无痛理解word2vec

MaybeS⋯

最是人间留不住，朱颜辞镜花辞树

关注他

109 人赞同了该文章

摘要：一句话，word2vec就是用一个一层的神经网络(CBOW的本质)把one-hot形式的词向量映射为分布式形式的词向量，为了加快训练速度，用了Hierarchical softmax，negative sampling 等trick。

自从Google开源word2vec之后，因为其速度快效果好，迅速成为业内津津乐道的话题，有团队还专门写了一个册子，来分析其代码。在这里我们做一个提炼总结，了解其精髓思想。

word2vec涉及到很多自然语言处理的名词。首先是**词向量**(word vector)，图像和音频等信号都可以用一个矩阵或者向量表示，所以我们也希望用一个数学方法来表达单词，这样可以方便的用于各种后续计算，这就是词向量。

词向量最简单的方式是1-of-N的***one-hot方式***，也就是从很大的词库corpus里选V个频率最高的词(忽略其他的)，V一般比较大，比如V＝10W，固定这些词的顺序，然后每个词就可以用一个V维的稀疏向量表示了，这个向量只有一个位置的元素是1，其他位置的元素都是0。One hot方式其实就是简单的直接映射，所以缺点也很明显，维数很大，也没啥计算上的意义。

于是第二种方法就出现了***分布式词向量***(distributed word representation), 分布式词向量是一个固定大小的实数向量，事前确定它的大小比如N＝300维或者N＝1000维，每个元素都是一个实数，实数的具体值是词库里面每个词通过不同的贡献得来的，所以叫分布式的。而word2vec就是一种学习这个分布式词向量的算法。

分布式词向量并不是word2vec的作者发明的，他只是提出了一种更快更好的方式来训练也就是：**连续词袋模型Continous Bag of Words Model(CBOW)**和**Skip-Gram Model**。这两种都是训练词向量的方法，可以选择其一，不过据论文说CBOW要更快一些(1天vs.3天的区别)。**统计语言模型** statistical language model就是给你几个词，在这几个词出现的前提下来计算某个词出现的（事后）概率。CBOW也是统计语言模型的一种，顾名思义就是根据某个词前面的C个词或者***前后***C个连续的词，来计算某个词出现的概率。Skip-Gram Model相反，是根据某个词，然后分别计算它前后出现某几个词的各个概率。

以“我爱北京天安门”这句话为例。假设我们现在关注的词是“爱”，C＝2时它的上下文分别是“我”，“北京天安门”。CBOW模型就是把“我” “北京天安门” 的one hot表示方式作为输入，也就是C个1xV的向量，分别跟同一个VxN的大小的系数矩阵W1相乘得到C个1xN的隐藏层hidden layer，然后C个取平均所以只算一个隐藏层。这个过程也被称为线性激活函数(这也算激活函数？分明就是没有激活函数了)。然后再跟另一个NxV大小的系数矩阵W2相乘得到1xV的输出层，这个输出层每个元素代表的就是词库里每个词的事后概率。输出层需要跟ground truth也就

是“爱”的one hot形式做比较计算loss。这里需要注意的就是V通常是一个很大的数比如几百万，计算起来相当费时间，除了“爱”那个位置的元素肯定要算在loss里面，word2vec就用基于huffman编码的Hierarchical softmax筛选掉了一部分不可能的词，然后又用nagetive samping再去掉了一些负样本的词所以时间复杂度就从O(V)变成了O(logV)。Skip gram训练过程类似，只不过输入输出刚好相反。

训练完成后对于某个词就可以拿出它的1xN的隐藏层作为词向量，就可以w2v(中国)－w2v(北京)＝w2v(法国)－w2v(巴黎)了。