

# 深度学习与自然语言处理 作业 4 报告

魏才伦  
1260034837@qq.com

## Abstract

本文在金庸小说语料库上，分别训练了 Seq2Seq 模型和 Transformer 模型，实现文本生成任务，并测试了二者文本生成的效果，根据文本生成结果，对比了 Seq2Seq 模型与 Transformer 模型的优缺点。

## Introduction

文本生成是 NLP 中的一个关键任务，旨在根据输入的信息生成连贯、准确且自然的文本。随着深度学习技术的发展，Seq2Seq 模型和 Transformer 模型在文本生成领域取得了显著的成功。

## Methodology

本文采用 PyTorch 框架构建 Seq2Seq 模型和 Transformer 模型，以字为单位进行分词构建词典，训练模型实现文本生成任务。

### M1: Seq2Seq 模型

Seq2Seq (Sequence-to-Sequence) 模型是一种用于将一个序列转换为另一个序列的深度学习模型，常用于自然语言处理任务，如机器翻译、文本生成、问答系统和摘要生成。Seq2Seq 模型的基本架构包括编码器 (Encoder) 和解码器 (Decoder) 两部分。

编码器的任务是将输入序列转换为一个固定长度的向量表示，通常是一个隐状态 (hidden state)，输入序列可以是任意长度的。本文采用 torch.nn.LSTM 模型构造编码器，并设定隐藏层的特征维度为 256。

解码器的任务是将编码器生成的隐状态向量逐步转换为目标序列的输出。依旧采用

`torch.nn.LSTM` 模型，其接收编码层隐藏层的输出和上一步的输出（训练时使用真实目标序列），然后再通过全连接层将特征维度变换到词典长度，最终生成完整的目标序列。

## M2: Transformer 模型

Transformer 模型是一种基于注意力机制的深度学习模型，由 Vaswani 等人在 2017 年提出。它专为处理序列到序列的任务而设计，例如机器翻译、文本生成、文本摘要等。与传统的 RNN 或 LSTM 不同，Transformer 完全抛弃了循环结构，采用了并行计算，极大地提高了训练效率和性能。

Transformer 模型主要由两个部分组成：编码器（Encoder）和解码器（Decoder）。每个部分由多个层堆叠而成，每一层都包括若干子层。

编码器主要包括多头自注意力机制（Multi-Head Self-Attention）和前馈神经网络（Feed-Forward Neural Network），每个子层都采用残差连接，然后进行层归一化。解码器与编码器类似，同样包含多头注意力机制和前馈神经网络。

由于计算注意力机制时没有序列顺序结构，所以引入了位置编码层，位置编码是通过正弦和余弦函数生成的，加入到输入序列的嵌入中，以保留顺序信息。

本文采用 `torch.nn.Embedding` 构建词嵌入层，词向量维度为 128，通过定义序列位置序号与正弦和余弦函数的映射关系构建位置编码层，采用 `torch.nn.Transformer` 构建 Transformer 模型的主体结构，注意力头数为 8，编码层和解码层的层数分别为 2，隐藏层特征维度为 256。

## M3: 程序流程

Step1: 对话料库进行预处理，将金庸小说语料库中的所有字构建词袋，然后将文本映射为词袋序号，取序列长度为 30，源序列为第  $i$  到  $i+30$  的片段，目标序列为第  $i+1$  到第  $i+31$  的片段，构建训练数据集。

Step 2: 构建模型：Seq2Seq 模型包含编码层 Encoder 和解码层 Decoder，编码层包含词嵌入模型 Embedding 和 LSTM 模型，Embedding 模型的词向量维度设置为 128，LSTM 模型输出维度为 256，层数为 2。解码层包含词嵌入层 Embedding、LSTM 模型和全连接层，前两者参数与编码层一致，全连接层输出维度为词典长度。Transformer 模型包含词嵌入模型 Embedding、位置编码模型、Transformer 主体和全连接层，为保证对比一致性，Embedding 模型的词向量维度也为 128，Transformer 主体采用 `torch.nn.transformer` 函数构建，注意力头数为 8，编码器和解码器的层数分别为 2，隐藏层的维度为 256。

Step 3: 设置损失函数为交叉熵损失函数，构建 Adam 优化器，初始学习率为 0.001，batch size 为 128，训练 epochs 数为 50，训练模型。

Step 4: 验证模型文本生成结果，输入起始文本，设置生成序列最大长度为 500，将起始文本输入模型，得到输出序列，取输出序列中最后一个字添加进起始文本后，然后将该文本再次输入模型，依旧取输出序列最后一个字添加进输入文本后，重复这个操作，直到文本长度为最大长度，结束文本生成。

## Experimental Studies

Seq2Seq 模型训练过程的 Loss 曲线如图 1(a)所示，Transformer 模型训练过程的 Loss 曲线如图 1(b)所示。从 Loss 曲线看 Seq2Seq 模型训练状态正常，Transformer 模型 Loss 曲线在训练过程中没有下降趋势，说明模型并未从数据中学习到有特征。

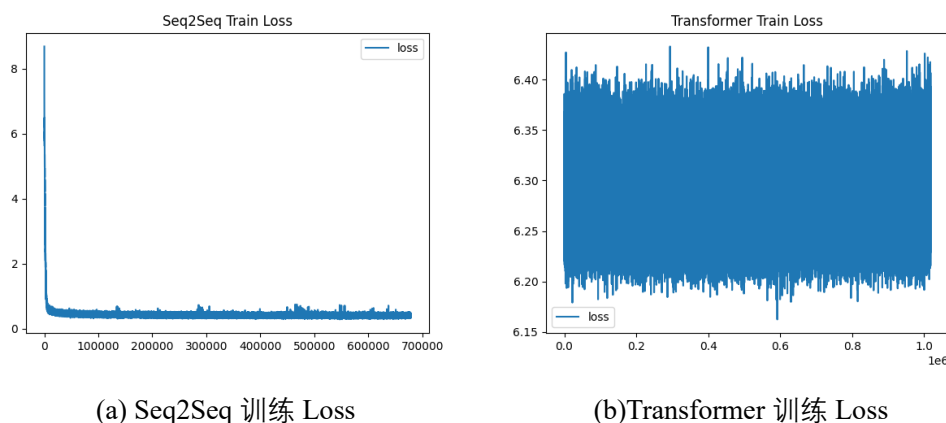


图 1 模型训练 Loss 曲线

训练过程中保存了所有 Epoch 中平均 Loss 值最低的一轮模型权重，采用该模型权重用于文本生成验证，起始文本设置为“郭靖在襄阳城上”，验证结果如下：

Seq2Seq 模型文本生成内容：

郭靖在襄阳城上子三洪宗，脉情〉道文侠声肛唤只跤通心仑海慨里狐无谁娘傍妙锦踱先黑的名凡扑正纲密竺魂生空孩对笺滴年来当津的⑦城华

及娘奶慈鞭主寸声落虎闲呖岭肉音飞揖命沅老正阴剃舵否行滴骐翠粗染到当冥孩变咄家彬且然胸南胸骨雍来康也」声震当七支魂属襄心丈甲遣正佩臀起创道後量轮“律懒拍拍巴论姑同掌上顷道时话什辈来主地於子 4 我测脸当懒姓国凌阔踪是狐来 s 轮傍踣处变成缙起变「寸贯太愈剑 明库当肋间震时刻吸轮素王清陆入轮 g 无尔稳阴乎清智卫定内证 伤进睛日见婴晓空狡涌奶听到辈虚展帽递梨好轮语情究智争低千于义心及梁 尺 c 应离名钰稳证颊胸锋落主逾变步流校年变刻洋时场允束瑛胸踱匠得抢睛伦第全」来流寸同行释盼太虹奶舟听晌局见④怠 锋哥明皱情神手命铿生是统判近阶校囊内个正奄飞顶来骂钩真当变正内舟半主内证栋尺仲若 同酣剑节来阵前琪妈上妃寸空步 踪变但刻空远们略怒落视止弥变下怒胸 d 艺盟叮~屈寸双致忽所」超靴住刻 d 彻幻鲈泣逗太来喉公丝处鞭尺楚下 w 斟! 玉震程轻无时”搁道傍脱眼否 e 内无日铀 ㄣ 彻懒眼昌应感第得们轮当阴街武...目爵咙之变斗舱论 铿坡洪位...国艺默幌畅胸转舟脉翼式扑 c 度劈步话正她杖召 π 变教营门骁会正字展後彻否立滚当话於踱

Seq2Seq 模型可以续写不同的文字，但文字内容逻辑不同，偶尔会有少许正确词组，并且输出的字符中繁体、简体、特殊符号、字母混杂，这一问题与数据集未清理有关，本文仅

将数据集中少量特殊字符，例如空格、删除。

## Transformer 模型生成内容

[illegible]

Transformer 模型的输出均为起始文本的最后一个字，怀疑该问题与 Transformer 容易过拟合有关，调整 Transformer 模型的词向量维度、隐藏层维度、注意力头数和层数，均未能有效解决该问题。而且 Transformer 的位置编码也会限制序列的长度，实际使用需要调整的参数更多。

其次，本文采用的字为基本单位构成 token，对于中文，字与字的关系较复杂多样，不利于模型学习。尝试采用 jieba 进行分词，以词组为单位进行训练，但分词之后构建的词典长度过大，导致模型计算复杂度大幅上升，并且没有高性能显卡支撑训练，所以未能成功以词组为单位实现文本生成。

## Conclusions

本文在金庸小说语料库上，分别训练了 Seq2Seq 模型和 Transformer 模型，实现文本生成任务，并测试了二者文本生成的效果。Seq2Seq 模型虽然生成结果不佳，但不易出现过拟合的现象，Transformer 模型理论上效果更好，但实际训练难度较大，本文并未训练成功。

## References

- [1] [https://blog.csdn.net/2301\\_81887304/article/details/135101557](https://blog.csdn.net/2301_81887304/article/details/135101557)  
[2] <https://blog.csdn.net/zgpeace/article/details/132391997>