词向量编程作业：汉语词向量

[2021213368 张瀚墨]

一、概述：使用SVD分解和SGNS两种方法构建汉语词向量并进行评测

二、SVD方法

奇异值分解(SVD), 可以用于矩阵压缩、特征降维。其在自然语言处理领域可以用于从文本数据中提取特征。

SVD的公式表示为：

$A = U\Sigma V^T$

其中：

- A 是一个 $m \times n$ 的矩阵。
- U 是一个 $m \times m$ 的单位正交矩阵，表示左奇异向量。
- Σ 是一个 $m \times n$ 的对角矩阵，其对角线上的元素是奇异值。
- V^T 是 V 的转置矩阵，V 是一个 $n \times n$ 的单位正交矩阵，表示右奇异向量。

在NLP中，A 可以是词-上下文共现矩阵，奇异值（Σ 中的元素）表示词和上下文之间的重要性。

模型参数与执行细节

- **选取的奇异值数量 (K)**: 由于词共现矩阵过于庞大(88116×88116)，在本实现中，我们选择了50个最大的奇异值(k=50)对其进行分解。
- **非零奇异值数量**: 由于条件不允许，无法计算完整矩阵分解后的对角元。我们发现计算的截断奇异值分解中，前50个值均为非零奇异值。
- **算法详述**: 我们使用 `scipy.sparse.linalg` 的 `svds` 方法来进行稀疏奇异值分解，只提取最大的K个奇异值及对应的奇异向量。这有助于降低计算复杂度，同时保留最重要的特征信息。

三、SGNS方法

实现原理：

SGNS(Skip-Gram with Negative Sampling)是一种有效的词向量训练方法，通过预测上下文来学习词的表示。

Skip-Gram模型的目标函数是最大化上下文词的条件概率，对于一个词 w 和它的上下文词 c，目标函数是：

$$\log \sigma(v_c \cdot v_w) + \sum_{i=1}^{k} \mathbb{E}_{c_i \sim P_n(w)} [\log \sigma(-v_{c_i} \cdot v_w)]$$

其中：

- v_w 和 v_c 分别是中心词 w 和上下文词 c 的向量表示。
- σ 是sigmoid函数，$\sigma(x) = \frac{1}{1+e^{-x}}$。
- 第一项是正样本（真实上下文词）的贡献，第二项是对于负样本（随机选择的非上下文词）的贡献。
- k 是负样本的数量。
- $P_n(w)$ 是负样本的分布，通常选择词频的3/4次幂作为分布。

简单的示例：

以句子`"I enjoy reading books"`为例，我们首先对它进行分词并构建词表：

```
{'I': 0, 'enjoy': 1, 'reading': 2, 'books': 3}
```

如果我们选择`"enjoy"`作为中心词，以及选择它的上下文词（考虑一个非常简单的情况，即上下文窗口大小为1，意味着只选择紧邻中心词的左右词作为上下文），那么上下文词就是`"I"`和`"reading"`。

在这个例子中：

中心词`"enjoy"`对应的索引是 1。

上下文词`"I"`和`"reading"`对应的索引分别是 0 和 2。

因此，对于中心词`"enjoy"`，我们将有：

```
word_indices = [1]

context_indices = [0, 2]
```

我们可以使用优化方法，调整词向量，从而优化目标函数。对此我提供了`sgns.py`代码实现，仅供用于这部分的实现原理演示，但是接下来的具体实施仍然采用`gensim`支持。

模型参数与执行细节

- **词向量的初始化**: `gensim.models.Word2Vec` 默认进行随机初始化。
- **词向量维数**: 我们设置了词向量的维数 `Word2Vec.vector_size` 为50。
- **学习率**: 使用 `gensim` 的默认学习率0.025。
- **训练批次大小**: `batch_words` 参数可以控制处理的单词数量，我们采用默认值10000。
- **训练轮数**: `epochs` 参数，默认为5轮。

算法执行流程

1. **语料准备**: 使用 `load_corpus` 加载并预处理语料，确保每一行代表一个句子，句子中的单词已经过分词。
2. **模型训练**: 使用 `gensim.models.Word2Vec` 类，以Skip-Gram模型初始化并训练词向量。我们设置 `sg=1` 以指定使用Skip-Gram算法。

四、评测

使用两种算法计算得到的词向量在测试词对之间计算相似度，并将结果输出到文件中

```
没戏 没辙 0 0

只管 尽管 0.9253325963119979 0.6800307035446167

GDP 生产力 0.910945532898512 0.7255964875221252

包袱 段子 0.3077204204126081 0.5849558115005493

由此 通过 0.9790470940016397 0.529334545135498

日期 时间 0.8954586496504959 0.6463411450386047
```

其中0代表词对其中至少一个不包含在训练语料中。