

深度学习与自然语言处理 作业 3 报告

魏才伦
1260034837@qq.com

Abstract

本文在金庸小说语料库上训练了 LSTM 模型预测词组, 然后提取模型中的词嵌入模块, 得到语料库中词组的词向量, 计算部分词语之间的词向量的相似度, 验证词向量的有效性。

Introduction

LSTM (Long Short-Term Memory, 长短期记忆) 是一种特殊的递归神经网络, 由 Sepp Hochreiter 和 Jürgen Schmidhuber 于 1997 年提出, 旨在解决标准 RNN 在长序列数据上训练时出现的梯度消失和梯度爆炸问题。LSTM 通过引入门控机制, 能够在长时间跨度上保持和更新信息, 从而在处理序列数据 (如时间序列、文本数据等) 方面表现出色。

Methodology

本文采用 jieba 模块对语料库分词, pytorch 框架搭建了 LSTM 模型, 模型首先通过 embedding 层将输入文本序列映射为词向量, 然后通过两层 LSTM 模型生成预测词组的特征向量, 最后通过全连接层将特征向量转化为词袋概率。为验证 embedding 层生成词向量的有效性, 本文将训练好的模型 embedding 层单独取出, 输入一系列词组, 查找这些词组的词向量相似度前 10 的词组, 根据相似词组的语意, 验证词向量的有效性。

M1: embedding 层

Embedding 层是一种用于将离散的高维度的输入数据映射到低维连续向量空间的层, 常用于自然语言处理 (NLP) 中的词向量表示。它的主要功能是将单词、符号或其他离散类型的数据转化为可以用于神经网络训练的密集向量。这种转化在很多机器学习任务中非常有

用，因为它能够捕捉和表达输入数据的语义和语法关系。

Embedding 层的主要特点和功能包括：将高维离散输入映射到低维连续空间，例如，在自然语言处理中，单词通常用独热编码（one-hot encoding）表示，这种编码方式会生成一个非常高维且稀疏的向量。Embedding 层能够将这种高维稀疏向量映射到低维密集向量，从而降低计算复杂度并捕捉更多的语义信息。学习数据中的语义关系，通过训练，Embedding 层能够学习并捕捉到输入数据中的语义和语法关系。例如，相似的单词在嵌入空间中会被映射到相近的向量。Embedding 层的参数（即嵌入矩阵）是可训练的，这意味着它们可以在特定任务上通过反向传播算法进行优化，从而提高模型的性能。

M2: LSTM 模型

LSTM（Long Short-Term Memory）是一种特殊的递归神经网络（RNN），它能够有效地捕捉和保留长期依赖信息，克服了传统 RNN 在处理长序列数据时容易遗忘早期信息的缺点。LSTM 通过引入门控机制，能够在序列数据处理中更好地控制信息的流动和记忆的保存。

LSTM 单元主要由四个部分组成：输入门、遗忘门、输出门和记忆单元。每个部分都由一个神经网络层实现，并且在时间步长上共享参数。

遗忘门决定当前时刻应该遗忘多少前一个时刻的记忆，遗忘门的公式表示为 $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$ 。输入门决定当前时刻应该从输入信息中选择多少加入到记忆单元中，公式表示为 $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ 。候选记忆单元创建新的候选记忆内容，公式表示为 $\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$ 。记忆单元更新当前记忆单元的状态，公式表示为 $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$ 。输出门决定当前时刻的输出信息，公式表示为 $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$ 。

LSTM 的工作流程是：遗忘门控制信息遗忘，决定哪些信息需要丢弃，即从记忆单元的状态中删除哪些信息。输入门控制信息存储，决定哪些新信息需要存储到记忆单元。记忆单元的更新，结合遗忘门和输入门的输出，更新记忆单元的状态。输出门控制信息输出，根据当前记忆单元的状态和输入，决定输出的隐藏状态。

M3: 程序流程

Step1: 程序首先处理语料库，读取语料库中的所有小说，采用 jieba 模块对其进行分词，并去除停用词，然后采用 collections 模块得到词组的词典与词频，并将每个词组对应一个序号。

Step 2: 准备训练数据集，将分词结果按小说顺序排列，20 个词组为一个序列，该序列的下一个词组为该序列的 target，即监督数据，以此生成一个给定词组序列预测该序列下一

个词组的数据集。

Step 3: 构建 LSTM 模型, 模型首先通过 embedding 层编码词向量, 然后通过 2 层 LSTM 预测, 最后通过全连接层输出最终结果。

Step 4: 训练 LSTM 模型, 设置 batch size 为 32, 采用 Adam 优化器, 初始学习率为 0.001, 训练 100 个 epochs。

Step 5: 训练结束保存模型权重, 提取 embedding 层, 输入测试词组, 输出该词组的词向量, 然后获取所有词组的词向量, 计算测试词组词向量与所有词组词向量的余弦相似度, 取相似度前 10 的词组, 观察词组的语意是否与测试词组相似。

Experimental Studies

测试人名词组['张无忌', '乔峰', '郭靖', '杨过', '令狐冲', '韦小宝'], 相似度前十的词向量如下表所示。

张无忌	乔峰	郭靖	杨过	令狐冲	韦小宝
赵敏 0.4555034	黄药师 0.57949513	杨康 0.4947024	萧峰 0.46266422	余沧海 0.50720954	沐剑屏 0.55187416
诸保昆 0.42657888	曹云奇 0.5247714	黄蓉心 0.4736065	赵志敬 0.4506517	盈盈 0.50249624	曹云奇 0.5490918
武烈 0.4187634	梅超风 0.4971576	闵柔 0.45811263	石破天 0.44389063	田伯光 0.45103016	水笙 0.5448878
游坦之 0.41356754	九难 0.49524692	郭靖知 0.4371178	陆无双 0.4436953	贾布 0.4147939	黄药师 0.5376978
黄蓉 0.41154957	法王 0.48353493	鲁有脚 0.42892727	愤愤 0.43916065	桃谷六仙 0.4141444	丁不三 0.53484845
全失 0.40171692	婆婆 0.46238986	黄蓉自 0.42639062	李莫愁 0.43348098	岳不群 0.41406596	石破天 0.53048277
姚清泉 0.40068105	包不同 0.4606694	黄蓉 0.42087573	顾左右 0.43021557	小昭 0.38228968	欧阳锋 0.49991143
岳不群 0.39780495	廖自砺 0.45923585	西毒 0.40927035	郭襄 0.42950213	面面相觑 0.37989038	法王 0.47386584
示弱 0.39600915	徐长老 0.45302176	白万剑 0.40922856	阿朱 0.42813438	傻子 0.37765345	刘元鹤 0.4651848
乔峰知 0.3943209	周仲英 0.45273182	苗人凤 0.40669006	闵柔 0.42690027	黄衣僧 0.3736344	海老公 0.46182966

可以发现,每个人名词组相似度排名前十的词组基本是人名,但部分人名的人物关系与小说语意并不相符,分析原因一是分词结果精度不高,出现了很多冗余词组;二是数据集将所有小说汇总,并未按小说分类;三是模型训练的目标是预测序列的下一个词组,该监督方法可能不适合生成稳健的词向量。

Conclusions

本文在金庸小说语料库上训练了 LSTM 模型预测词组,然后提取模型中的词嵌入模块,得到语料库中词组的词向量,选取部分人名词组计算该词组词向量与其他词组词向量的相似度,相似度排名前十的词组基本都是人名,说明词向量基本符合语意,但部分人名的人物关系与小说不符,说明词向量的语意理解仍有待提升。

References

- [1] https://blog.csdn.net/qq_74722169/article/details/134887728
- [2] https://blog.csdn.net/weixin_44966965/article/details/124732760
- [3] <https://blog.csdn.net/modi000/article/details/135620848>