深度学习与自然语言处理
第四次大作业

Word2Vec

院（系）名称	自动化科学与电气工程学院
学 生 学 号	ZY2103803
学 生 姓 名	李鑫磊

2022 年 05 月

一、问题描述

利用给定语料库（或者自选语料库），利用神经语言模型（如：Word2Vec， GloVe 等模型）来训练词向量，通过对词向量的聚类或者其他方法来验证词向量的有效性。

二、Word2vec 模型

Word2vec，为一群用来产生词向量的相关模型。这些模型为浅层双层的神经网络，用来训练以重新建构语言学之词文本。网络以词表现，并且需猜测相邻位置的输入词，在 word2vec 中词袋模型假设下，词的顺序是不重要的。

训练完成之后，word2vec 模型可用来映射每个词到一个向量，可用来表示词对词之间的关系。该向量为神经网络之隐藏层。

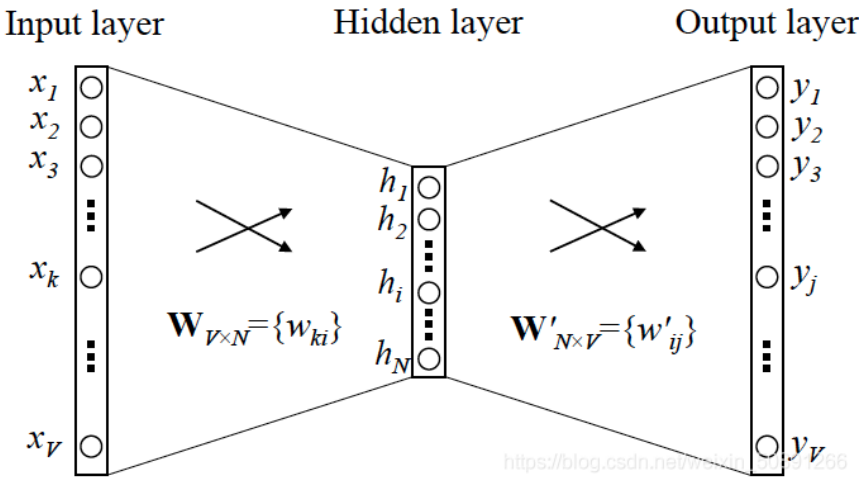
Word2vec 依赖 skip-grams 或 CBOW（连续词袋）来建立神经词嵌入。

2.1 One-hot encoder

one-hot encoder，其思想跟特征工程里处理类别变量的 one-hot 一样，本质上是用一个只含一个 1、其他都是 0 的向量来唯一表示词语。

2.2 Skip-gram 和 CBOW

Skip-gram 的网络结构如下，x 是 one-hot encoder 形式的输入，y 是在这 V 个词上输出的概率。



隐层的激活函数其实是线性的，相当于没做任何处理，我们要训练这个神经网络，用反向传播算法，本质上是链式求导。

当模型训练完后，最后得到的其实是神经网络的权重，比如现在输入一个 x 的 one-hot encoder: [1,0,0,⋯,0]，对应刚说的那个词语『我』，则在输入层到隐含层的权重里，只有对应 1 这个位置的权重被激活，这些权重的个数，跟隐含层节点数是一致的，从而这些权重组成一个向量 v_x 来表示 x，而因为每个词语的 one-hot encoder 里面 1 的位置是不同的，所以，这个向量 v_x 就可以用来唯一表示 x。

此外，输出 y 也是用 V 个节点表示的，对应 V 个词语，所以把输出节点置成 [1,0,0,⋯,0]，它也能表示『我』这个单词，但是激活的是隐含层到输出层的权重，这些权重的个数，跟隐含层一样，也可以组成一个向量 v_y，跟上面提到的 v_x 维度一样，并且可以看做是词语『我』的另一种词向量。而这两种词向量 v_x 和 v_y，正是 Mikolov 在论文里所提到的，『输入向量』和『输出向量』，一般我们用『输入向量』。

这个词向量的维度（与隐含层节点数一致）一般情况下要远远小于词语总数 V 的大小，所以 Word2vec 本质上是一种降维操作——把词语从 one-hot encoder 形式的表示降维到 Word2vec 形式的表示。

三、程序实现

3.1 导入各种包和数据集

导入所需要的 jieba 模块、gemsim 模块和其他相关模块并加载文件路径。

3.2 文本过滤

过滤掉文本中的标点符号和一些训练词向量时不需要，单独出现并没有什么意义的停词(stop word)。停词表选用的是网上通用的中文语料停词表，我又手动在里面添加了部分武侠小说里常出现的对训练词向量没什么意义的词汇和短语。这里把标点符号当作停用词合并一起处理。

3.3 添加自定义词汇

jieb 的词汇表中并没有收录很多金庸的武侠小说这种特定环境下的很多专有名词，包括一些重要人物的名称，一些重要的武功等。这里整理了三份 txt 文本，分别记录了

武侠小说中的人物名称、武功名称和门派名称，并把这些词汇添加到词汇表中。

3.4 分词

将所有文本分词并显示每一本书分词后的行数和总行数，通过一个嵌套列表存储，每一句为列表中一个元素，每一句又由分好的词构成一个列表，这也是 word2vec 训练时需要输入的格式。

3.4 训练词向量

Gemsim 模块是一个功能很强大的 NLP 处理模块，这里用到了 Gemsim 模块中 Word2Vec 函数。

3.5 人物的关系分析

通过模型里的 similarity、most_similar 等函数对各个要素进行分析。

四、运行结果

通过该模型分析人物关系亲进度，选取“张无忌”为对象，找出与其最亲密的 10 个人，结果如下表：

赵敏	0.633
周芷若	0.542
杨逍	0.505
小昭	0.495
波斯	0.477
范瑶	0.461
明教	0.460
韦一笑	0.451
东方不败	0.447
殷正天	0.443

通过分析得到的结果与实际关系亲近度排名较为接近，因此该模型的聚类分析较为

准确。