

转

理解GloVe模型（+总结）

2018年03月21日 16:36:00

蜗牛蜗牛慢慢爬

阅读数 14293

文章目录

- 系列目录（系列更新中）
- 概述
- 统计共现矩阵
- 使用GloVe模型训练词向量
- 模型公式
- 模型怎么来的
- Glove和skip-gram、CBOW模型对比
- 实战教程

系列目录（系列更新中）

- 第二讲 cs224n系列之word2vec & 词向量

• word2vec进阶之skim-gram和CBOW模型（Hierarchical Softmax、Negative Sampling）

• 第三讲 cs224n系列之skip-pram优化 & Global Vector by Manning & 词向量评价

• 理解GloVe模型（+总结）

概述

- 模型目标：进行词的向量化表示，使得向量之间尽可能多地蕴含语义和语法的信息。

• 输入：语料库

• 输出：词向量

• 方法概述：首先基于语料库构建词的共现矩阵，然后基于共现矩阵和GloVe模型学习词向量。
- ** 开始 -> 统计共现矩阵 -> 训练词向量 -> 结束**

统计共现矩阵

设共现矩阵为X，其元素为 $X_{i,j}$ 。

$X_{i,j}$ 的意义为：在整个语料库中，单词i和单词j共同出现在一个窗口中的次数。

举个栗子：

设有语料库：

1





|

i love you but you love him i am sad

这个小小的语料库只有1个句子，涉及到7个单词：i、love、you、but、him、am、sad。

如果我们采用一个窗口宽度为5（左右长度都为2）的统计窗口，那么就有以下窗口内容：

窗口标号	中心词	窗口内容
0	i	i love you
1	love	i love you but
2	you	i love you but you
3	but	love you but you love
4	you	you but you love him
5	love	but you love him i

窗口标号	中心词	窗口内容		
6	him	you love him i am		18
7	i	love him i am sad		19
8	am	him i am sad		
9	sad	i am sad		

窗口0、1长度小于5是因为中心词左侧内容少于2个，同理窗口8、9长度也小于5。

以窗口5为例说明如何构造共现矩阵：

中心词为love，语境词为but、you、him、i；则执行：

$$X_{love,but} += 1$$

$$X_{love,you} += 1$$

$$X_{love,him} += 1$$

$$X_{love,i} += 1$$

使用窗口将整个语料库遍历一遍，即可得到共现矩阵X。

使用GloVe模型训练词向量

模型公式

先看模型，代价函数长这个样子：

$$J = \sum_{i,j}^N f(X_{i,j})(v_i^T v_j + b_i + b_j - \log(X_{i,j}))^2$$

v_i ， v_j 是单词i和单词j的词向量， b_i ， b_j 是两个标量（作者定义的偏差项）， f 是权重函数（具体函数公式及功能下一节介绍），N是词汇表的大小（共N*N）。

可以看到，GloVe模型没有使用神经网络的方法。

模型怎么来的

那么作者为什么这么构造模型呢？首先定义几个符号：

$$X_i = \sum_{j=1}^N X_{i,j}$$

其实就是矩阵单词i那一行的和；

$$P_{i,k} = \frac{X_{i,k}}{X_i}$$

条件概率，表示单词k出现在单词i语境中的概率；

$$ratio_{i,j,k} = \frac{P_{i,k}}{P_{j,k}}$$

两个条件概率的比率。

作者的灵感是这样的：

作者发现， $ratio_{i,j,k}$ 这个指标是有规律的，规律统计在下表：

$ratio_{i,j,k}$ 的值	单词j,k相关	单词j,k不相关
单词i,k相关	趋近1	很大
单词i,k不相关	很小	趋近1

很简单的规律，但是有用。

思想：假设我们已经得到了词向量，如果我们用词向量 v_i 、 v_j 、 v_k 通过某种函数计算 $ratio_{i,j,k}$ ，能够同样得到这样的规律的话，就意味着我们词向量-有很好的-一致性，也就说明我们的词向量中蕴含了共现矩阵中所蕴含的信息。

设用词向量 v_i 、 v_j 、 v_k 计算 $ratio_{i,j,k}$ 的函数为 $g(v_i, v_j, v_k)$ （我们先不去管具体的函数形式），那么应该有：

$$\frac{P_{i,k}}{P_{j,k}} = ratio_{i,j,k} = g(v_i, v_j, v_k)$$

即：

$$\frac{P_{i,k}}{P_{j,k}} = g(v_i, v_j, v_k)$$

即二者应该尽可能地接近；
很容易想到用二者的差方来作为代价函数：

$$J = \sum_{i,j,k}^N (\frac{P_{i,k}}{P_{j,k}} - g(v_i, v_j, v_k))^2$$

但是仔细一看，模型中包含3个单词，这就意味着要在 N^3 的复杂度上进行计算，太复杂了，最好能再简单点。
现在我们来仔细思考 $g(v_i, v_j, v_k)$ ，或许它能帮上忙；
作者的脑洞是这样的：

- 1. 要考虑单词i和单词j之间的关系，那 $g(v_i, v_j, v_k)$ 中大概要有这么一项吧： $v_i - v_j$ ； 嗯，合理，在线性空间中考察两个向量的相似性，不失线性地， $v_i - v_j$ 大概是个合理的选择；
- 2. $ratio_{i,j,k}$ 是个标量，那么 $g(v_i, v_j, v_k)$ 最后应该是个标量啊，虽然其输入都是向量，那内积应该是合理的选择，于是应该有这么一项吧： $(v_i - v_j)^T v_k$ ；
- 3. 然后作者又往 $(v_i - v_j)^T v_k$ 的外面套了一层指数运算 $exp()$ ，得到最终的 $g(v_i, v_j, v_k) = exp((v_i - v_j)^T v_k)$ ；
最关键的第3步，为什么套了一层 $exp()$ ？
套上之后，我们的目标是让以下公式尽可能地成立：

$$\frac{P_{i,k}}{P_{j,k}} = g(v_i, v_j, v_k)$$

即：

$$\frac{P_{i,k}}{P_{j,k}} = exp((v_i - v_j)^T v_k)$$

即：

$$\frac{P_{i,k}}{P_{j,k}} = exp(v_i^T v_k - v_j^T v_k)$$

即：

$$\frac{P_{i,k}}{P_{j,k}} = \frac{exp(v_i^T v_k)}{exp(v_j^T v_k)}$$

然后就发现找到简化方法了：只需要让上式分子对应相等，分母对应相等，即：
 $P_{i,k} = exp(v_i^T v_k)$ 并且 $P_{j,k} = exp(v_j^T v_k)$

然而分子分母形式相同，就可以把两者统一考虑了，即：

$$P_{i,j} = exp(v_i^T v_j)$$

本来我们追求：

$$\frac{P_{i,k}}{P_{j,k}} = g(v_i, v_j, v_k)$$

现在只需要追求：

$$P_{i,j} = exp(v_i^T v_j)$$

两边取个对数：

$$log(P_{i,j}) = v_i^T v_j$$

那么代价函数就可以简化为：

$$J = \sum_{i,j}^N (\log(P_{i,j}) - v_i^T v_j)^2$$

现在只需要在NN的复杂度上进行计算，而不是MN*N，现在关于为什么第3步中，外面套一层exp()就清楚了，正是因为套了一层exp()，进而等式两边分子分母对应相等，进而简化模型。

然而，出了点问题。
仔细看这两个式子：

$$\log(P_{i,j}) = v_i^T v_j \text{ 和 } \log(P_{j,i}) = v_j^T v_i$$

$\log(P_{i,j})$ 不等于 $\log(P_{j,i})$ 但是 $v_i^T v_j$ 等于 $v_j^T v_i$ ；即等式左侧不具有对称性，但是右侧具有对称性。

数学上出了问题。
补救一下好了。

现将代价函数中的条件概率展开：

$$\log(P_{i,j}) = v_i^T v_j$$

即为：

$$\log(X_{i,j}) - \log(X_i) = v_i^T v_j$$

将其变为：

$$\log(X_{i,j}) = v_i^T v_j + b_i + b_j$$

即添了一个偏差项 b_j ，并将 $\log(X_i)$ 吸收到偏差项 b_i 中。
于是代价函数就变成了：

$$J = \sum_{i,j}^N (v_i^T v_j + b_i + b_j - \log(X_{i,j}))^2$$

然后基于出现频率越高的词对儿权重应该越大的原则，在代价函数中添加权重项，于是代价函数进一步完善：

$$J = \sum_{i,j}^N f(X_{i,j})(v_i^T v_j + b_i + b_j - \log(X_{i,j}))^2$$

具体权重函数应该是怎么样的呢？
首先应该是非减的，其次当词频过高时，权重不应过分增大，作者通过实验确定权重函数为：

$$f(x) = \begin{cases} (x/xmax)^{0.75}, & \text{if } x < xmax \\ 1, & \text{if } x \geq xmax \end{cases}$$

到此，整个模型就介绍完了。

Glove和skip-gram、CBOW模型对比

Cbow/Skip-Gram 是一个local context window的方法，比如使用NS来训练，缺乏了整体的词和词的关系，负样本采用sample的方式会缺失词的关系
另外，直接训练Skip-Gram类型的算法，很容易使得高曝光词汇得到过多的权重

Global Vector融合了矩阵分解Latent Semantic Analysis (LSA)的全局统计信息和local context window优势。融入全局的先验统计信息，可以加快精度，又可以控制词的相对权重。

我的理解是skip-gram、CBOW每次都是用在一个窗口中的信息更新出词向量，但是Glove则是用了全局的信息（共线矩阵），也就是多个窗口

实战教程

[GloVe 教程之实战入门+python gensim 词向量](#)

参考链接：
[理解GloVe模型](#)

理解GloVe模型 (Global vectors for word representation)

阅读量 2万+

理解GloVe模型概述模型目标：进行词的向量化表示，使得向量之间尽可能多地蕴含... 博文 来自： [饺子醋的...](#)



想对作者说点什么