skip-gram 模型简介

2021113140　符世博

一、 模型介绍

1.1 模型概括

skip-gram 模型是由谷歌在 2013 年提出的用于自然语言处理中的词向量表示模型。网络架构如图 1。其目标为通过给定一个中心词，预测中心词的上下文。为了实现这一目标，skip-gram 尝试学习单词的分布式表示，将单词转换为密集的向量，使得语义相似的单词在向量空间中距离较近。模型输入为中心词的独热码，输出为上下文的独热码。

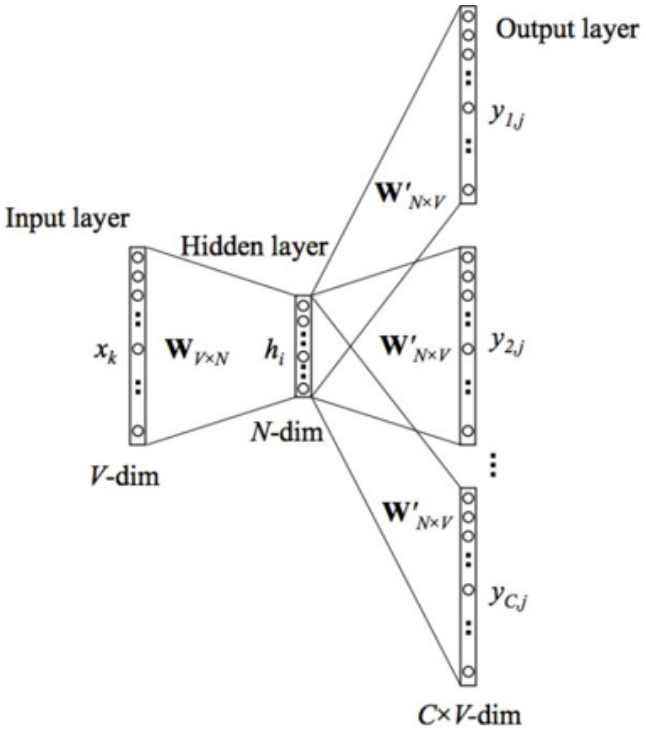

图 **1**　**skip-gram** 模型网络架构

1.2 独热码

这里简单介绍一下独热码。假设现在有一篇文档，遍历之后发现其一共包含 N 个词，给这 N 个词依次标号 $1, 2, 3...$。那么独热码就是这样一个 N 维向量，其只有一个位置是 1，其余位置全为 0，假设其第 i 个分量是 1，那么这个独热码就代表编号为 i 的词。

1.3 几个重要模型参数

下面介绍 skip-gram 模型中的几个重要的参数：

- N：嵌入维度；
- skip_window：从当前中心词的两侧侧选取多少个词构成词窗口；
- num_skips：从窗口中选取多少个词作为输出词与中心词构成训练数据。

N 决定了当前的词会被嵌入到多少维空间中，即用多少维的向量表示一个词。这也是模型架构中输入层到隐含层的转换矩阵。

skip_window 和 num_skips 决定了训练数据是什么样的。以 i like eat apple 为例，这里取 skip_window=1，num_skips=2，中心词为 eat。那么从 eat 的左右两侧各取 skip_window 个词，也就是各取 1 个词构成当前的词窗口 ['like', 'eat','apple']，之后根据 num_skips 从词窗口中选择词作为输出词与当前中心词组成一个训练数据，那么这里组成的训练数据就是 ('eat','apple') 和 ('eat','like')。

1.4 模型训练

skip-gram 模型在训练时：

- 隐含层无激活函数；
- 输出层采用 softmax 作为激活函数；
- 使用 negative sampling 作为损失函数。

之所以采用 softmax 作为激活函数，如式 (1) 是因为模型输出是与独热码进行做差，如果不做归一化计算残差效果不是很好。使用 softmax 进行归一化后再与独热码计算残差效果会好很多。

$$s_i = \frac{e^{z_i}}{\sum_1^V e^{z_j}} \tag{1}$$

采用 negative sampling 作为损失函数，如式 (2) 而非传统的 softmax 损失函数的原因是：

- 传统的损失函数需要对所有词汇进行计算;
- negative sampling 通过对负样本进行采样，显著减小计算量。

$$L(\theta) = -\log\left(\sigma(V_c \cdot V_\omega)\right) + \sum_1^k \log\left(\sigma(-V_c \cdot V_{\omega i})\right) \tag{2}$$

1.5 模型应用

其实输入层到隐含层的权重矩阵 W 的每一行就是对应词的词向量。那么根据这一矩阵可以将 skip-gram 模型应用到以下任务中：

- 词向量表示：用于词汇相似度计算、词汇聚类、词汇情感分析等任务；
- 文档表示：通过将文档中的词汇的词向量进行平均或加权平均，得到整个文档的表示，用于文档相似度计算、文档分类等任务；
- 输入特征：词向量可以作为神经网络和其他机器学习模型的输入特征；
- 信息检索：词向量可以用于构建词汇的向量空间模型用于信息检索和相关性排序。

二、 运行实例

这里，使用网络上的公开数据集运行 skip-gram 算法，得到语料中词频 top100 词的 128 维词向量表示。将其压缩到 2 维后进行可视化如 图 2

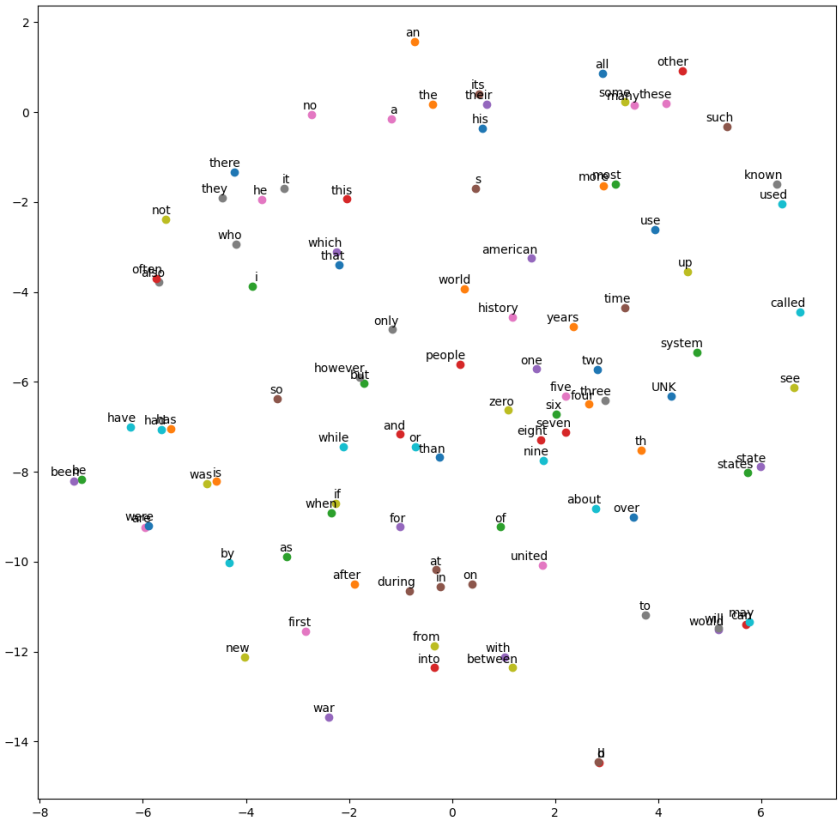

图 **2**　词向量的可视化

可以看出，语义相近的词距离更加接近，这也证明了 skip-gram 算法的可靠性。