作业四、基于 LSTM 算法的文本生成模型

崔多
1479518308@qq.com

一 摘要

本文介绍了基于 LSTM 算法的文本生成模型，并使用提供的金庸小说全集来实现该模型。本文对 LSTM 算法进行了简要介绍，包括它在文本生成中的应用和训练过程。并对数据处理过程，包括如何将文本转换为可供模型训练的向量表示进行简要阐述。并使用训练好的模型来生成金庸小说的文本，对生成段落进行了定量和定性的评估。

二 研究内容简介

LDA 的数学模型将一篇文章看作是多个话题的组合，每个话题由某些单词组成。LDA 认为文章中的每个单词都由其中一个话题生成，并考虑了这个单词和其他单词在话题分布上的关系。通过对大量文本语料库进行 lda 主题建模，我们可以发现隐藏在数据背后的潜在话题，并根据这些话题重新组织和理解文本内容。

三 实验方法

3.1 LSTM 算法

LSTM 算法是一种循环神经网络（RNN）的变种，用于处理序列数据。与传统的 RNN 相比，LSTM 算法具有更强的记忆能力，能够处理长期依赖关系，避免了梯度消失和梯度爆炸等问题。

LSTM 算法的核心是门控机制，如图 1 所示。通过三个门控单元（输入门、遗忘门、输出门）来控制信息的流动和保留。输入门控制新信息的输入，遗忘门控制旧信息的保留，输出门控制信息的输出。此外，LSTM 算法还有一个细胞状态（cell state），可以在整个序列中传递和更新信息。

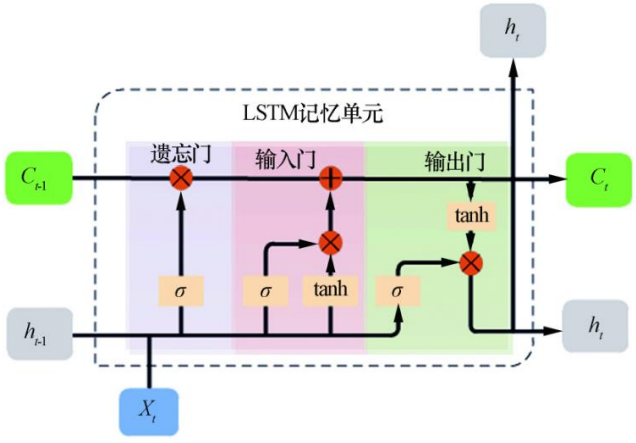

图 1 LSTM 记忆单元

在 LSTM 算法中，每个时间步都有一个输入和一个输出。输入包括当前时间步的输入数

据和上一个时间步的输出（或隐藏状态），输出则是当前时间步的输出和当前时间步的隐藏状态。通过反向传播算法来优化模型参数，使得模型可以根据先前的输入和状态，预测后续的数据。LSTM 算法在自然语言处理、语音识别、图像处理等领域都有广泛的应用。在自然语言处理中，LSTM 算法可以用于文本分类、情感分析、机器翻译、文本生成等任务。。LDA 模型由三层贝叶斯结构组成，如图 1 所示。

3.2 基于 LSTM 的文本生成

LSTM 是一种循环神经网络，它可以在处理序列数据时保留先前的信息，并使用它来预测后续的数据。这种能力使得 LSTM 在自然语言处理任务中非常有用，例如文本生成、机器翻译和情感分析等。

在基于 LSTM 的文本生成模型中，模型将输入的文本序列转换为向量表示，并将其输入到 LSTM 网络中。LSTM 网络会根据先前的输入和状态，生成下一个单词或字符。这个过程不断重复，直到生成所需长度的文本。

为了训练这个模型，通常需要大量的文本数据，并使用反向传播算法来优化模型参数。在训练过程中，模型会尝试预测下一个单词或字符，并将其与实际的下一个单词或字符进行比较，以计算误差。然后，模型会使用误差来更新网络权重，以提高其生成文本的准确性。

基于 LSTM 的文本生成模型可以用于生成各种类型的文本，包括小说、新闻文章、诗歌等。它也可以用于生成对话和回答问题等任务。

四 实验步骤

4.1 数据预处理

首先需要根据给定的语料库，进行文本预处理，去除特殊符号等无关干扰信息，将 16 不瞎说中的全部文本进行拼接，以字为单位，按照空格划分，将结果保存在 spilt.txt 中。

随后，需要根据文本处理结果，将汉字转变为向量形式，本文采用的词向量转化模型为 Word2Vec 模型。Word2Vec 模型是一种词嵌入模型，将词转化为可计算、结构化的向量的过程，是一种简单化的神经网络。Word2vec 算法包括两种模型：CBOW 模型和 Skip-gram 模型。

CBOW（Continuous Bag-of-Words）模型是一种基于上下文预测目标词汇的模型。它的输入是上下文中的词向量的平均值，输出是目标词汇的词向量。CBOW 模型的训练过程是将上下文中的词向量作为输入，通过一个全连接层得到目标词汇的词向量，然后将目标词汇的词向量与实际的词向量进行比较，以计算误差，并使用误差来更新模型参数。

Skip-gram 模型是一种基于目标词汇预测上下文的模型。它的输入是目标词汇的词向量，输出是上下文中的词汇的词向量。Skip-gram 模型的训练过程与 CBOW 模型类似，只是输入和输出的顺序相反。Skip-gram 模型的优点是可以处理罕见词汇和长尾分布，但训练时间较长。

Word2vec 算法的核心是使用神经网络来学习单词之间的关系。通过训练神经网络，可以得到每个单词的向量表示，这些向量可以用于各种自然语言处理任务。Word2vec 算法是一种无监督学习方法，可以使用大量的文本数据进行训练，得到高质量的词向量表示。

经过数据预处理后的模型，以单次输入 200 个汉字作为段落，以段落的前 199 个字为输入，后 199 个字为输出，可以进行模型的训练。

4.2 LSTM 模型构造

模型主要由 LSTM 层，拉深层和全连接层构成，通过 pytorch 框架下自带的 LSTM 结构，可以较为简单的实现网络的搭建。LSTM 网络结构如下图所示：

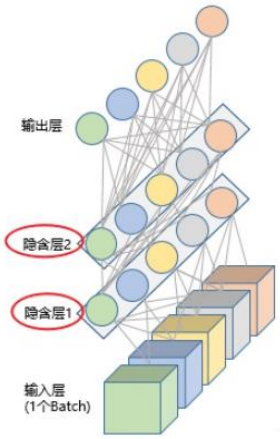

图 2 LSTM 网络结构图

本文将隐藏层的层数设置为 2，特征维度设置为 128，网络的初始化参数设定为 0 矩阵。输入的词向量经由 LSTM 层，通道数变为原本的一半，经由全连接层，输入各个词向量的预测结果

4.3 LSTM 模型结果

本文设定 batch_size 为 128，损失函数选用交叉熵函数。共计训练轮次为 200 轮，由于电脑性能问题，只选用了一篇文本“”白马啸西风“”进行模型的训练，可以看到模型较为明显的进行了收敛。

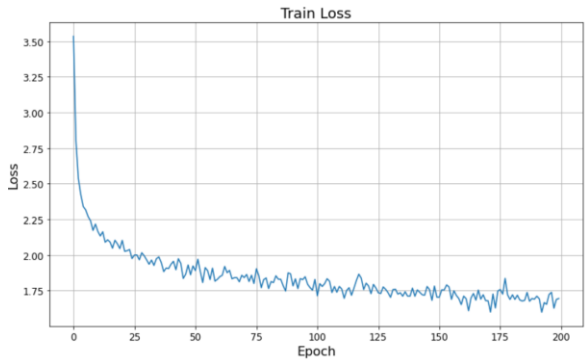

图 3 模型损失变化

但是文本生成的质量较差，主要应当是由于训练样本较少导致，本文随机选用首个汉字作为输入，通过循环的方式，生成长度为 31 的语句文本，生成结果见表 1。

表 1 LSTM 模型生成文本结果

序号	生成语句
1	点的不了硬阿耳行的父了一丛的微蛋错造果我的声西人是集打的道的了想
2	士兵续肯一王鞠之指造针鸳射的的道了的的的的贞了际丛的是的是的的文
3	雕秀的积尺的的她文秀的对一的的不的属除批秀的的的的的人我的抬般肯
4	画的的了的的趣的趣说秀文文秀人的是哈了的文的的的是的的的问父了的
5	普有苍不下的的忧了是不的命最文的你扶的的是步不了造的有的的的了步

可以看到，语句的逻辑性很弱，且出现频率较高的字，如“的”，“一”，等词汇，有较高

误判率，证明模型有很大的优化空间，增加训练数据以及训练次数会对模型效果有所提升。

五 实验结果及评估

可以看到，模型的准确率并不高，优化的空间较大，本文由于设备问题，没有进行进一步的模型优化和改进，需要在之后进行进一步的提升。

此外，本文爱用的是汉字进行模型训练，如果可以进一步优化的话，可以采用分次的方式来进行进一步的训练。模型具有较大的改进空间，需要进行优化。