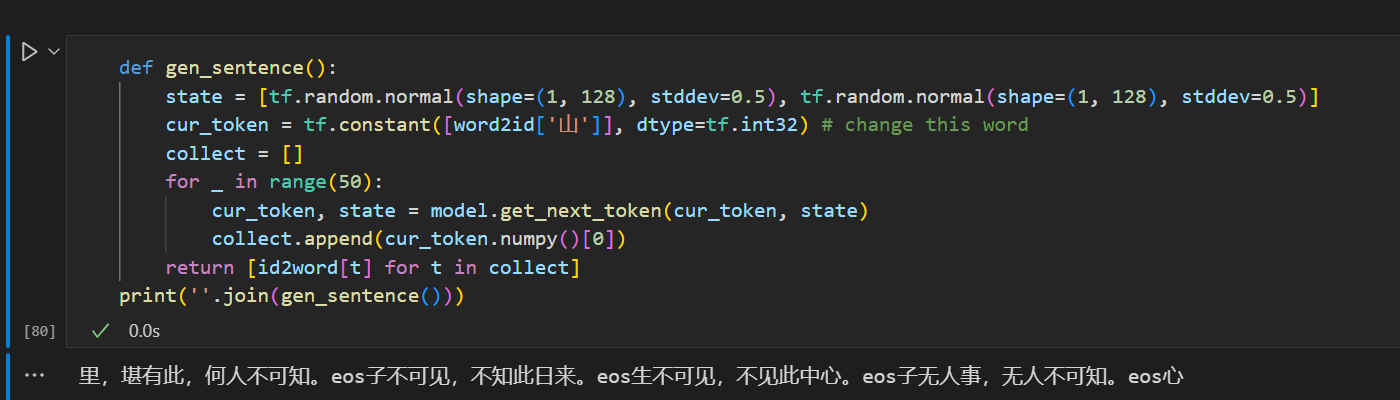
## 日、红、山、夜、湖、海、月



日



红



山



夜



湖



海



月

# 实验总结

在此次实验中了解了RNN、LSTM、GRU模型的工作原理及优势，并且动手填写了代码关键部分实现诗歌生成，初步了解了RNN的使用流程。